Open access • Proceedings Article • DOI:10.1109/ICDAR.2009.239

Two-stage Approach for Word-wise Script Identification — Source link

Sukalpa Chanda, Srikanta Pal, Katrin Franke, Umapada Pal

Institutions: Gjøvik University College

Published on: 26 Jul 2009 - International Conference on Document Analysis and Recognition

Topics: Feature (machine learning), Feature extraction, Optical character recognition and Contextual image classification

Two-stage Approach for Word-wise Script Identification

Sukalpa Chanda[1], Srikanta Pal[2], Katrin Franke[1], Umapada Pal[2]
Norwegian Information Security Lab,
Department of Computer Science and Media Technology, Gjøvik University College[1],
Gjøvik-2802, Norway
CVPR Unit, Indian Statistical Institute[2], 203 B.T. Road, Kolkata-700108, India
E-mail:- umapada@isical.ac.in

Abstract

A two-stage approach for word-wise identification of English (Roman), Devnagari and Bengali (Bangla) scripts is proposed. This approach balances the tradeoff between recognition accuracy and processing speed. The 1st stage allows identifying scripts with high speed, yet less accuracy when dealing with noisy data. The advanced 2nd stage processes only those samples that yield low recognition confidence in the first stage. For both stages a rough character segmentation is performed and features are computed on segmented character components. Features used in the 1st stage are a 64-dimensional chain-code-histogram feature, while 400-dimensional gradient features are used in the 2nd stage. Final classification of a word to a particular script is done via majority voting of each recognized character component of the word. Extensive experiments with various confidence scores were conducted and reported here. The overall recognition accuracy and speed is remarkable. Correct classification of 98.51% on 11,123 test words is achieved, even when the recognition-confidence is as high as 95% at both stages.

1. Introduction

In India, a single document page may contain words in two or more language scripts. So, multi-script Optical Character Recognition (OCR) is necessary to process these documents. For OCR development of such multi-script documents, it is necessary to separate different script words before feeding them to the OCRs of individual scripts. In this paper, an automatic technique for word-wise identification of English, Devnagari and Bengali scripts is proposed. Among the pieces of earlier work on script separation, Spitz [2] developed a method to separate Han-based or Latin-based script. He used optical density distribution of characters and frequently occurring word shape characteristics. Jaeger et al. [14] used K-NN, SVM, weighted Euclidean distance, and Gaussian mixture model to identify English from Arabic, Chinese, Korean and Hindi scripts Using cluster-based templates; an automatic script identification technique has been described by Hochberg *et al.* [7]. Using fractal-based texture features, Tan [3] described an automatic method for identification of Chinese, English, Greek, Russian, Bengali and Persian text. All the above mentioned works deal with Non-Indian script. Among Indian script, there are some related works. Pal et al.[4] proposed a generalized scheme for line-wise script identification from a single document containing twelve Indian scripts. Sinha et al. [13] proposed a word-wise script identification scheme with a combination of Indian languages. Dhanya and Ramakrishnan [5] proposed a Gabor-filter-based technique for word-wise segmentation from bi-lingual documents containing English and Tamil scripts, they have used classifiers like LSVM and K-NN. Patil and Subbareddy [8] proposed a neural-network-based system for word-wise identification of English, Hindi and Kannada language scripts. Zhou et al. [6] proposed a Bengali/English script-identification scheme using connected component analysis. In this paper a technique is proposed for word-wise script identification from Indian documents containing English, Devnagari and Bengali. In the proposed system individual scripts are identified using combination of two different features. In the first processing stage, 64-dimensional chain-code features are used for classification; the rejected samples of this stage are further processed with 400-dimensional gradient feature for classification at second stage. Final classification of a word to a particular script is done via majority voting of each recognized character component of the word.

The organization of the paper is as follows. Pre-processing like text digitization, line and word segmentation etc. are described in Section 2. Different

978-0-7695-3725-2/09 $25.00 © 2009 IEEE
DOI 10.1109/ICDAR.2009.239

features used in the identification scheme are discussed in Section 3. Classifier details are given in Section 4. Experimental results and discussions are provided in Section 5 followed by conclusion and reference in section 6 and 7 respectively. We intentionally skip discussion about English, Devnagari and Bengali script characteristics as they can be found in [1] and also www.omniglot.com.

2 Preprocessing

2.1. Digitization and noise cleaning

The documents are digitized by a HP scanner at 300 DPI. The digitized images are in gray tone and we have used a histogram-based thresh-holding approach to convert them into two-tone images (0 and 1). The digitized image may contain spurious noise pixels and irregularities on the boundary of the characters, leading to undesired effects on the system. We smoothed the image to remove some of the above noises [4]. Utmost care was taken to de-skew the digitized images [1].

