Handwritten Word Spotting in Indic Scripts using Foreground and Background Information

Ayan Das
Dept. of ECE,
IEM Kolkata, India
das.ayan.iem@gmail.com

Ayan Kumar Bhunia
Dept. of ECE,
IEM Kolkata, India
ayanbhunia007@gmail.com

Partha Pratim Roy
Dept. of CSE
IIT Roorkee, India
proy.fcs@iitr.ac.in

Umapada Pal
CVPR Unit, ISI
Kolkata, India
umapada@isical.ac.in

Abstract

In this paper we present a line based word spotting system based on Hidden Markov Model for offline Indic scripts such as Bangla (Bengali) and Devanagari. We propose a novel approach of combining foreground and background information of text line images for keyword-spotting by character filler models. The candidate keywords are searched from a line without segmenting character or words. A significant improvement in performance is noted by using both foreground and background information than anyone alone. Pyramid Histogram of Oriented Gradient (PHOG) feature has been used in our word spotting framework and it outperforms other existing features of word spotting. The framework of combining foreground and background information has been evaluated in IAM dataset (English script) to show the robustness of the proposed approach.

1. Introduction

Handwritten text recognition is still one of the challenging problems in the field of pattern recognition. Due to the free-flow nature of handwriting and many writing variations, the recognition performance is not satisfactory even with sophisticated pre-processing and OCR techniques. A special form of word recognition technique, the so called “Word Spotting” has been proposed [3] that does not require OCR of the entire document. Word spotting has been extensively studied [1, 2, 4, 5] to detect a word in a handwritten document page (or line) as per the user’s query keyword [5] or a template image [10, 13]. This searching or browsing approach in a fast way often overcomes the problem of conventional recognition. Text search using word spotting techniques [3] provides an alternative approach for indexing and retrieval. As a result, it has been popular in extracting information from historical documents, handwritten forms, etc.

Word spotting with Query By Example (QBE) principle takes instances of query word image for searching. Whereas Query By Text (QBT) [15] which uses learning based approach for retrieval proved more effective recently. This paper presents a Query-By-Text based word spotting system in segmented text lines using Hidden Markov Models. We propose here a novel approach of combining foreground and background information of text images for keyword-spotting by character filler models. The candidate keywords are searched in line without segmenting character or words. A significant improvement in performance is noted by using both foreground and background information than anyone alone. The framework is applied to Indic scripts such as Bengali and Devanagari along with Latin script for evaluation.

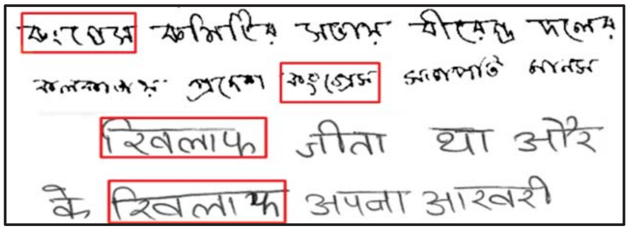


Fig.1. Examples of word spotting in Bangla and Devanagari script. First two Bengali lines were searched with the keyword “কংগ্রেস” and last two Devanagari lines were searched with the keyword "खिलाफ".

2. Related Work

Handwritten word spotting [8] is traditionally viewed as an image matching task between one or multiple query word images and a set of candidate word images in a database [3, 4]. The techniques of query by example (QBE) or image template matching [10] was adopted by researchers in the early days of word spotting. The modern approach namely query by text (QBT) or the learning based approach [12, 13] which outperformed the older one, is being extensively used in recent systems also. Some work exists in which character template based [13] spotting has been considered whereas others depict spotting at word level. Several works exist towards the application of word spotting such as keyword finding in historical documents [4, 13], searching and browsing through a digitized document, etc. A script independent word spotting method has also been proposed recently [2]. In retrieval of important information from poorly written old documents [3], word spotting has been considered. Several local features have been used for achieving better

978-1-4799-6100-9/15/$31.00 ©2015 IEEE

performance among which some outperformed others in conjunction with Dynamic Time Wrapping (DTW). In another kind of approach, keyword spotting has been done at character level using BLSTM-NN (Bidirectional long short-term memory neural network) [9]. There exists several page level segmentation free techniques which uses scale invariant features (i.e. SIFT) [10]. Recently, Fischer et al. [5] has described the word spotting performance using character filler model using Marti-Bunke feature.

