Bag-of-Features HMMs for Segmentation-Free Bangla Word Spotting

L. Rothacker and G. A. Fink
Department of Computer Science
TU Dortmund University
44221, Dortmund, Germany
leonard.rothacker@udo.edu
gernot.fink@udo.edu

P. Banerjee, U. Bhattacharya and
B. B. Chaudhuri
Computer Vision and Pattern Recognition Unit
Indian Statistical Institute
203, B. T. Road, Kolkata-700 0108, India
purnendubannerjee@yahoo.com
ujjwal@isical.ac.in
bbc@isical.ac.in

ABSTRACT

In this paper we present how Bag-of-Features Hidden Markov Models can be applied to printed Bangla word spotting. These statistical models allow for an easy adaption to different problem domains. This is possible due to the integration of automatically estimated visual appearance features and Hidden Markov Models for spatial sequential modeling. In our evaluation we are able to report high retrieval scores on a new printed Bangla dataset. Furthermore, we outperform state-of-the-art results on the well-known George Washington word spotting benchmark. Both results have been achieved using an almost identical parametric method configuration.

1. INTRODUCTION

Word spotting systems search digital document collections for words of special interest. In this paper we consider a segmentation-free Query-by-Example scenario. The query is, therefore, given by a word that is selected in a document image. Visually similar regions in the document database are then retrieved and returned in a ranked list. This way digital archives containing, for example, historical documents are made accessible. Otherwise they cannot be efficiently explored, because state-of-the-art text recognition systems do not achieve sufficient recognition results. This is especially a problem in handwritten or degraded printed documents where characters and words substantially vary in their visual appearance [12].

In this paper we address the spotting of printed words in old Bangla documents. Bangla is a popular language of the Indian subcontinent used by about 250 million people. Its script, also called Bangla, is an alpha-syllabary script. The character shapes are more complex and abundant compared to e.g., Roman alphabetic scripts. Because of this complexity and the lack of annotated training corpora, even the printed Bangla text Optical Character Recognition (OCR) systems do not achieve acceptable recognition rates in general. This is due to degradations like ink smearing, bleed-through or bad paper color. Primitive printing systems cause problems by creating undulated text lines and arbitrary word spacings. The use of obsolete font shapes and differences in the word spellings create further variabilities.

Although only limited work on Bangla word spotting is available in the literature, a sufficient number of studies on Bangla OCR approaches has been reported. Chaudhuri and Pal [3] suggested a template-matching tree classifier for character recognition. In [9], Mahmud et al. presented a Neural Network based character recognition approach depending on Freeman Chain Codes. Hasant et al. [5] proposed an open Tesseract based OCR for Bangla script. A common property of these methods is that they are very sensitive with respect to font size and type. Large amounts of annotated samples are required for classifier training.

In this paper, we use Bag-of-Features Hidden Markov Models (HMM) for spotting printed Bangla words. The approach has first been presented in [14] where Bag-of-Features HMMs have been applied to offline Arabic handwriting recognition. Later, the method has been adapted to segmentation-free handwritten word spotting [13]. The approach is now further explored for a different script in a printed, old document scenario. One key aspect is to show that state-of-the-art recognition results can be achieved by adapting the method to different word spotting tasks with minimal effort. Additionally, only a single sample, the query selected by a user, is required for estimating our model. No preprocessing techniques like line or word segmentation and text normalization have to be applied.

Bag-of-Features are a standard representation in Computer Vision [10]. They belong to the most effective methods for object categorization and retrieval. Powerful but simple modifications like the Spatial Pyramid [6] made them widely noticed. Recently, they also became more popular in the document analysis field. In [15] a Bag-of-Features representation was proposed for segmentation-free word spotting and [17] exploited the approach for handwritten and machine-printed text separation. Finally, the application domain considered in [16] is quite similar to ours. Bag-of-Features were used for indexing and retrieving segmented word images in four different Indian document collections. The method takes advantage of the large scale indexing capabilities in Bag-of-Features approaches. Hundred thousands of word images can efficiently be retrieved this way. However, the efficiency comes at the cost of decreased precision. This can be compensated to some extent by re-ranking the retrieval results.

Bag-of-Features image representations are built upon local image descriptors cf. [10], like the SIFT descriptor [8]. First the *features* in the Bag-of-*Features* must be defined. As these are specific to the application domain, this can be considered as the estimation part that is usually only performed initially. Afterwards, images can be represented by a histogram over the features, thus by a Bag-of-Features. In the first step the features are usually computed by clustering descriptors. In the second step for each descriptor the most similar feature is determined by quantization. The property that makes Bag-of-Features powerful is that the *features*, that are representative for the problem domain, can be found in a purely data-driven and unsupervised manner. This way the representation can easily be re-estimated and adapted to a different scenario, as demonstrated in [15].

