Segmentation-free Query-by-String Word Spotting with Bag-of-Features HMMs

Leonard Rothacker
Department of Computer Science
TU Dortmund University
44221, Dortmund, Germany
Email: leonard.rothacker@udo.edu

Gernot A. Fink
Department of Computer Science
TU Dortmund University
44221, Dortmund, Germany
Email: gernot.fink@udo.edu

***Abstract*—Word spotting allows to explore document images without requiring a full transcription. In the query-by-string scenario considered in this paper, it is possible to search arbitrary keywords while only limited prior information about the documents is required. We learn context-dependent character models from a training set that is small with respect to the number of models. This is possible due to the use of Bag-of-Features HMMs that are especially suited for estimating robust models from limited training material. In contrast to most query-by-string methods we consider a fully segmentation-free decoding framework that does not require any pre-segmentation on word or line level. Experiments on the well-known George Washington benchmark demonstrate the high accuracy of our method.**

I. Introduction

Word spotting is the task of retrieving word occurrences from document images. It is of particular interest when it is infeasible to perform ad-hoc automatic text transcription, like in handwritten and historical documents (cf. Lladós et al. [1]). This is due to the high variability of the script's visual appearance caused by human writing or document degradations. In order to automatically transcribe text in these scenarios, a substantial amount of annotated training data is required. However, this is not easily available because especially historical documents are very unique in backgrounds, fonts or writing styles. If it is necessary to manually transcribe a large volume of documents before a recognizer can be applied, it might in the end be easier to directly put the effort into organizing the manual transcription of the entire corpus [2].

Word spotting offers a compromise where certain word instances can be found automatically without performing a full transcription first. In this scenario the user queries the digital document archive with a keyword and the system searches all words in document images that are relevant to the query. The resulting list is ranked according to relevance and, therefore, aids the user by presenting only the potentially interesting information. In contrast to an explicit transcription, the system does not make any final decisions. Conceptually, this is equivalent to using an Internet search-engine.

An important characteristic of a word spotting system is the query format. In query-by-example scenarios, the user has to provide an exemplary occurrence of the query word. The major advantage is that usually no annotated training data is required if the script's visual appearance throughout the document collection is relatively homogeneous. The disadvantage is that it is impossible to search for arbitrary queries. Early methods computed similarity measures between segmented word images for this purpose (cf. [1]). One successful approach was to extract upper and lower word profiles as well as background-ink transitions. The resulting sequences of feature vectors for word images were aligned with Dynamic Time Warping [3]. A more recent approach, that is also related to our method, omitted the segmentation step by searching for query words in a patch-based segmentation-free framework [4]. Queries and patches were modeled by the popular Bag-of-Features (BoF) representation. BoF are orderless collections of local image features that are typical for the problem domain, e.g., document images, (cf. Section II-A). They can be computed without requiring any prior segmentation. The histogram of typical image features can simply be obtained from the query word or the analysis patch. The segmentation-free property is advantageous because word and line segmentations are usually based upon heuristics. If these heuristics fail there is no recovering from the errors made at this early stage in the recognition pipeline.

In query-by-string scenarios the user provides a textual query. This way the advantages of word spotting systems, i.e., robustness of the recognition results, and the advantages of transcription-based text search, i.e., use of arbitrary queries, can be combined (cf. [5]). In the following we will discuss related methods that are either built upon full-transcription recognizers or recognizers that learn associations between local image features and character models. Two methods that are based upon full transcription recognizers have been presented in [6] and [5]. Given a segmented line image, a sequence of geometrical features is extracted. In [6] a score for the text line and the query word is obtained by the log-likelihood ratio of a keyword and a background Hidden Markov Model (HMM). In [5] a bidirectional long term short term memory neural network produces a sequence of letter probabilities. The probability for the query word is then obtained in a final decoding step.