The contributions of this paper are the following: 1) A unique feature extraction method for word spotting has been performed using combination of foreground and background information, 2) the frame work for word spotting has been analyzed in Indic scripts namely Bangla and Devanagari. 3) The system has been tested in IAM dataset of English to ensure the robustness of our approach. A comparative study between PHOG and LGH feature has been performed to evaluate their performance in word-spotting for Indic script.

The rest of the paper is organized as follows. The word spotting framework is explained in details in Section II. We have demonstrated the performance of our novel feature extraction for word spotting in Section III. Finally, conclusions and future work are presented.

3. Proposed Approach on Word Spotting

The major goal of word spotting is to detect specific keyword in a pool of document images. Our system is able to search arbitrary words in the text lines. For this purpose, the document image is first binarized with a global binarization method. Next, the binary document is segmented into individual text lines using a line segmentation algorithm [6]. For skew-correction, we consider all the points at the extreme bottom of the text stroke and use *Linear Regression* analysis on these points to find out the best fitted line. The slope of the straight line δ represents skew of the text. Thereafter, a rotation by δ is done to correct the skew. After skew correction each text line is normalized to cope up with different handwriting style.

Fig.2. provides the graphical description of the word spotting framework where concatenated features are fed to HMM. Word spotting is being performed using text line scoring based on the filler and character model of HMM.For the word spotting system we have used a novel feature extraction technique. Concatenation of foreground feature and background features are considered here. The details of each step are described below.

3.1. Feature extraction

Feature is a representation of an image which is more discriminative than the image. PHOG feature has been found to provide better result in Bangla handwritten script recognition [7]. PHOG [12] is the spatial shape descriptor which gives the feature of the image by spatial layout and local shape, comprising of gradient orientation at each pyramid resolution level. To extract the feature from each sliding window, we have divided it into cells at several pyramid level. The grid has 4^N individual cells at N resolution level (i.e. N=0, 1,2..). Histogram of gradient orientation of each pixel is calculated from these individual cells and is quantized into L bins. Each bin indicates a particular octant in the angular radian space.The concatenation of all feature vectors at each pyramid resolution level gives 168 dimensional feature vectors considering 8 bins and limiting the level to N=2 in our implementation.

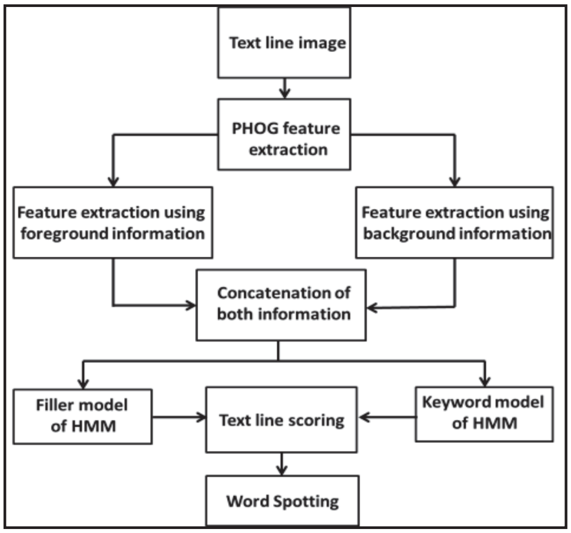

Text line image
PHOG feature extraction
Feature extraction using foreground information
Feature extraction using background information
Concatenation of both information
Filler model of HMM
Text line scoring
Keyword model of HMM
Word Spotting

Fig.2. Proposed word spotting framework

For calculating background information we take care of the morphology of character set in Bangla and Devanagari scripts. In Bangla or Devanagari script it is noted that most of the characters have a horizontal line (Shirorekha) at the upper part. When two or more characters sit side by side to form a word, the horizontal lines of the characters touch and generate a long line called head-line. Because of such touching nature the characters in a word create big white regions (spaces) in Bangla or Devanagari scripts. These empty spaces are found by water reservoir principle [11]. For each pair of joining characters we will get unique reservoir formation, these reservoirs contain information about the combination of characters forming the word. In Fig.3 the formation of bottom reservoirs are shown for Devanagari and Bangla text line, respectively.