When recognizing text it is very important to take its sequential nature into account. Consequently statistical sequence models are among the most successful approaches in the field. A Bag-of-Features, on the other hand, discards all spatial information by capturing the simple occurrence of a feature in an image. In [15, 16] this is overcome by using a Spatial Pyramid consisting of concatenated Bag-of-Features representations for adjacent spatial cells. Bag-of-Features HMMs allow for a spatial sequential modeling in a dynamic probabilistic way. These HMMs model the generation of a Bag-of-Features representation at each point in time. The discrete characteristic of the Bag-of-Features allows to directly model feature probabilities within each HMM state. A continuous output model, like a Gaussian Mixture Model, can be entirely omitted this way [14]. Furthermore a Bag-of-Features HMM can be estimated by a single sample. This is important for the Query-by-Example word spotting task.

Thus, we present two major contributions. This is the first attempt of spotting unsegmented printed Bangla words with Bag-of-Features HMMs. And most importantly, our method can easily be adapted from spotting handwritten Latin words, while achieving top results in both cases.

2. BAG-OF-FEATURES HMMS FOR WORD SPOTTING

Bag-of-Features HMMs for segmentation-free word spotting have first been presented in [13]. In this section we will review all important aspects of the method and show exemplarily how it is applied for a printed Bangla word spotting task. For this purpose Figure 1 shows an overview of the entire process.

Starting from a collection of document images, four different steps are necessary for spotting query words. (i) The document images must be represented by the features that the Bag-of-Features will be created from. (ii) The query must be modeled by a Bag-of-Features HMM. (iii) Given the query model, probabilistic similarity scores are computed on each image in the document collection. (iv) These scores are used to extract non-overlapping regions of interest that are ranked according to their similarity to the query.

Figure 1 (1) and (2) show the document image feature representation. As widely applied for computing Bag-of-Features of images, SIFT descriptors are extracted in a dense grid. Figure 1 (2) visualizes this in a simplified manner. In practical applications the descriptors are highly overlapping. In order to estimate the features considered in the Bag-of-Features representation, 20% of the descriptors from each document image are randomly sampled. These are then clustered with the Generalized Lloyd algorithm [7]. The features are given as the centroids in the codebook obtained. Finally, all descriptors in the dense grids of all document images are quantized according to their most similar feature. In Figure 1 (2) this is shown by a dense grid of colored points. Each point refers to a descriptor and the point's color to the feature that the descriptor has been assigned to. For the dense grid a spacing of 5×5 pixels has been used. The SIFT descriptor covers 60×60 pixel. Note that the document images have been smoothed for avoiding binarization artifacts but no scale space representation has been considered. The descriptor orientation is always set to zero degrees. The number of features computed through clustering has been set to 4096. These parameters have been chosen based on the resolution and text height of the document images and prior experiments from [13].

For modeling a query, an example of a respective word instance must be selected in the image. Features in this region are available from the previous step. Bag-of-Features representations are now obtained for each column in the dense grid. Figure 1 (3) visualizes the column-wise extraction with the sliding window. The Bag-of-Features HMM finally models the generation of this Bag-of-Features *frame* sequence. Figure 1 (3) shows exemplarily how the feature probabilities are estimated for each HMM state. As in this example many cyan-colored points can be observed at the beginning of the word, consequently this feature has also a high probability in the first HMM state. For the query model estimation there are three parameters. The *feature pruning* parameter refers to selecting features in the query word bounding box. When creating the Bag-of-Features sequence, it is not useful to consider all features. Features close to the upper and lower boundary might encode information that is not specific to the query word. Apart from that there are the HMM parameters *number of states* and *number of Baum-Welch iterations* that have to be specified cf. [14, 13]. These parameters mainly influence how specific the model is with respect to the query word.

In order to perform segmentation-free word spotting, we use a patch-based framework. We, therefore, densely sample patches on all document images. The horizontal and vertical patch overlap is 75%. Each patch has exactly the same size as the bounding box of the word used as query. As for the query model, a sequence of Bag-of-Features frames is extracted. Using this sequence we can decode the query model with the Viterbi algorithm in order to obtain a probabilistic score for each patch. Figure 1 (4) and (5) visualize this process. In Figure 1 (5) the interpolated probabilistic scores are visualized with blue to red colors.