2.2. Line and word segmentation

The lines of a text block are segmented by finding the valleys of the 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. For word segmentation a text line is scanned vertically. If in one vertical scan two or less object pixels are encountered then the scan is labeled "0", else the scan is labeled by the number of object pixels in that column. In this way a vertical scanning histogram is constructed. Now, if there is a run of at least K_1 consecutive 0's in the histogram then the midpoint of that run is considered as the boundary of a word. The value of K_1 is set as 4 times of stroke width (R_L). Stroke width (R_L) is the statistical mode of the black run lengths of the components of the text portion to be identified. R_L is calculated as follows. Text portion 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 after these two scanning of the text portion, then value of R_L will be r_i if $f_i = \max(f_j)$, j = 1, 2,…..n.

2.3. Character segmentation

After getting the word boundary we look for head-line feature in the word (see section 4). This feature is totally absent in English words but present in high frequency amongst the other two scripts. Though getting head-line feature in a Devnagari or Bengali word is not obvious always. If we cannot find a head-line in a word, it signifies that the characters are already isolated. Hence, we directly apply feature extraction procedures on individual character components in the word. If we get a head-line in a word it indicates that the characters are connected to each other. Then we try to segment the word as discussed below: A vertical scanning for each column is performed. We start vertical scanning from a row which is at R_L distance below the head-line and the scanning for each column continues till the bottom most part of the word. While scanning a vertical column, if we encounter less than (R_L) number of black pixel, we call that column as a candidate column. Consecutive candidate columns are found and the middle of the candidate columns is chosen as the segmentation line. We remove all black pixels along the segmentation line and the characters get separated from each other. Then those separated characters are passed on to our feature extraction modules. See the figure below where character segmentation in one Bengali and one Devnagari word has been shown. The vertical marks labeled as C denotes the segmentation line.

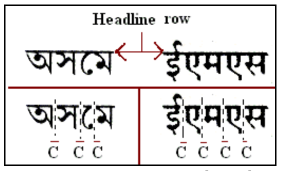

Figure 1. Character segmentation in a Bengali and Devnagari word are shown.

3. Feature extraction

3.1. Head-line feature

If we take the longest horizontal run of black pixels on the rows of a text line then such run length for Devnagari and Bengali script will be much higher than that of English scripts. This is because, most of the time the characters in a word are connected by head-line in Devnagari and Bengali script. We say head-line feature exists in a word if one of the following two conditions satisfies in the word (a) If the length of the longest run is greater than 70% of the width of a word (b) if the length of the longest run is greater than 2 times of the height of middle zone (x-height). x-height is shown with dotted line in Fig.2.

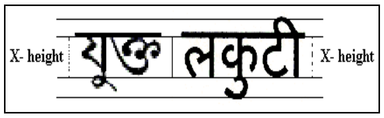

Figure 2.The X-height is shown with a dotted line in a Bengali and a Devnagari word.

3.2. 64-dimensional chain-code histogram

In order to compute this feature for a component, at first the bounding box of a component is segmented into few blocks. Then a chain-code histogram is computed in each of the blocks. Features are mainly based on the direction chain-code histogram of the contour points of these blocks. Details about 64-dimensional feature extraction technique used in this paper is available at [10], hence we are not providing it here.

3.3. 400-dimensional gradient feature

To obtain 400-dimensional features [10] we apply the following steps. (i) The input binary image is converted into a gray-scale image applying a 2×2 mean filtering *5* times. (ii) The gray-scale image is normalized so that the mean gray scale becomes zero with maximum value 1. (iii) Normalized image is than segmented into 9×9 blocks. (iv) A Roberts filter is then applied on the image to obtain gradient image. The arc tangent of the gradient (direction of gradient) is quantized into 16 directions and the strength of the gradient is accumulated with each of the quantized direction. By strength of Gradient ($f(x,y)$) we mean $f(x,y) = \sqrt{(\Delta u)^2 + (\Delta v)^2}$ and by direction of gradient ($\theta(x,y)$) we mean $\theta(x,y) = \tan^{-1}\frac{\Delta v}{\Delta u}$, where $\Delta u = g(x+1, y+1) - g(x,y)$ and $\Delta v = g(x+1, y) - g(x, y+1)$ and $g(x,y)$ is a gray scale at (x, y) point. (v) Histograms of the values of 16 quantized directions are computed in each of 9×9 blocks. (vi) 9×9 blocks is down sampled into 5×5 by a Gaussian filter. Thus, we get $5 \times 5 \times 16 = 400$-dimensional feature.