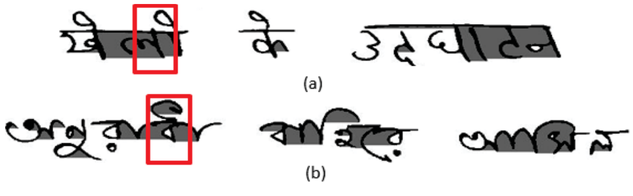


Fig.3. Water Reservoir formation in (a) Devanagari and (b) Bangla text line image and position of sliding window is marked in red color.

We have calculated PHOG feature from foreground as well as background regions, formed by the reservoir. These features are then concatenatedfor the final feature from the text line image. An illustration of feature extraction technique is given in Fig. 4.

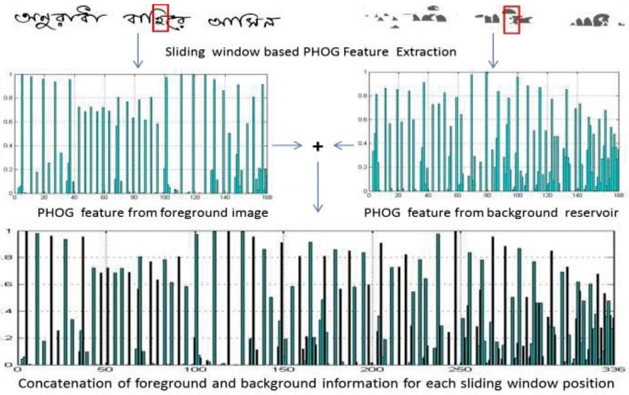


Fig.4. Feature extraction method shown graphically. The features are extracted from the sliding window marked in red color.

3.2. Hidden Markov Model

In the field of handwritten text recognition, Hidden Markov Models have been extensively used because of its peculiar nature of being efficient at recognition in the cases of touching characters, distorted characters even without being properly preprocessed [14]. The simplest model is the character HMM which consists of J hidden states (S_1, S_2 ... S_J) in a linear topology as an observation O where i^th observation (O_i) represents an n-dimensional feature vector **x**modeled using a Gaussian Mixture Model (GMM) with probability$P_{S_j}(\boldsymbol{x})$, 1<j<J given by

$$\boldsymbol{P_{S_j}(x) = \sum_{k=1}^{G} W_{jk}\, N(x|\mu_{jk}, \Sigma_{jk})}$$

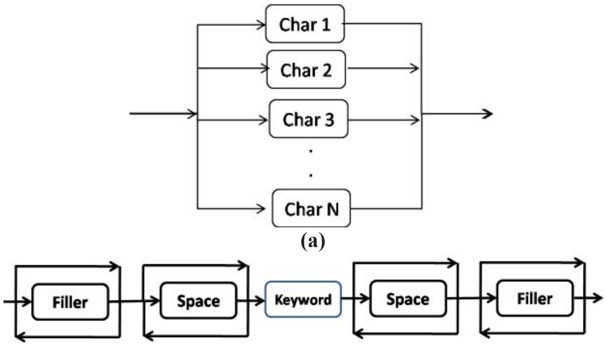
Where G is the number of Gaussians and***N***refers to a multivariate Gaussian distribution with mean $\boldsymbol{\mu_{jk}}$, covariance matrix $\boldsymbol{\Sigma_{jk}}$and probability W_{jk} for k^th Gaussian in state j**.**

For training the model, firstly, feature vectors (different features have been considered separately) have been extracted from labeled text line images with multiple words. The probability of the character model of the text line is then maximized by Baum-Welch algorithm assuming an initial output and transitional probabilities.Using the character HMM models, a filler model has been created which is shown in Fig.5(a). Fig.5(b) shows the keyword model which has been used in our system to spot a keyword in a text line image. The filler model represents a single character model consisting of any characters among 'Char i's, where 1 ≤ i ≤N (see Fig.5(a)). A 'Space' model has been used in the keyword model shown in Fig. 5.(b) which is accounted for modeling white spaces.

