2012 Third International Conference on Emerging Applications of Information Technology (EAIT)

Holistic Recognition of Handwritten Tamil Words

Thadchanamoorthy Subramaniam
Department of Computer Science
Trincomalee Campus, Eastern University, Sri Lanka
Sri Lanka
stmoorrthy@gmail.com

Umapada Pal
CVPR Unit
Indian Statistical Institute
India
umapada@isical.ac.in

H. Premaretne; N. Kodikara
School of Computing
University of Colombo
Sri Lanka
hlp;ndk@ucsc.cmb.ac.lk

***Abstract*—In this paper, we describe a writer independent method for recognizing offline unconstrained handwritten Tamil words/strings. Touching and overlapping are main bottleneck of handwriting recognition and to overcome this, we adopted a holistic approach here. To handle various handwritings of different individuals, at first, some pieces of preprocessing work are done on an input word. Next, Gabor based features are computed on the processed word. These Gabor features along with other geometric features of the word image are then fed to an SVM classifier for recognition. In our experimental study, we have used 4270 samples (collected from the class of 217 country names) written in Tamil and obtained 86.36% recognition rate.**

Keywords- Handwritten Character Recognition, Unconstrained Tamil word recognition, Holistic approach, Gabor features, SVM

I. INTRODUCTION

Recognition of handwritten characters has been a popular research area for many years because of its various application potentials. Some of its potential application areas are Postal Automation, Bank cheque processing, automatic data entry, etc. There are many pieces of work towards handwritten recognition of Roman, Japanese, Chinese and Arabic scripts, and various approaches have been proposed by the researchers towards handwritten character recognition [1]. Although there are some reports on offline Tamil isolated handwritten character recognition [2-3], only a few reports is available for offline handwritten Tamil word recognition. In [4], only 40 words are considered with the concept of HMM and obtained a 80.75% recognition accuracy. To take care of various handwritings of different individuals, in this paper a holistic approach is proposed for recognition of offline unconstrained handwritten Tamil words.

Tamil is an ancient, classical Dravidian language in existence for over two thousand years. The Tamil script traces its roots to the Brahmin script and continues to undergo a lot of changes to transgress itself as a portable medium. There are 30 basic shapes (12 vowels and 18 consonants) in Tamil. In addition to this there are six Grantha characters and one Ayutham**.** These shapes are shown in Fig.1. With the combination of these 30 basic shapes and one ayutham letter, in total we get two hundred and forty seven characters that are used in Tamil language.

Touching with nearby letters is a serious problem in the area of handwritten character recognition. Among the rare solutions for this problem, the water reservoir concept is a very popular technique used to segment two digit numbers and to segment the characters with the headline in Bangla words [5].

Vowels	அ ஆ இ ஈ உ ஊ எ ஏ ஐ ஒ ஓ ஔ
Ayutham	ஃ
Consonants	க் ங் ச் ஞ் ட் ண் த் ந் ப் ம் ய் ர் ல் வ் ழ் ள் ற் ன்
Grantha letters in use	ஜ் ஸ் ஷ் ஹ் க்ஷ் ஸ்ரீ

Figure 1. Basic shapes of Tamil characters

To handle various toughing and overlapping of handwritings of different individuals in this paper, at first, some pieces of preprocessing work are done on an input word. Through preprocessing some of the unwanted parts from the handwritten words are removed and some of the lost information are obtained which are helpful for recognition. Next, Gabor based features along with other geometric features of the word image are computed and fed to an SVM classifier for recognition.

Rest of the paper is organized as follows. Different preprocessing steps utilized in our scheme are discussed in Section II. Section III deals with feature extraction and Section IV of the paper deals with data preparation and classifier. Detailed experimental results are demonstrated in Section V. Finally, the paper is concluded in Section VI.

II. PRE-PROCESSING

To handle various types of handwritings of different individuals, a few preprocessing steps performed and they are Binarizing, Standardizing, Thinning, Aligning, Clipping, and Pruning. Brief descriptions of them are given below.

A. Binarizing

The Otsu's Algorithm function from the MatLab Programming Tool is used to binarize the input image.

B. Standadizing

As the height of a word may vary from person to person, the heights of all words are standardized to 100 pixels and the length is also scaled to as per the height ratio.

C. Thining Process

After the standardization, the word is dilated with a 2 x 2 structuring elements and undergoing for a thinning process. And also the thinning process takes place at every Gabor feature extraction processes after the dilation operation with suitable structuring elements.