4. Classifier Details

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:

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

where $\{x_j\}$ are the set of support vectors and the parameters α_j and b has been determined by solving a quadratic problem [12]. The linear SVM can be extended to various non-linear variants, details can be found in [11][12]. In our experiments 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:

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

We noticed that Gaussian kernel gave highest accuracy when the value of its gamma parameter is 42.00 and the penalty multiplier parameter is set to 100.

5. Results and Experimental Details

We applied our identification scheme on 3299 English, 3875 Bengali and 3949 Devnagari test words respectively, which sums up to 11,123 words. The training of the system has been done with a different set of 3000 words from each script respectively. The images were scanned from newspaper, magazine, story books, etc. We noticed during experimentation that our 64-dimensional chain-code-histogram feature computation is fast, but we were unable to properly classify a large section of our test words, especially those which are noisy or degraded by some other means of image quality. To overcome this, we replace our 64-dimensional feature by a 400-dimensional gradient-based feature. In effect, the accuracy was increased but we had to sacrifice our computational speed. We then took a two-stage approach of feature extraction in order to balance the tradeoff between computational speed and accuracy. To achieve this goal, in stage 1 we compute chain-code histogram feature on the test samples. We pass the feature to SVM for identification task. Some samples got correctly identified. But the samples which are rejected by the classifier at this stage are passed on to the second stage. In stage 2, we compute 400-dimensional gradient- based feature on the rejected samples of stage 1. We noticed that most of the rejected samples of stage 1 got correctly identified in stage 2. We also noticed that the two-stage approach hugely outperformed the one stage approach in terms of computational speed, and high recognition accuracy. Though the time taken by the two-stage approach is a bit more compared to the one stage approach using chain-code feature. But considering the high accuracy rate we can ignore this small time factor. The details of our experiments are given in following sub-sections. The first two sub-sections are on our experimental details on our single stage approaches, with chain-code-histogram feature and gradient-based features respectively. The last sub-section is about the experimental details of our 2-stage combined approach.

5.1. Experimental Frame work

The following steps are common to all three of our experiments. (a) First, we try to identify the head-line feature in a word. (b) If there is no head-line in a word, we directly move to step d. (c) If there is a head-line in a word, we do rough character segmentation within that word. This entire process is executed using the technique discussed in section 2. Eventually we get isolated character components in the word. (d) Features are extracted from each segmented character components of a word. (e) Features are sent to the classifier and the classifier identifies the script of each of those character components. (f) The character components which do not satisfy the confidence score threshold during the recognition process are considered as rejected character components. (g) We classify a word to a particular script on the basis of majority voting of each recognized character component. (h) We reject a word sample if sufficient character components in a word gets satisfactory confidence score value or our majority voting scheme ends up in a tie. In our two-stage approach, those rejected words are passed to stage 2 for gradient-based feature extraction and classification. By confidence score value of recognition we mean to say the probability estimation of the recognized class [9].

5.1.1. Results on Chain-code histogram feature. In Table 1, at each column we report the accuracy of our system with only 64-dimensional chain-code histogram-based features, when confidence score of the top choice recognition label is as high as 95%, 90%, 80%, 70%, 60%, 50% respectively. For example, the accuracy of 91.19% in column 1 is achieved by the following steps: Within a word, the character components with top-choice-confidence score 95 or more are considered as correct identification. Then classification of a word to a particular script is done via majority voting of each of the correctly identified character components. We reject a word sample if no character component in a word gets satisfactory confidence score value. Eventually, in this manner we were able to correctly identify the script of 91.19% word samples out of our total test set.

5.1.2. Results on Gradient feature. In Table 2, at each column we report the accuracy of our system with only 400-dimensional gradient-based features. The accuracy of the system with respective confidence score are shown. The words are classified with the help of majority voting scheme as discussed in the previous section.

Table 1. Script identification rate on different confidence score with 64-dimensional feature.

	Accuracy on diff. confidence scores					
	95	**90**	**80**	**70**	**60**	**50**
Accuracy Time in Sec.	91.19 306 s	93.70 305 s	95.41 302 s	96.49 302 s	96.80 304 s	97.13 305 s

Table 2. Script identification rate on different confidence score with 400-dimensional feature.