3.3. Text line scoring

Word spotting mechanism is based on the scoring of text image(X) for the keyword (W). If the score value is greater than a certain threshold then it gives a positive value for the occurrence of that particular keyword in that text line.The score assigned to the text line image X for the keyword W is based on the posterior probability $P(W_j|X_{a,b})$ trained on keyword models.Where a and b correspond to starting and ending position of the keyword whereas $X_{a,b}$ gives the particular part of text line containing the keyword [5]. Applying Bayes' rule we get $\log p(W|X_{a,b}) = \log p(X_{a,b}\,|W) + \log p(W) - \log p(X_{a,b})$

Considering equal probability we can ignore the term $\log p(W)$. The term $\log p(X_{a,b}\,|W)$ represents the keyword text line model where it is assumed that exact character sequence of the keyword to be present separated by 'Space'. The rest part of the text line is modeled with Filler text line model. Then we can find the position a, b for the keyword alongside with the log-likelihood$\log p(X_{a,b}\,|W) = \log p(X_{a,b}\,|K)$.



(b)

Fig.5. (a) Filler Model and (b) Keyword Model

$\log p(X_{a,b})$ is the unconstrained filler model F.The general conformance of the textimage to the trained character models is given by obtained log-likelihood $\log p(X_{a,b}) = \log p\big(X_{a,b}|F\big)$.The difference between the log-likelihood value of keyword model and filler model is normalized with respect to the length of the word to get the final text line score.

$$Score(X,W) = \frac{[\,\log \mathrm{p(Xa,b\mid K)} - \log \mathrm{p(Xa,b\mid F)}]}{b-a}$$

Then this $Score(X,W)$ is compared with a certain threshold for word spotting.

4. Experiment Results

We have collected document images written by different professional. The input of our system can be either arbitrary keyword string or text line image. We have collected word images of different writer for both Bangla and Devanagari script. Then we have generated a total of 8592 line images for Bangla and7902 line images for Devanagari; both containing two to six word images in a line. We have also used IAM (English) dataset. The details of data used in our experiment are shown in Table 1. Some of the keywords used for word spotting in three scripts are shown in Table 2. Some qualitative results in three scripts are shown in Fig.6. Note that the system is efficient with the handwriting variability and space between characters in a word.

Table.1.The dataset used for the experiment

	Bangla	Devanagari	IAM(English)
Training	6824	6214	6029
Validation	854	810	822
Testing	914	878	875
Keywords	671	621	821

Table 2.Some examples of keywords

Bangla	Devanagari	IAM (English)
লাগাতার	खिलाफ	being
ফেরিওয়ালা	परियोजना	House
মতপার্থক্য	भेजने	Government
মহানগরী	शिकायत	People
শশীকর	अधिकतर	would
সংবাদদাতা	गोलाबारी	Should

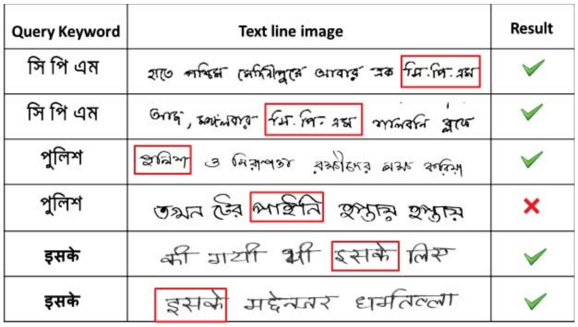

Query Keyword
Text line image
Result
সি পি এম
সি পি এম
পুলিশ
পুলিশ
इसके
इसके
✓
✓
✓
✗
✓
✓

Fig.6. Qualitative results of few word spotting instances where spotted words are indicated by red boxes and results are given by correct (by tick) and incorrect (by cross) labels.

We have measured the performance of our word spotting system using precision, recall and mean average precision (MAP). The precision and recall are defined as follows.

$$Precision = \frac{TP}{TP + FN} Recall = \frac{TP}{TP + FP}$$

Where, TP is true positive, FN is false negative and FP is false positive. MAP value is evaluated by the area under the curve of recall and precision.

For our experiment, 32 Gaussian mixture and 6 states provided optimum results. We have evaluated the performance for word spotting considering local threshold and global threshold, both in Fig. 7. For local threshold single image has been considered for optimization of the threshold, value whereas a standard value has been used for all query keywords in case of global threshold. We have considered a total of 200 keywords for our word spotting performance evaluation in Bangla and Devanagari scripts.