978-1-4673-1827-3/12/$31.00 ©2012 IEEE

D. Aligning

The letters are found individually as well as combined at different vertical positions. Therefore, these letters are aligned on a center line so that features could be extracted easily. An alignment results is shown in Figure 2.

a. A Tamil Country Name b. Results after alignment

Figure 2. Aligning along a central line

E. Prunning

Sometimes a part of a Tamil letter may be extended beyond the actual portion of the letter, when cursively written. And also, when thinning process is carried out, some protrusions are left at the end of lines. This prolongs and unwanted protrusions may affect the actual geometrical features of the country name. Therefore, an algorithm is developed to remove those unwanted effects from the main body of the word. Results of pruning, done on the image of Fig.2 (a), are shown in Fig. 3.

Figure 3. Unwanted smaller protrusions and prolongs removed

F. Clipping

The unwanted single line extensions incurred above the top and below the bottom of the main body of the word are clipped off so that the size of the word image could be more reasonably manageable. Output of various level of clipping is shown in Fig.4.

a. Clipped Up and Down b. Clipped at the middle

Figure 4. Examples of various levels of clipping

III. FEATURE EXTRACTION

We considered the geometrical features of the word image as they are independent of the behavior of overlapping/touching. We apply Gabor based technique to obtain some of those features by applying in the required orientation. Different features are discussed below.

A. Gabor Filter based Features

There are four Gabor filters used with 0,90,45,-45 degree orientations.

Gabor Features: Gabor filters are capable of representing signals in both frequency and time domain. A two-dimensional Gabor filter in spatial and frequency domain can be defined by the following formula:

$$g(x,y;\lambda,\theta,\psi,\sigma,\gamma) = \exp\left(-\frac{x'^2+\gamma^2y'^2}{2\sigma^2}\right)\cos\left(2\pi\frac{x'}{\lambda}+\psi\right), \quad (1)$$

where $x' = x\cos\theta + y\sin\theta$

and $y' = -x\sin\theta + y\cos\theta$

In this equation, λ represents the wavelength of the cosine factor, θ represents the orientation of the normal to the parallel stripes of a Gabor function, ψ is the phase offset, σ is the sigma of the Gaussian envelope and γ is the spatial aspect ratio, and specifies the ellipticity of the support of the Gabor function. We tried a combination of different values of those parameters during our experiment.

The method used in extracting features is as follows:
(i) Slide a non-overlapping window of *NxN* pixels over the fragment image. (ii) Compute the corresponding Gabor value within each *NxN* window using the formula above. (iii) Encode the Gabor value of each of the pixels in the sliding window as a vector component of our feature vector. We experimented with the window size of 7 *x* 7 pixels.

B. Number of Dots

In Tamil the consonant letters contain small dots or smaller closed circles on top of the consonant letters. For example: க், ன், ட்,ல் etc. Also letters like ஃ and ஈ have dots in it. All these dots are counted relative to left, center and right positions of the word and we have used these possitional counts as three features from the word concerned. To get the features we divide a word into three equal parts vertically and count the number of dots in each of the three parts as shown in Fig.5. This number gives us three features.

1	0	0

Figure 5. Positional distribution of dots in 3 parts is shown.

C. Number of Horizontal Lines

Gabor filter with zero degree orientation is used to detect the horizontal line pieces present in the word image. Since many lines may not properly lie horizontally in unconstrained cursive letters, a dilation process is conducted so that the very nearby horizontal lines could be combined together. After a thinning process, these combined lines will appear as one single line. Therefore, the exact number of horizontal lines could be counted easily. To compute the feature, at first, we divide a word into four equal parts vertically (as shown in Fig. 6) and count the number of horizontal lines presents in each of the four parts. This number gives us four features. For illustration see Fig.6 where dot shows the CG of each line. Number of CG points present in a part gives us the number of horizontal lines presents in that part. In this image, we obtained 5, 7, 5 and 3 horizontal lines from 1st, 2nd, 3rd and 4th parts, respectively. Similarly, we also divide a word into three equal parts horizontally and count the number of horizontal lines presents in each of these horizontal parts and this number gives us three features. As a results we have 7 (4+3) features based on horizontal lines.