Finally, overlapping patches are eliminated through non-maximum-suppression. The remaining patch-candidates are ordered according to their scores and the top 200 responses per page are returned as word spotting result. Figure 1 (6) shows the patches obtained from the patch-scores visualized in Figure 1 (5). Again, the colors denote similarity to the query word. Patches filled with color show relevant hits in the considered example.

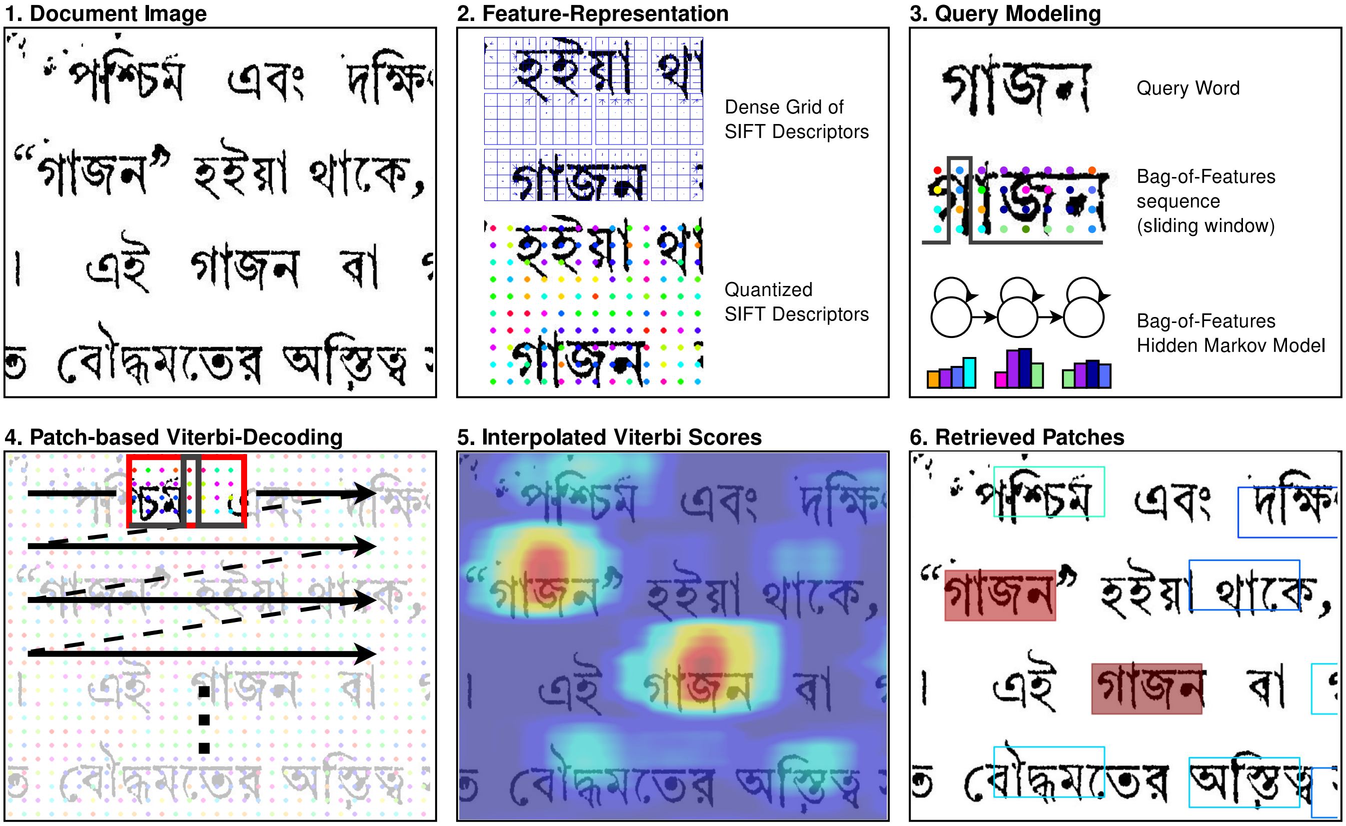
1. Document Image
2. Feature-Representation
Dense Grid of SIFT Descriptors
Quantized SIFT Descriptors
3. Query Modeling
Query Word
Bag-of-Features sequence (sliding window)
Bag-of-Features Hidden Markov Model
4. Patch-based Viterbi-Decoding
5. Interpolated Viterbi Scores
6. Retrieved Patches

Figure 1: Overview of using Bag-of-Features HMMs for segmentation-free printed Bangla word spotting.