	Accuracy on diff. confidence scores					
	95	**90**	**80**	**70**	**60**	**50**
Accuracy Time in Sec.	95.49 1208 s	96.38 1207 s	97.42 1208 s	98.11 1210 s	98.43 1211 s	98.68 1209 s

5.1.3. Experiment Results on our two-stage approach. In Table 3, at each column we report the accuracy of our combined two-stage approach. At stage 1, 64-dimensional feature extraction is executed for all the character components in a word. The features are sent to classifier and characters components are identified to be of any of the three scripts. Based on the majority voting of the identified character components we classify the script of the word. The rejected word samples were sent to stage 2. In stage 2 we perform gradient-based feature extraction on each character components of the rejected word. The features are sent to classifier and each character component is identified to a particular script. Then, here also script identification of a word is done via majority voting of each of those correctly identified character components in the word. Note that we have maintained respective top choice confidence score threshold, for all character components in a particular word at both stages. Overall accuracy of the 2-stage approach is given in the last row of the Table 3. From the tables we can easily conclude that the combined approach worked well in terms of speed and accuracy while compared with the normal one stage approaches here. Overall accuracy for each confidence level is computed as follows: Overall accuracy=Accuracy of 64 dimensional feature + Accuracy of 400 dimensional feature on rejected samples of stage 1 with 64 dimensional feature *(100-Accuracy of 64 dimensional)/100.

5.2. Error Analysis

From the experiments, we noticed that most of the errors came out of poor and noisy images and for words where the number of characters in the word is less than three. Also we noticed that the miss-classification is mostly between Devnagari and Bengali scripts. See the figure below where the middle Bengali word got misclassified to Devnagari. Please note that the middle character of the Bengali word এম.এ. is broken. Also as no head-line was detected in the word, no processing for character segmentation is performed. The first two characters from left are connected by a small horizontal line, which makes it a single component. We extract feature from two separate components in the word. The unusual shape of the left most component in the word contributed the most to get it misclassified.

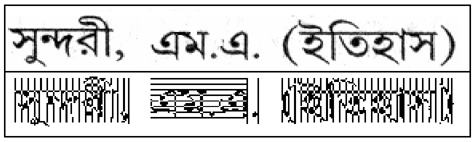

Figure 3. Words recognized as Devnagari (Bengali) are marked with horizontal (vertical) lines.

Table 3. Script identification rate on different confidence score on two-stage approach.

	Accuracy on diff. confidence scores					
Feature	**95**	**90**	**80**	**70**	**60**	**50**
64 Dim.	91.19	93.70	95.41	96.49	96.80	97.13
400 Dim.	83.20	83.28	85.19	87.03	87.85	88.51
Accuracy Time in Sec.	98.51 545 s	98.94 505 s	99.32 496 s	99.54 481 s	99.60 469 s	99.67 467 s

5.3. Comparison with previous similar works

We can have an idea about the recognition accuracy of other similar existing pieces of work on script identification. Zhou et al. [6] developed a rule-based system for identification of Bengali and English script on postal envelops, where they have used connected component profile as the feature. They obtained an accuracy of 99% (95%) for printed (handwritten) documents. Dhanya et al. [5] tried to identify Tamil and English scripts and obtained accuracy of 96.03%. Similar works on other Indian scripts with English has been reported in [13] where the system accuracy for Identification between {Devnagari, English} is 97.14% and {Bengali, English} is 98.30%. Overall accuracy of our scheme is 98.51%. But note that here we consider three scripts simultaneously.

6. Conclusion

Script identification is very important for development of multi-script OCR systems. In this paper, we propose a SVM based scheme for identification of word-wise printed English, Devnagari and Bengali scripts from a single document page. We tested our scheme on 11,123 word samples and obtained 98.51% accuracy from the proposed scheme. The next step would be to build up OCR modules for each of these individual scripts, and develop a complete multi-script OCR system for these three scripts.

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

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

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

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

[7] J. Hochberg, P Kelly, T Thomas and L. Kerns, "Automatic script identification from document images using cluster-based templates", IEEE Trans. on PAMI, vol. 19, pp. 176-181, 1997.

[8] S. B. Patil and N. V. Subareddy, "Neural network based system for script identification in Indian scripts", Sadhana, vol.27, part-1, 83-97, 2002.

[9] T. F. Wu, C. J. Lin, and R. C. Weng. "Probability Estimates for Multi-class Classification by Pairwise Coupling". Journal of Machine Learning Research, vol. 5, pp. 975-1005, 2004.

[10] U. Pal, N. Sharma, T. Wakabayashi, F. Kimura, "Handwritten Numeral Recognition of Six Popular Indian Scripts". In Proc. 9th ICDAR 2007, pp. 749-753.

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

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

[13] S. Sinha, U. Pal and B. B. Chaudhuri, "Word-wise Identification from Indian documents", Lecture Notes on Computer Science (LNCS-3136), Eds. S. Marinai and A. Dengel, pp.310-321, 2004.

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