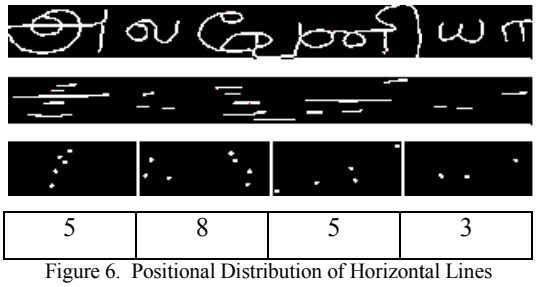
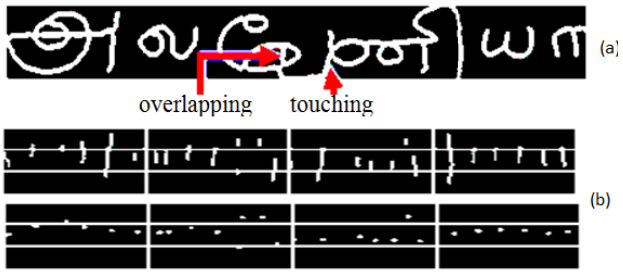


Figure 6. Positional Distribution of Horizontal Lines

D. Number of Vertical Lines

In this case, Gabor filter is oriented with an angle of 90 degree. After a dilation process, like the case of horizontal line, combining nearby vertical line pieces together, it has been thinned and the lines are counted. We divide the image into three equal parts horizontally (top, middle and bottom) and four equal parts vertically. Thus we have 12 parts and we count the number of vertical lines in each of these parts. (For illustration see Fig.7(c) where dot shows the CG of each line. Number of CG points present in a part gives us the number of vertical lines presents in that part).Thus for 12 parts we get 12 features from the vertical lines.



Number of vertical lines before (after) the correction applied:

1 (1)	2 (2)	1(1)	0 (0)
5 (5)	5 (5)	6(7)	6 (6)
0 (0)	1 (1)	0 (0)	0 (0)

(c)

Figure 7. Positional Distribution of Vertical Lines

In addition to this, lost vertical lines due to touching effects are also compensated at the appropriate positions using a simple checking of the gradients of the lines coming towards the touching point. The following three cases (see Table I) are considered for getting the information of lost vertical line. But vertical line touching is impossible to recover. This information is used to get the correct number of vertical line in spite of any loss of vertical lines due to overlapping and touching. To get the correct number of vertical line following technique is applied. Simply by analyzing both sides of every vertical line, any curve that touching the line is identified with a help of a 2 column matrix, which has rows equal to number of lines cut by those two columns. And the continuity of the other side also verified. If there exists two column matrix with at least one row of non-zero values, we assume that a lost vertical line exists there and its number is corrected as shown in Fig.7(c). (see Fig. 8 for illustration where two column matrix on the left and right side for the example are shown).

Table I. Type of touching points handled

Curve at the left of vertical line	Curve at the right of vertical line	Curves both side of vertical line

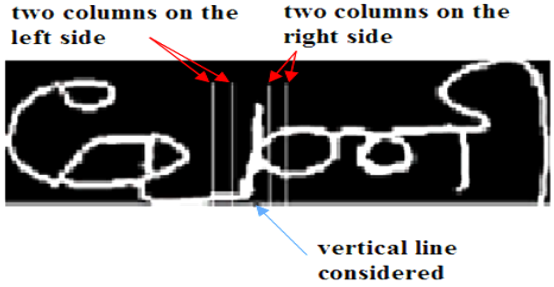


Figure 8. Consideration of two closest columns on either side of the vertical line of touching

Table II. Possiblities of Touching Points

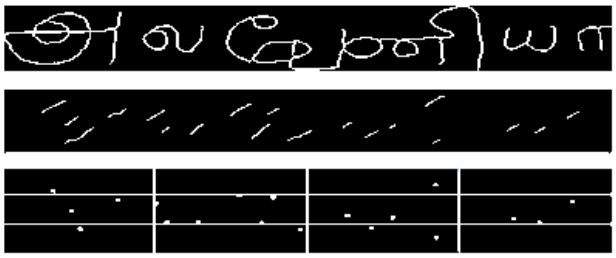
Case	Left side	Right side	Height of vertical line	Example
01	Line	Curve	Big	
02	Curve	Nothing	Big	(Within the letter)
03	Curve	Line	Big	
04	Curve	Curve	Small (normally)	
05	Curve	Curve	Big (Due to the nature of writing)	

Therefore, it is ensured that there is a pair of curves with the nature of descending and ascending towards the vertical line. The Table –II provides the possibilities of all kinds of touching behavior of such cases. A curve is decided by the pair of descending and ascending lines. The existences of a line or a curve ensure the continuity in that side. See Fig.9 for illustrations.

