SVM Based Scheme for Thai and English Script Identification

S. Chanda[1], Oriol Ramos Terrades[2] and U. Pal[1]
[1]Computer Vision and Pattern Recognition Unit, Indian Statistical Institute, Kolkata-108, India
Email: umapada@isical.ac.in
[2]Computre Vision Center, Universitat Autonoma de Barcelona, 08193, Barcelona, Spain
Email: oriolrt@cvc.uab.es

Abstract

In some Thai documents, a single text line of a document page may contain both Thai and English scripts. For the Optical Character Recognition (OCR) of such a document page it is better to identify, at first, Thai and English script portions and then to use individual OCR system of the respective scripts on these identified portions. In this paper, a SVM based method is proposed for identification of word-wise printed English and Thai scripts from a single line of a document page. Here, at first, the document is segmented into lines and then lines are segmented into character groups (words). In the proposed scheme, we identify the script of the individual character group combining different character features obtained from structural shape, profile, component overlapping information, topological properties, water reservoir concept etc. Based on the experiment on 6110 data we obtained 99.36% script identification accuracy from the proposed scheme.

1. Introduction

There are many Thai documents where a single document page contains both Thai and English texts. An example of such document is shown in Fig.1. For OCR development of such a document page, it is better to separate different scripts before feeding them to the OCRs of individual scripts [1]. In this paper, a SVM based technique is proposed for the identification of Thai and English scripts from a single document.

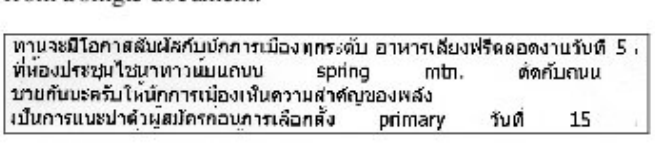

Fig.1 Example of a document containing both Thai and English.

There are many pieces of work on script identification [1-7]. Among them Spitz [2] developed a method for Han based or Latin based script separation. He used optical density distribution of characters and frequently occurring word shape characteristics for the purpose. Ding et al. [3] proposed a method for European and Oriental script identification. Using fractal-based texture features, Tan [4] described an automatic method for identification of Chinese, English, Greek, Russian, Malayalam and Persian text. Among Indian scripts, Pal and Chaudhuri [5] proposed a line-wise script identification scheme from Indian tri-lingual documents. Later, Pal et. al [6] proposed a generalized scheme for line-wise script identification from a single document containing twelve Indian scripts. Dhanya et al. [7] used Linear Support Vector Machine (LSVM), K-NN and Neural Network (NN) classifiers on Gabor-based and zoning features to classify Tamil and English scripts. Zhou et al. [12] proposed a Bangla/English script identification scheme using connected component analysis. Recently, Jaeger et al. [10] used K-NN, SVM, weighted Euclidean distance, and Gaussian mixture model to identify English from Arabic, Chinese, Korean and Hindi scripts.

There are many pieces of work on script identification [1] but there is no work to identify Thai and English scripts present in a single line. In this paper, a scheme for identification of different portions of Thai and English scripts from a single text line is presented and to the best of our knowledge this is the first work of its kind. Different features obtained from structural shape, profile, component overlapping information, topological properties, water reservoir concept etc. of the characters of a word (character group) are computed and fed to a SVM classifier for its identification.

2. Properties of Thai Script

Thai, a Tai-Kadai language is used by more than 25 million people in Thailand, the Midway Islands, Singapore, the UAE and the USA. Thai alphabet set consists of 44 basic consonants, 17 vowels, 4 tones and 2

punctuation marks. For some consonants there are multiple letters. Originally they represented separate sounds, but over the years the distinction between those sounds was lost and the letters were used instead to indicate tones. Thai character set is shown in Fig.2.

In Thai alphabet, consonants are divided into three classes and this division helps to determine the tone of a syllable. The sounds represented by some consonants change when they are used at the end of a syllable. Some consonants can only be used at the beginning of a syllable. Thai is a tonal language with 4 tones. The tone of a syllable is determined by a combination of the class of consonant, the type of syllable (open or closed), the tone marker and the length of the vowel.

Fig.2. Examples of Thai (a) vowels (b) consonant.