Methods based on local image descriptors are presented in [7], [8], [9] and [10]. In [7] the query is constructed using a glyph book of letter templates. In the retrieval stage this model is matched with zones of interest in the document image. This basic principle of dynamically generating query model representations from templates that have been generated synthetically or obtained from training data is quite popular (cf. [7]). Their query-by-string method, however, is the first that is not based upon any word or line segmentation. A strong limitation of these methods is their sole applicability in very

homogeneous document collections, like printed or uniformly written manuscripts.

That query word representations generated from synthetic data can also be used to retrieve images of handwritten words in a multi writer scenario was demonstrated in [8]. With their semi-continuous HMM a codebook is learned from handwritten word images. At retrieval time they estimate state dependent transition probabilities and mixture weights according to synthetically generated word images using the previously estimated codebook. Their query-by-string approach does not require any manually annotated training data.

The methods presented in [9] and [10] are conceptually very similar. The basic idea is to learn a common subspace for mapping visual and textual BoF representations. Given a textual query, word images' visual representations can be ranked according to their similarity to the textual representation within the subspace. While both methods are built upon local SIFT images features (cf. [11]), they mainly differ in their approach to learning the common subspace. In [9] associations between spatial pyramid representations and n-gram statistics are estimated by *latent semantic indexing* (LSI). In contrast, in [10] relations between textual and visual features are modeled by embeddings and are represented in a joint vector space with *common subspace regression* (CSR).

The above-mentioned query-by-string methods mostly rely on an initial segmentation ensuring that word image representations contain exactly the relevant information. In segmentation-free approaches such representations are hardly obtainable and models have to cope with additional noise. Furthermore, usually a lot of annotated training data is required. Otherwise, methods only work in scenarios where the variability of the script's visual appearance is limited.

In this paper we present a query-by-string word spotting system that is very robust with respect to the amount of training data. Furthermore, it is completely segmentation-free. No prior segmentation on line or word level is required. This is achieved by Bag-of-Features HMMs (BoF-HMMs) that are applied in a patch-based segmentation-free framework. BoF-HMMs model sequences of BoF representations. This can be seen as a dynamic, probabilistic extension of the popular spatial pyramid representation that models spatial relations in a fixed grid of cells. BoF-HMMs have first been presented in [12] and have been applied for segmentation-free query-by-example word spotting in [13]. The novelty, presented here, is their application in a query-by-string scenario where we estimate a large number of context dependent character models with comparatively limited amounts of training data. In order to operate in a segmentation-free scenario, we estimate spatial character model sizes that are required for dynamically generating an analysis patch size in the word retrieval stage.

II. Bag-of-Features HMMs for Segmentation-free Query-by-String Word Spotting

The method for segmentation-free query-by-string word spotting is based upon our previous work for segmentation-free query-by-example word spotting presented in [13]. The main idea is to model the query word with a BoF-HMM and search for similar document image regions in a patch-based decoding framework. Both methods share the feature representation and model decoding step. The difference lies in the query modeling. In the query-by-example scenario the BoF-HMM can directly be created from the word image given as an exemplary occurrence of the query word. The patch size required for model decoding is given by the size of the query word image, accordingly. In order to spot arbitrary, textually defined words, the query word model is dynamically constructed from character models. Character HMMs are concatenated to a query word HMM and the query patch size is estimated from the character model width and height estimates.

The following sections contain a detailed discussion of the entire process that is also visualized in Figure 1. The figure's top row illustrates the document image representation (Section II-A). The middle row refers to the character model estimation (Section II-B). Finally, the bottom row shows the query generation and decoding required for spotting query word instances in document images (Section II-C).

A. Document image representation

The entire word spotting method is built upon BoF representations. BoF-HMMs model BoF sequences that are generated locally in document images. The document image representation is the same as in [13]. For that reason we will review the most important details.