A. Number of +45 and -45 Degree Slated Lines

Once again the Gabor filter is oriented with +45 degree and then with -45 degree and then thinned followed by a dilation. Like vertical line features, slanted lines are then counted according to three equal horizontal (top middle and bottom) positions and four equal vertical positions. We have 12 features from +45 degree slanted lines and another 12 features from -45 degree slanted lines.

So, finally we have 46 features for the input of the SVM classifier and the details of these features are given in Table III.



1	0	1	0
1	6	2	3
1	1	2	0

Figure 9. Positional Distribution of+45 degree slanted Lines (Number of +45 degree slanted lines are shown in the table)

IV. DATA PREPARATION AND CLASSIFIER

In this attempt, we used 217 country name classes. There are 4270 samples collected from all the classes. Each of the sample was binarised, standardised and aligned along a center line. At this stage, the number of dots were counted at three positions and removed from the image. This image was undergone for a clipping process followed by a pruning process and once again clipped and aligned. Finally for each orientation, the image was convolved with coressponding Gabor filter kernel. The output of the Gabor filter is converted to binary format and the lines were counted with positional information.

Table III. List of Features

Sl No.	Feature Description	Feat-ure no.
1	Number of Dots (Left, middle, Right)	03
2	Number of Horizontal lines (top, middle, bottom) (Equal four vertical divisions)	07
3	Number of vertical lines (three horizontal and equal four vertical parts)	12
4	Number of +45 degree slanted lines (three horizontal and equal four vertical parts)	12
5	Number of -45 degree slanted lines (three horizontal and equal four vertical parts)	12
TOTAL Features		46

These features were converted to a text format along with the class value for each sample. In this case each country name was given an ordinal number from 1 to 217 as its class value. This text file was given as input to the classifier. In our experiments, we have used a Support Vector Machine (SVM) as classifier. The SVM is defined for two-class problem and it looks for the optimal hyper plane, which maximizes the distance, the *margin*, between the nearest examples of both classes, named support *vectors* (SVs). Given a training database of M data: $\{x_m | m=1,...,M\}$, the linear SVM classifier is then defined as:

$$f(x) = \sum_j \alpha_j x_j \cdot x + b \qquad (2)$$

where $\{x_j\}$ are the set of support vectors and the parameters α_j and b has been determined by solving a quadratic problem [6]. The linear SVM can be extended to various non-linear form, and details can be found in [6] [7]. In our experiments we noted Gaussian kernel SVM outperformed other non-linear SVM kernels, hence we are reporting our recognition results based on Gaussian kernel only. The Gaussian kernel is of the form:

$$[k(x,y) = \exp(-\frac{\| x-y \|^2}{2\sigma^2})]. \qquad (3)$$

The parameters, gamma and C are set to an optimal value with the RBF Kernal so that maximum recognition rate could be obtained.

V. RESULTS

A. Data details

As mentuioend earlier, for the experiment of our Tamil word recognition scheme we collected 217 country names and the number of samples for each country name varies in between 20 and 40. In total there were 4270 samples collected from all the classes. These data were collected from individuals of different professionals like students, researchers, businessmen etc. The images were scanned at 300 DPI and stored in tagged image file (TIF) format.

B. Results computation method

We have used 5-fold cross validation scheme for recognition result computation. Here database was divided into 5 subsets and testing was done on each subset using other four subsets for training. The recognition rates for all the test subsets were averaged to calculate recognition accuracy.

For recognition result computation we used different measures and the measurers are defined as follows:

$$Recognition\ rate = \frac{N_C * 100}{N_T} \qquad (4)$$

$$Error\ rate = \frac{N_E * 100}{N_T} \qquad (5)$$

Where N_C is the number of correctly classified country-names, N_E is the number of misclassified country-names. Here $N_T = N_C + N_E$.

C. Global Results

We did not consider any rejection algorithm in the present scheme and we obtained overall 86.36% accuracy from system without any rejection. We observed classwise-accuracy of our system and noted that from 17 classes we obtained 100% accuracy. From 113 classes we obatined more than the overall accuracy (86.36%). From 188 classes we obatined more than 75% accuracy. Also minimum accuracy obtained from our system was 60.00% and this is only from one class. This lower

accuracy was obtained due to smaller number of samples of this class.