There is white space between two consecutive words in English text but there is not such white space between two consecutive words in Thai, instead spaces in a Thai text indicate the end of a clause or sentence. Thus word separation is difficult in Thai script. But if a single line contains both English and Thai text then generally there is enough space between English and Thai text. For example, see the 2nd and 4th lines of Fig.1. We use this space information for segmentation of text into character groups (words) and identify the script of these groups.

3. Preprocessing and features extraction

The digitized images are in gray tone and we have used a histogram based thresholding to convert them into binary images. The image is de-skewed if there is any skew. The digitized image may contain spurious noise pixels and irregularities on the boundary of the characters, leading to undesired effects on the system. For removing these noise pixels we have used a simple and efficient method due to [5]. The lines are segmented from the documents by finding the valleys of the horizontal projection profile computed by counting the number of black pixels in each row. The trough between two consecutive peaks in this profile denotes the boundary between two text lines. A text line can be found between two consecutive boundary lines. After a text line is segmented, it is scanned vertically. If in one vertical scan two or less black pixels are encountered then the scan is denoted by 0. Else, the scan is denoted 1. In this way a string of 0 and 1 is constructed. Now, if in the string there exist a run of 0's with minimum length $2*k_1$ then the midpoint of that run is considered as a boundary for segmentation of text into groups (words). The value of k_1 is taken as statistical mode of the white runs among the characters of a line obtained by its row-wise scanning. In other words k_1 is an estimation of white gap between two consecutive characters of a line. For individual character extraction from a word we mainly use vertical histogram. Component labeling is also used in some cases when two consecutive characters are *kerned* in nature and vertical segmentation is not possible (kerned characters are the characters who overlap with neighboring characters).

We will now discuss some of the features used in our script identification scheme.

3.1 Loop feature

Loop is an important property of Thai script. In general, the width of a loop is very small with compare to the character width in Thai script. Some English characters (for example, A, a, b, B, d, D, e, o, p, q, Q, R) have loop but the loops of English characters have larger width, which is at least half of the respective character width. Fig.3 illustrates the fact. We use this loop information as a feature of our scheme.

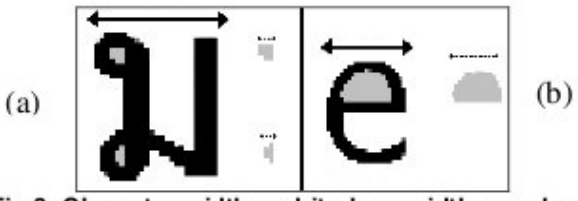

Fig.3. Character width and its loop width are shown in (a) Thai (b) English character. Character width is shown by thick line. Loops of the characters are marked in gray shade and the loop width is shown by dotted line.

3.2 Water reservoir principle based features

The water reservoir principle is as follows. If water is poured from a side of a component, the cavity regions of the component where water will be stored are considered as reservoirs [8].

Top (bottom) reservoir: When water is poured from top (bottom) side of the component, the cavity regions of the component where water will be stored are considered as top (bottom) reservoirs. (A bottom reservoir of a component is visualized as top reservoir when water will be poured from top after rotating the component by 180°).

Left (Right) reservoir: Left (right) reservoirs are obtained when water is poured from left (right) side of a component. For illustration see Fig.4. Here top, bottom, left and right reservoirs of some English characters are shown.

Water flow level: The level from which water overflows from a reservoir is called as water flow level of the reservoir (see Fig.5).

Reservoir base-line: A line passing through the deepest point of a reservoir and parallel to water flow level of the reservoir is called as reservoir base-line (see Fig.5).

Height of a reservoir: By height of a reservoir we mean the depth of water in the reservoir. In other words, height of a reservoir is the distance between reservoir base-line and water flow level of the reservoir (see Fig.5).

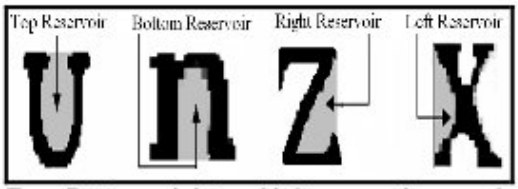

Fig. 4. Top, Bottom, right and left reservoirs are shown.

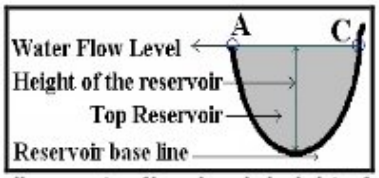

Fig.5. Base-line, water flow-level, height of a reservoir are shown in top reservoir.