Initially, we extract highly overlapping SIFT descriptors in a dense grid of 5×5 pixels. By using a dense grid, the entire document image is represented and processed uniformly. No decisions about more or less important image regions are required. The SIFT descriptors (cf. [11]) capture the main directions of the underlying pen-stroke and have been used in many document analysis methods (cf. e.g., [4], [12], [9]). In the following the descriptors have a fixed size of 40×40 pixels. The size, as well as the dense grid resolution, have been determined in prior experiments (also cf. [4], [13]) and are related to the typical line height in the document images. Also the descriptors' orientations are fixed as we want to distinguish between, for example, horizontal and vertical pen-strokes. In order to compute a BoF representation, a visual vocabulary is required. This is obtained by clustering randomly sampled 20% of the SIFT descriptors from all document images with Lloyd's algorithm. The centroids in the resulting codebook are the visual words in the visual vocabulary. According to our previous experiments we use 4096 visual words. This can be seen as a compromise between sufficiently high precision and computational efficiency (cf. [14]). Finally, we quantize all descriptors with respect to their most similar visual word.

B. Character model estimation

The major advantage of query-by-string methods is the possibility of spotting arbitrary query words. For this purpose query word models are generated dynamically by combining character models. A character model consists of two parts. Its appearance is modeled by a BoF-HMM. Its size is modeled by its typical width and height. Character models must be estimated in an initial training step.

Given a database of annotated word images, a sequence of feature vectors is extracted for each word image. Here, we extract a sequence of BoF with a sliding window. At each window position we obtain a histogram of visual words