3. BANGLA WORD SPOTTING DATASET

There exists a huge heritage of Indian literature. In order to increase its accessibility to the public, the mission Digital Library of India [2] is creating a large repository that should be free-to-read and searchable. Next to other Indian languages, alone 16723 Bangla books consisting of more than 5.5 million pages have been scanned. While it is possible to browse the document images and filter for meta information, a content-based search through OCR is largely not applicable. This is due to the quality of the old documents and the scanning process. We, therefore, propose to use word spotting as a more robust alternative. For our simulation study we downloaded 34 scanned pages from the old Bangla book entitled Aadyer Gambhira [11]. An exemplary text excerpt can be seen in Figure 1 (1). The ground truth has been created manually and consists of transcriptions and bounding box coordinates on word level.

In the Bangla alphabet there exists a huge variety of different words that are built from 11 vowel characters, 39 consonant characters and more than 200 compound characters. The compound characters are created through 10 vowel and 2 consonant modifiers that can be attached to a character on its left, right, on both sides or at the bottom. For further details refer to [4]. On the 34 pages considered for our word spotting benchmark there are a total of 5118 words. In Table 1 we distinguish between words from the lexicon over the dataset (lexical items) and word instances. Due to the huge number of character variants, over 25% of all word instances only appear a single time in the dataset. With respect to the lexicon almost 65% of all items appear only once. We will investigate to what extent this effects our word spotting results in Section 4.

Table 1: Words occurring in the Bangla benchmark

Property	Total	Multiple	Single
Word instances	5118	3803	1315 (25.7%)
Lexical items	2025	710	1315 (64.9%)

Another important property for a word spotting benchmark is its distribution of different word lengths. While, for instance, in Roman scripts the word length can easily be measured by the number of characters, this is not feasible in Bangla. An additional unit named ortho-syllable has to be considered in alpha-syllabary Indian scripts. From the orthographic point of view, an ortho-syllable can be (i) a basic (vowel or consonant) character, (ii) a basic consonant with a vowel modifier or consonant modifier and (iii) a conjunct character with a vowel modifier, in the word. As the modifiers may be attached on all sides, a simple left-to-right character sequencing is impossible. Therefore, the number of ortho-syllables gives a notion of the word length. Figure 2 shows the word length histogram for the 34 printed Bangla pages. It visualizes that a major percentage of the words is rather short in numbers of ortho-syllables. The overall scores in the evaluation (Section 4, cf. Figure 3) are biased respectively.

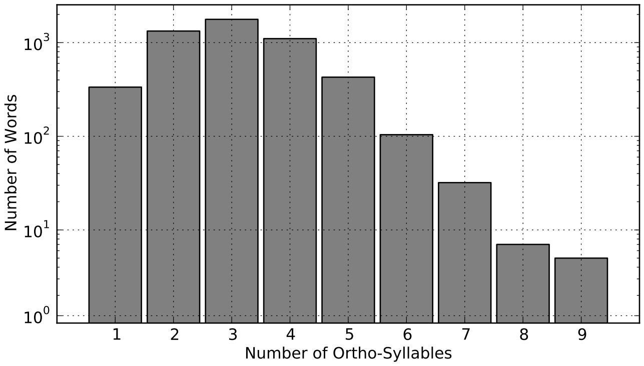

Figure 2: Word length histogram

Table 2: Influence of query feature pruning

Query feature pruning	MAP	MR
No feature pruning	94.8%	97.8%
Vertical feature pruning	**95.4%**	**98.3%**
Horizontal and vertical feature pruning	86.3%	98.4%

4. EVALUATION

We evaluate our method on the printed Bangla dataset (BG) presented in Section 3. In order to show that the method can be easily adapted to a different scenario, we also report results obtained on the George Washington dataset [12] (GW). The almost same parametric configuration that is producing the best results on the BG benchmark has been used. Originally, results for spotting words with Bag-of-Features HMMs on the GW benchmark have been reported in [13]. The dataset, consisting of 20 handwritten pages, has first been used for word spotting in [12]. For comparability we have been following the evaluation protocol used in [15, 1]. It will also be considered for the BG benchmark. For both benchmarks ground truth annotations on word level are available. In the evaluation protocol every word acts as query and for each query a ranked list of patches is returned. We calculate the average precision and recall scores respectively and finally report the word spotting performance in terms of Mean Average Precision (MAP) and Mean Recall (MR). Following [15, 1] a patch is considered as relevant if it overlaps with a relevant annotation in the ground truth by more than 50%. Note that in our segmentation-free framework also the query itself can be retrieved and counted as a relevant item. In addition to this standard protocol we address another aspect that has already been discussed in Section 3. A substantial amount of words is occurring only a single time and it is, therefore, relatively easy to obtain good average precision and recall values in those cases. In the Mean Average Precision and Mean Recall calculation, finally, all scores are weighted equally which might bias the overall result. For that reason we will also report scores that have been obtained without considering those queries.