Now we shall discuss some of the reservoir-based features used in the scheme.

Top reservoir can be found in many Thai characters. Also top reservoir can be obtained in some of the English characters like H, k, K, L, M, N, u, U, v, V, w, W, x, X, y, Y etc. We consider those top reservoirs whose heights are greater than 40% of the character height. The difference in the top reservoirs found in English and Thai characters are as follows:

(a) The water flow level in English characters mainly coincides with the character upper boundary, which is not true in most of the Thai characters. Sometimes for the English letters 'k' and 'L', water flow level of top reservoir does not coincide with upper boundary. In case of 'k' the reservoir base-line is almost at the middle of the character, and in case of 'L' height of right half of the character is very small compare to its left half. These are the notable distinguishing feature for their identification from Thai characters.

(b) Thai and English characters having top reservoir and water flow level coincides with upper boundary can mostly be distinguished by the following property. Thai characters have a small loop at the left upper part of the characters and this loop is absent in English characters. Loops in some of such Thai characters are shown in Fig.6.

Bottom reservoirs also play an important role in our identification scheme. There is a distinct difference between the shapes of the bottom reservoirs in some of the English and Thai characters. There is a definite sequence of reservoir width at different rows as we move towards the bottom of the reservoir starting from water flow level (by reservoir width of a particular row we mean the distance between two border points of that reservoir in that particular row). We use this width information for their identification. For English character the reservoir width at different rows will either remain similar or tends to decrease as we move upwards and this is not true in Thai. For illustration see Fig.7. Here shapes of bottom reservoir of one Thai and one English character are shown. Reservoir width decreases from A to B and again increases from B to C in the Thai character. For the English character reservoir width remains similar or decreases from A to C.

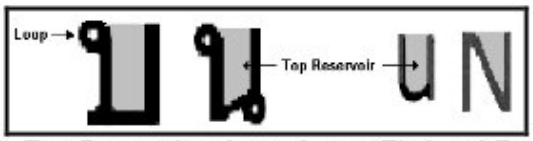

Fig.6. Top Reservoirs shown in two Thai and English Characters. Thai characters have loop and English characters do not have loop.

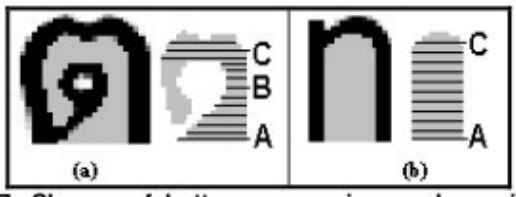

Fig. 7. Shapes of bottom reservoir are shown in (a) Thai (b) English character. Reservoir width of different rows is shown by horizontal lines.

3.3 Component overlapping feature

Component overlapping feature is another distinct feature between English and Thai texts and it is used for their identification. In English text, vertical overlapping of two components is absent except the characters 'i' and 'j' but in Thai there are many such situations where one component vertically overlaps with other component. For illustration see Fig.8. To compute this overlapping feature, we draw the bounding box of each component and check whether bounding boxes of two components vertically overlap or not. Two English characters 'i' and 'j' have vertical overlapping but these two characters have distinct behavior for their separation from Thai. For these English characters a vertical line-like structure is present in the position just bellow the upper component and ratio of the length of the two overlapping components (main part of 'i', and its dot) is more than four times the stroke width (R_L) of the character. This property identifies 'i' and 'j' from the Thai characters. R_L is the statistical mode of the black run lengths of the character. For a character, R_L is calculated as follows. The character is, at first, scanned row-wise (horizontally) and then column-wise (vertically). If n different runs of lengths $r_1, r_2,...,r_n$ with frequencies $f_1, f_2,.....f_n$, respectively are obtained by the scanning from the character, then value of R_L will be r_i if $f_i = \max(f_j)$, $j = 1, 2,.....n$.

Fig.8. Bounding box of each component is shown in (a) Thai (b) English text.