D. Recognition accuracy versus class sizes

To get idea about the results on different class sizes of country name, we also computed accuracies of our scheme based on different class sizes. From our experiment we got 91.67%, 89.44%, 88.43%, 86.41%, and 86.36% accuracy when we considered lexicon of sizes 50, 100, 150, 200, and 217 classes of country names, respectively. These results are graphically shown in Fig.10.

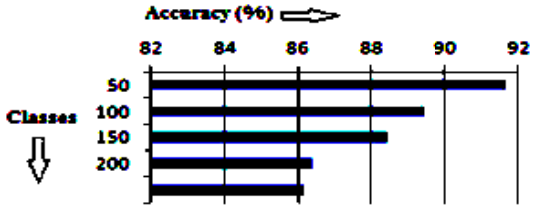


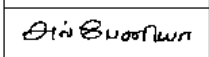
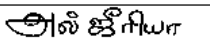
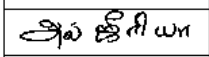
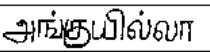
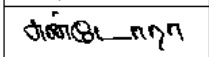
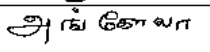
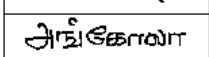
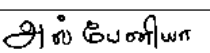
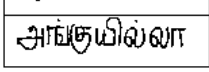
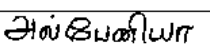
Figure 10. Graphical representation of accuracy vs. class numbers

E. Error Analysis and comparative results

From the experiment we noted that most of the errors are due to noise and small number of samples in the class. Some errors incurred due to slanting effect or due to horizontal touching. In future we plan to consider these cases.

To get the idea about the samples where our system generates erroneous results, we provide some of such samples in Table IV. Actual handwritten samples are shown in the first column of this table and their printed samples are shown in the respective rows of third column. Recognized class samples of the actual samples are shown in respective rows of the second column of the table. Since the actual handwritten class and its recognized class have many characters which are similar in shape, such miss-recognition occurred.

Table IV. Some Misclassified Sample

Actual Sample	Recognized Class	Printed Version of actual Sample
		அல்பேனியா
		அல்ஜீரியா
		அன்டோரா
		அங்கோலா
		அங்குயில்லா

To get an idea of comparative results of our system, we report a recent result here. Sigappi et al. [4] obtained 80.75% results when they considered only 40 word classes whereas we obtained 86.36% accuracy from the proposed system considering 217 word classes.

VI. Conclusion

In this paper, a holistic approach for identifying off-line Tamil handwritten words is proposed. A simple solution for detection of horizontal touching between letters was discused here. Similary detection of vertical touching within the letter is also proposed. A dataset of 217 country names was created for the experiment of the work. The data were collected from different group of people and in total we have 4270 samples. From the experiment we obtained 86.36% recognition accuracy. In future we plan to improve the system by analyzing the erroneous samples.

References

[1] R. Plamondon and S. N. Srihari. On-Line and off-line handwritten recognition: A comprehensive survey. IEEE Trans on PAMI, Vol.22, pp.62-84, 2000.

[2] J. Sutha and N. Ramaraj. Neural Network Based Offline Tamil Handwritten Character Recognition System: Int. Conf. on Comp. Intell.& Mult. Applications, Vol.2, pp.446-450, 2007.

[3] M. Antony Robert Raj and S. Abirami. A Survey on Tamil Handwritten Character Recognition using OCR Techniques: David C. Wyld, et al. (Eds): CCSEA, SEA, CLOUD, DKMP, CS & IT 05, Vol. 2, pp. 115–127, 2012.

[4] AN. Sigappi, S. Palanivel and V. Ramalingam. Handwritten Document Retrieval System for Tamil Language: Int. J. of Computer Applications , Volume 31, No.4,pages 42-47, 2011.

[5] U. Pal, K. Roy, and F. Kimura. A Lexicon Driven Method for Unconstrained Bangla Handwritten Word Recognition: IEICE Trans. Info. And Systems. Vol. E92-D, no. 5, 2009.

[6] C . Burges, "A Tutorial on support Vector machines for pattern recognition" Data mining and knowledge discovery, vol. 2, pp. 1-43, *1998*.

[7] V. Vapnik, "The Nature of Statistical Learning Theory" Springer Verlang, 1995.