Table 3: Influence of the number of model states

Model states scaling	MAP	MR
15% of query frame number	95.0%	98.3%
30% of query frame number	95.0%	98.2%
Linear combination (**30%** − **15%**)	**95.4%**	**98.3%**

Table 4: Influence of Baum-Welch training

Baum-Welch iterations	MAP	MR
0	94.4%	98.5%
5	**95.4%**	**98.3%**
10	95.3%	98.1%

The parametric evaluation of the BG benchmark focuses on the three query model parameters *feature pruning*, *number of model states* and *number of Baum-Welch iterations*. The overall parameter configuration is always the same and only one parameter is varied at a time. The parameters considered as best in Tables 2, 3 and 4 are marked with a bold font. Figure 3 shows the Mean Average Precision per word length for this best configuration. In general, it is more difficult to retrieve short words. The shorter the Bag-of-Features frame sequence the less distinctive the representation. Also, shorter words are likely to appear within longer words. Table 2 shows the results for different feature selection schemes. Pruning vertical features refers to only considering features that do not overlap with the upper and lower query word boundary. Horizontal and vertical feature pruning refers to additionally only keeping features that do not overlap with the left and right query word boundary. The clear benefit of pruning vertical features can be explained with their low specificity. The upper and lower parts of Bangla characters are often similar, as can also be observed in Figure 1 (1). The substantial performance drop for additionally pruning horizontal features can be explained with the lack of word context. Query words appearing within other words receive high retrieval scores in this case.

The number of query model states is determined relative to the number of frames in the Bag-of-Features sequence. Table 3 shows the evaluation for 15% and 30% as well as for their linear combination from 30% for low frame numbers to 15% for high frame numbers. Fewer states allow for a more flexible modeling of longer words. This is helpful because the patch sampling density depends on the patch size in our segmentation-free retrieval approach (cf. Section 2). Longer words might, therefore, not be detected as precisely as shorter words.

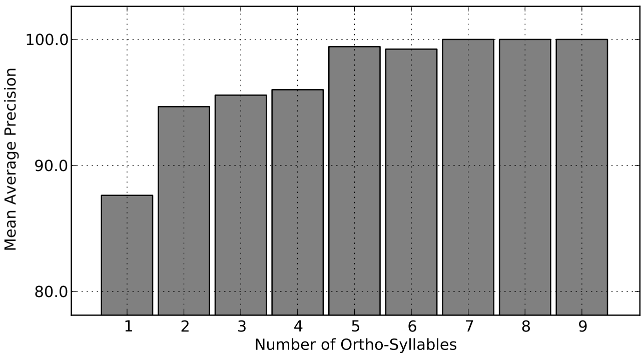

Figure 3: Mean Average Precision per word length

Table 5: Bag-of-Features HMMs for word spotting

Method	Dataset	MAP	MR
Proposed	BG, 5118 queries	95.4%	98.3%
Proposed	BG, 3803 queries	94.1%	97.7%
Proposed	GW, 4860 queries	67.2%	82.3%
Proposed	GW, 4221 queries	62.5%	79.7%
Rothacker et al. [13]	GW, 4860 queries	61.1%	95.5%
Almazán et al. [1]	GW, 4856 queries	54.4%	–

Table 4 shows the effect of re-estimating the model with the Baum-Welch algorithm. Due to this training the model gets more and more specific to the query word. A trade-off between generality and specificity is found after 5 iterations.