3.4 Rotated 'J' feature

In Thai there is a frequent occurring character 'ไ' (an image of this character is shown in Fig.9) which looks similar to the English character **J** when it is flipped vertically. Presence of this shape has been used as a feature and we check the presence of this particular character shape as follows. Suppose each character is located within a rectangular boundary, a frame. We compute the horizontal distance from left and right boundary of the frame to the character edge for each row within the frame, we call this distance as *left distance* and *right distance* respectively (left and right distance is shown in Fig.9). If we find that in a particular row the *left distance* is at least 70% of the character width and in that corresponding row the *right distance* is less than 30% of the character width then we say that row as *candidate row*. If for a component the number of such candidate rows is at least 65% of the character height and situated at the lower portion of the character, and upper part of the component is wider than its lower part then we assume that this shape is present. Different threshold values used to compute this feature are obtained from the experiment.

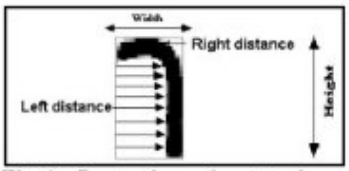

Fig.9. Detection of rotated 'J' feature

Fig.10. Illustrations of profile detection

3.5 Profile feature

We use presence of two or more vertical lines as a feature and profile feature [6] helps us to detect whether vertical-line like structure present in a character or not. In Thai there are many characters with two vertical lines (For example see the text shown in Fig.1). For detection of vertical-line like feature we compute profile of a character and we observe the behavior of the profiles. We compute the number of transitions (changing from increasing mode to decreasing mode or vice versa) in the profiles of the characters and use this value for feature extraction. If left (or right) profile has only zero transition (either are all increasing or all are decreasing mode) then we say that a vertical line exists in that character. If both in left and right profiles there are zero transitions then normally two vertical lines exist. For example see Fig.10, where left and right profiles of the italics character P is shown. Here all the left profiles in the character are in decreasing mode from top to bottom. Hence we assume that one vertical line exists in the character. Note that even if a character is italic this method can detect vertical line-like structures.

Based on these features we obtained a feature vector of dimension 7 from a word. The values of a feature vector lies between 0 and 1. To get this normalized value (between 0 and 1) for a particular feature (say, x) we proceed as follows. Let a word consists of N characters. If out these N characters, P characters have the feature x, then value of this feature x is P/N. We feed this normalized feature vector to our SVM classifier for identification.

4. SVM classifier for script identification

We use Support Vector Machine (SVM) classifier for script identification. The SVM is defined for two-class problem and it looks for the optimal hyper-plane which maximize 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:

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

where $\{x_j\}$ are the set of support vectors and the parameters α_j and b has been determined by solving a quadratic problem [9].

The linear SVM can be extended to a non-linear classifier by replacing the inner product between the input vector x and the SVs x_j, to a kernel function k defined as: $k(x, y) = \phi(x) \cdot \phi(y)$. This kernel function should satisfy the Mercer's Condition [11]. Some examples of kernel functions are polynomial kernels $(x \cdot y)^p$ and Gaussian kernels $\exp(-||x-y||^2/c)$, here c is a real number. Details of SVM can be found elsewhere [11] so we are not giving details of SVM here.

5. Results and discussions

The data used for the experiment are taken from printed document pages. These document pages are obtained from Thai newspaper, books and some printout of Thai literatures obtained from internet. We considered 6110 data (words) for our script identification result computation. The documents are scanned at 300 dpi.

We have used 5-fold cross validation scheme for recognition result calculation. Here database is divided into 5 subsets and testing is done on each subset using rest of the subsets for learning. The recognition rates for all the test subsets are averaged to get the accuracy. We tested our

results using Linear and Gaussian SVM provided in the OpenCV (http://sourceforge.net/projects/opencvlibrary) library. Detail results of Linear and Gaussian SVM using different values for the Gaussian parameter c (cf, Section 4) are given in Table 1. From linear SVM we obtained 96.89% accuracy and from Gaussian SVM we got 99.36% script identification accuracy. From the experimental results we computed standard deviation of the recognition accuracies obtained from 5 subsets. Standard deviation of the recognition accuracies obtained from linear SVM is 0.42 and standard deviation of the recognition accuracies obtained from Gaussian SVM are 0.29, 0.29, 0.17, 0.17, 0.17 and 0.17 when we consider c as 1.0, 2.0, 4.0, 8.0,16.0 and 32.0, respectively. From the Table 1 it can be seen that the choice of the parameter c does not significant influence in the accuracy of the Gaussian SVM. For different values of c the accuracy moves into a range between 99.26% (when c=1.0 and 2.0) and 99.36% (when c=16.0).