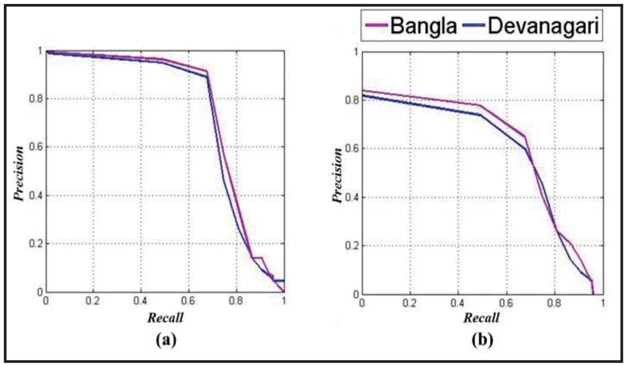


Fig.7. Word Spotting Performance taking (a) local threshold and (b) global threshold (using foreground information only).

A comparative evaluation is shown in Fig.8 for combination of foreground and background information with only foreground and only background information. It has been observed that there is a significant improvement in the word spotting performance using our combined feature extraction method.

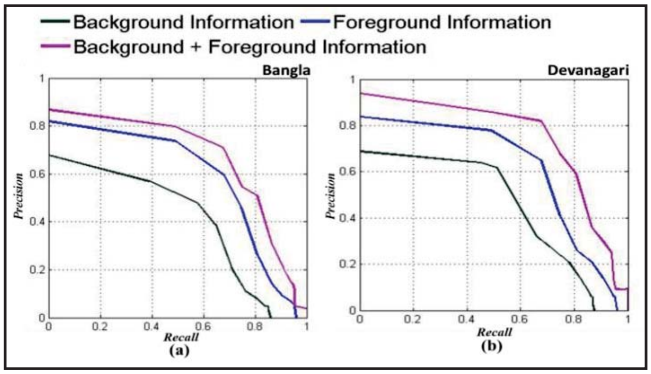


Fig.8. Comparative study of word spotting performance on (a) Bangla (b) Devanagari and using concatenation of foreground and background information with foreground or background information alone using global threshold.

Also we have checked the result using different number of keywords in our dataset, considering global threshold, using concatenation of foreground and background features. The results are shown in Fig. 9 based on the precision recall curve.

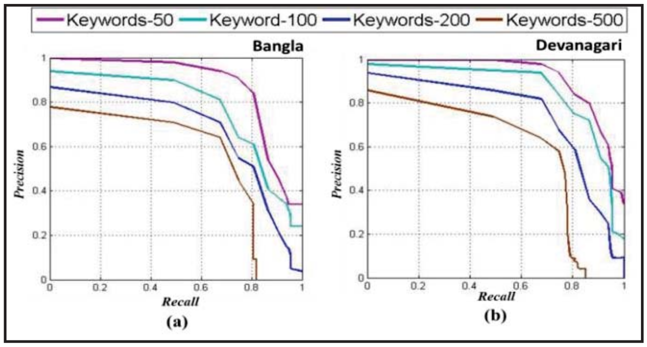


Fig.9. Comparative study of word spotting performance on (a) Bangla and (b) Devanagari scripts with different number of keywords

A comparative study of 2 different features, namely LGH and PHOG, is also performed to check the efficiency of our feature extraction approach by concatenating foreground and background information.LGH feature [15], which was found to be useful in Latin text, also gives a close accuracy to PHOG feature. In LGH features, with 8 bins we found 128 dimensional feature vector for each sliding window position. Word spotting performance is slightly found to be better using PHOG feature than LGH feature. MAP value has been given in Table 3. Our proposed word spotting framework has been testedonthe IAM dataset of English script. A detailed analysis of the results on IAM dataset is shown in Table 4.

Table.3. Comparison of LGH and PHOG features in word spotting using global threshold

Script	LGH	PHOG
Bangla	51.84	52.64
Devanagari	52.55	53.71
IAM (English)	48.04	48.98

Table.4. Analysis in IAM dataset for English

Feature	MAP (Local)	MAP (Global)
Foreground Information	69.58	48.98
Background Information	44.28	32.19
Foreground + Background	72.28	51.87

We have also checked the performance using keyword of different lengths considering global threshold which is shown in Fig.10. The computation time for the occurrence of given keyword in a particular text line is 1.34 seconds for Bangla and 1.39 seconds for Devanagari using Intel(R) Pentium(R) CPU(2.80 GHz) and 4 GB RAM.