Finally, an overview showing Bag-of-Features HMM word spotting performance on the printed Bangla dataset and the handwritten George Washington dataset can be found in Table 5. First, results for the *Proposed* method are given. The only difference in the parameterization for BG and GW benchmarks is in the descriptor size. It has been adapted based on the typical line height in the datasets. Furthermore, we show the effect of not considering query words appearing only a single time. The performance decay for the BG benchmark is not as severe as for the GW benchmark. This is due to the overall good retrieval performance. In the last part of Table 5 a comparison to other word spotting methods on the GW benchmark is given. Results for the Bag-of-Features HMM reported in [13] were obtained with a different parameterization and evaluation protocol. The major difference was a patch - ground truth overlap of 20% for considering a retrieved region as relevant. This allows for higher Mean Recall scores. In [1] a Histogram-of-Gradients descriptor was used to model a query word. Patches were then retrieved in a segmentation-free framework. Since here we have been using the same evaluation protocol, the results are comparable.

5. CONCLUSION

In this paper we presented a word spotting benchmark on printed Bangla documents. Our method achieves high recognition scores without being specifically designed for Bangla scripts or printed documents. Furthermore, we are able to report state-of-the-art results on a completely different handwritten Roman word spotting task using the almost same parametric configuration. Only the descriptor size has been adapted to the typical text line height in the documents. Regarding a comparison to other state-of-the-art methods we clearly outperform the results reported in [1]. Future research will address the problem of spotting and recognizing handwritten Bangla words what can be considered as an open research topic.

6. REFERENCES

[1] J. Almazán, A. Gordo, A. Fornés, and E. Valveny. Efficient exemplar word spotting. In *Proc. of the British Machine Vision Conf.*, pages 67.1–67.11, 2012.

[2] N. Balakrishnan. Universal Digital Library: Future research directions. *Journal of Zhejiang University Science*, 6A(11):1204–1205, 2005.

[3] B. B. Chaudhuri and U. Pal. A complete printed Bangla OCR system. *Pattern Recognition*, 31(5):531–549, 1998.

[4] G. A. Fink, S. Vajda, U. Bhattacharya, P. S., and B. B. Chaudhuri. Online Bangla word recognition using sub-stroke level features and hidden Markov models. In *Int. Conf. on Frontiers in Handwriting Recognition*, pages 393–398, 2010.

[5] M. A. Hasnat, M. R. Chowdhury, and M. Khan. An Open Source Tesseract Based Optical Character Recognizer for Bangla Script. In *Proc. of the Int. Conf. on Document Analysis and Recognition*, pages 671–675, 2009.

[6] S. Lazebnik, C. Schmid, and J. Ponce. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In *IEEE Comp. Soc. Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 2169–2178, 2006.

[7] S. P. Lloyd. Least squares quantization in PCM. *IEEE Trans. on Information Theory*, 28(2):129–137, 1982.

[8] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *Int. Journal of Computer Vision*, 60:91–110, 2004.

[9] J. Mahmud, M. Raihan, and C. Rahman. A complete OCR system for continuous Bengali characters. In *TENCON Conf. on Convergent Technologies for the Asia-Pacific Region*, pages 1372–1376 Vol.4, 2003.

[10] S. O'Hara and B. A. Draper. Introduction to the Bag of Features Paradigm for Image Classification and Retrieval. *Computing Research Repository*, arXiv:1101.3354v1, 2011.

[11] H. Palit. *Aadyer Gambhira.* Krishnacharan Sarkar, Maldaha, 1913.

[12] T. Rath and R. Manmatha. Word spotting for historical documents. *Int. Journal on Document Analysis and Recognition*, 9(2–4):139–152, 2007.

[13] L. Rothacker, M. Rusiñol, and G. A. Fink. Bag-of-Features HMMs for Segmentation-Free Word Spotting in Handwritten Documents. In *Proc. of the Int. Conf. on Document Analysis and Recognition*, 2013.

[14] L. Rothacker, S. Vajda, and G. A. Fink. Bag-of-Features Representations for Offline Handwriting Recognition Applied to Arabic Script. In *Proc. of the Int. Conf. on Frontiers in Handwriting Recognition*, 2012.

[15] M. Rusiñol, D. Aldavert, R. Toledo, and J. Lladós. Browsing heterogeneous document collections by a segmentation-free word spotting method. In *Proc. of the Int. Conf. on Document Analysis and Recognition*, pages 63–67, 2011.

[16] R. Shekhar and C. Jawahar. Word image retrieval using bag of visual words. In *Int. Workshop on Document Analysis Systems*, pages 297–301, 2012.

[17] K. Zagoris, I. Pratikakis, A. Antonacopoulos, B. Gatos, and N. Papamarkos. Handwritten and Machine Printed Text Separation in Document Images Using the Bag of Visual Words Paradigm. In *Proc. of the Int. Conf. on Frontiers in Handwriting Recognition*, pages 103–108, 2012.