Table 1. Accuracy rates for Linear and Gaussian SVM on test data

Linear SVM	Gaussian SVM		
	c =1.0	c =2.0	c =4.0
96.89 ± .42%	99.26 ± .29%	99.26 ± .29%	99.34 ±.17%
	c =8.0	c =16.0	c =32.0
	99.35 ± .17%	99.36 ± .17%	99.30 ± .17%

Table 2. Results obtained in varying the size of test data

SVM	*Number of samples*			
	1500	***3000***	***4500***	***6110***
Linear	97.87%	97.77%	96.07%	96.89%
Gaussian, c=1.0	97.67%	98.57%	99.02%	99.26%
Gaussian, c=8.0	98.27%	98.80%	99.18%	99.35%
Gaussian, c=32.0	98.27%	98.80%	99.15%	99.30%

We also computed accuracy of our classifier using different size of data and the results are given in Table 2. From the table it can be seen that tendency of the accuracy rates for Linear and Gaussian SVM (c=1.0, 8.0 and 32.0). We can observe that for Gaussian SVM the accuracy rates is increasing with the number of samples, whereas for Linear SVM it is not true.

Since this is the first work on Thai-English script recognition, we cannot compare our results with other work on Thai-English script identification. However to get an idea about the recognition accuracy with other existing pieces of work, some comparison results are given in Table 3. From the table it can be seen that our proposed method gives better results than that of other published work.

Table 3. Comparison of results

Method proposed by	Script used	Classifier used	Accuracy obtained
Dhanya et al. [7]	Tamil-English	SVM	96.03%
Jaeger et al. [10]	Arabic-English	SVM	90.93%
	Chinese-English	SVM	93.43%
	Korean-English	K-NN	94.04%
	Hindi-English	K-NN	97.51%
Proposed method	Thai-English	SVM (Linear)	96.89%
		SVM(Gaussian)	99.36%

6. Conclusion

Script separation is very useful for multi-lingual and multi-script OCR development. In this paper we proposed a SVM based scheme for identification of word-wise printed English and Thai scripts from a single document page and in future we plan to explore other classifiers for the purpose. We tested our scheme on 6110 data and obtained 99.36% accuracy from the proposed scheme. To the best of our knowledge this is the first work on Thai and English script identification. The proposed work will not work properly on degraded and broken documents, which is the main drawback of the work.

References

[1] U. Pal, "Automatic Script Identification: A Survey", Vivek, vol.16, pp.26-35, 2006.

[2] A.L. Spitz, "Determination of the script and language content of document images", IEEE Trans on PAMI, vol.19, pp.235-245, 1997.

[3] J. Ding, L. Lam and C. Y. Suen, "Classification of oriental and European scripts by using characteristic features", In Proc. 4th ICDAR, pp.1023-1027, 1997.

[4] T. N. Tan, "Rotation invariant texture features and their use in automatic script identification", IEEE Trans. on PAMI, vol. 20, pp. 751-756, 1998.

[5] U. Pal and B. B. Chaudhuri, "Identification of different script lines from multi-script documents", Image and Vision computing, vol. 20, pp. 945-954 2002.

[6] U. Pal, S. Sinha and B. B. Chaudhuri, "Multi-Script Line identification from Indian documents", Proc. 7th ICDAR, pp.880-884, 2003.

[7] D. Dhanya, A. G. Ramakrishna and P. B. Pati, "Script identification in printed bilingual documents", Sadhana, vol.27, part-1, pp.73-82, 2002.

[8] U. Pal, A. Belaïd and Ch. Choisy "Touching numeral segmentation using water reservoir concept", Pattern Recognition Letters, vol.24, pp. 261-272, 2003.

[9] C. Burges, "A Tutorial on support Vector machines for pattern recognition", Dataming & Knowl. Discovery,vol.2, pp.1-43, 1998.

[10] S. Jaeger, H. Ma, and D. Doermann, "Identifying Script on Word-Level with Informational Confidence", In Proc. 8th ICDAR, pp.416-420, 2005.

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

[12] L. Zhou, Y. Lu and C. L. Tan, "Bangla/English script identification based on analysis of connected component profiles", In Proc. 7th DAS, pp. 243-254, 2006.