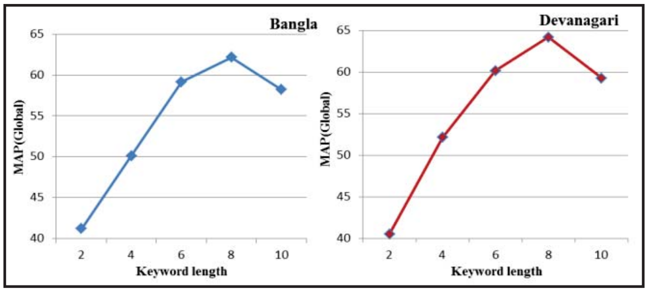


Fig.10.Word spotting performance using keywords of different length

5. Conclusion and future work

In this paper we have proposed a novel feature extraction method combining foreground and background features for word spotting. We noted that PHOGfeature outperformed the LGH feature for word spotting performance. Line level word spotting gives better performance than word segmenting approach where the gap between two consecutive words is not regular. In future we shall work on time efficient approach for word spotting.

References

[1] A. Vinciarelli, “A survey on off-line cursive word recognition”, Pattern Recognition, Vol. 35 (7), pp. 1433–1446, 2002.

[2] S. Wshah, G. Kumar and V. Govindaraju, “Script independent word spotting in offline handwritten documents based on hidden markov models”, In Proc. ICFHR,pp. 14-19, 2012.

[3] T. M. Rath and R. Manmatha, “Word spotting for historical documents”, International Journal of Document Analysis and Recognition, Vol. 9(2),pp. 139–152, 2007.

[4] Y. Leydier, A. Ouji, F. Le-Bourgeois and H. Emptoz, “Towards an omni-lingual word retrieval system for ancient manuscripts”, Pattern Recognition, Vol. 42 (9), pp. 2089–2105, 2009.

[5] A. Fischer et al., “Lexicon-free handwritten word spotting using character HMMs”, Pattern Recognition Letters, Vol. 33 (7), pp. 934–942, 2012.

[6] P. P. Roy, U. Pal and J. Lladós, “Morphology Based Handwritten Line Segmentation Using Foreground and Background Information”, In Proc. ICFHR, pp. 241-246, 2008.

[7] A. K. Bhunia, A. Das, P. P. Roy and U. Pal “A Comparative Study of Features for Handwritten Bangla Text Recognition”, In Proc. ICDAR,2015.

[8] S. Thomas, C. C., L. Heutte and T. Paquet, “An Information Extraction Model for Unconstrained Handwritten Documents”, InProc. ICPR, pp. 3412–3415, 2010.

[9] V. Frinken, A. Fischer, R. Manmatha and H. Bunke, “A Novel Word Spotting Method Based on Recurrent Neural Networks”, IEEE Trans. Pattern Anal. Mach. Intell., Vol. 34(2),pp. 211-224,2012.

[10] M. Rusinol et al, “Browsing heterogeneous document collections by a segmentation-free word spottingmethod”,In Proc. ICDAR,pp. 63–67,2011.

[11] P. P. Roy, U. Pal and J. Lladós. "Text Line Extraction in Graphical Documents using Background and Foreground Information" International Journal of Document Analysis and Recognition, Vol. 15 (3), pp. 227-241, 2012.

[12] Y. Bai, L. Guo, L. Jin and Q. Huang, “A novel feature extraction method using PHOG for smile recognition”,InProc.ICIP, pp.3305-3308, 2009.

[13] P. P. Roy, J.Y. Ramel and N. Ragot, “Word Retrieval in Historical Document Using Character-Primitives”,InProc. ICDAR,pp.678-682,2011.

[14] S. Young. The HTK Book, Version 3.4, 2006.

[15] J. R. Serrano and F. Perronnin, “Handwritten word-spotting using hidden Markov models and universal vocabularies”, Pattern Recognition, Vol.42(9), pp. 2106-2116,2009.