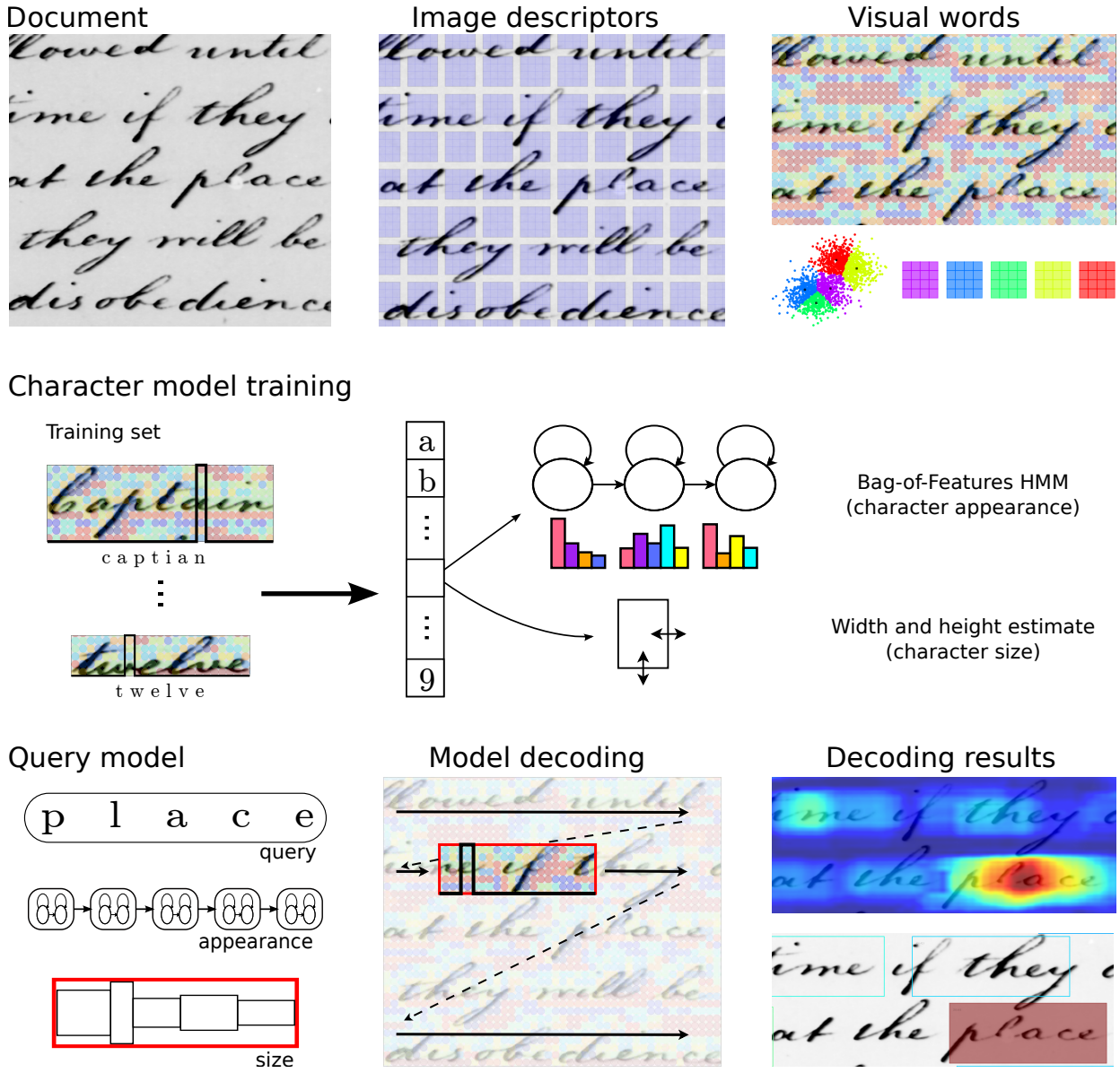

Figure 1. Segmentation-free query-by-string word spotting method. The document's feature representation is shown in the top row. Some exemplary non-overlapping descriptors are visualized by blue patches. Different visual words in the dense grid a indicated by colored points. A corresponding exemplary visual vocabulary is shown below. The middle row shows the character estimation process. Character BoF-HMMs are estimated from sequences of BoF representations. Spatial width and height estimates are indicated by a bounding box. Finally, the retrieval stage is shown in the bottom row. In this example, the query *place* is dynamically generated from character HMMs. Its patch size is visualized by the red rectangle. In the patch-based decoding framework similarity scores are computed and indicated by blue to red colors. Detection results are generated for the best scores obtained after non-maximum suppression.

in a sparse vector representation. Character models can now be estimated with the Baum-Welch algorithm following a standard approach for HMM-based systems (cf. [6], [15]). The distinguishing property of BoF-HMMs is the possibility to model the observation of more than a discrete feature at a point in time. Visual word probabilities are associated with each HMM state. A BoF's output probability in a state at a given time is obtained by a probabilistic similarity measure between the state's visual word probability distribution and the observed visual word distribution. Further details regarding their application to word spotting can be found in [13].

For the BoF-HMM model estimation procedure we focus on standard parameters that are already extensively studied in the literature (cf. e.g., [15]). We experiment with the models' context dependency which is very important due to the nature of the SIFT descriptors. As Figure 1 shows, the descriptors are highly context dependent as they usually capture small groups of characters. For this reason a character's context is defined by its directly adjacent characters, also considering white space at word boundaries. Due to the limited amount of training data, not all context dependent character models that might be required at query time, have been observed during training. In order to guarantee the possibility to use arbitrary query words, we estimate context independent models additionally.

The number of HMM states as well as the number of Baum-Welch training iterations are very important for the

Table I. MODEL PARAMETER EVALUATION

Context	Topology	# States	# Iter.	Height Perc.	mAP	mR
indep.	Bakis	6	7	20	68.2 %	91.4 %
dep.	Bakis	6	7	20	**76.5 %**	**92.6 %**
dep.	Linear	6	7	20	69.3 %	90.9 %
dep.	Linear	3	7	20	71.3 %	92.2 %
dep.	Bakis	3	7	20	71.7 %	92.3 %
dep.	Bakis	9	7	20	73.5 %	91.7 %
dep.	Bakis	6	3	20	73.6 %	93.1 %
dep.	Bakis	6	10	20	75.6 %	92.8 %
dep.	Bakis	6	7	5	75.0 %	91.0 %
dep.	Bakis	6	7	50	74.0 %	90.0 %

specificity of the models. The more states the more detailed the visual appearance model becomes. The Baum-Welch algorithms fits the models to the training data, thus influencing the capability of generalizing to unseen data. Finally, the model topology is important for the flexibility of the models. While a linear topology only allows transitions to the current and the next state, state skips are allowed with Bakis models.

With respect to the models' size, width and height estimates are required. The width estimate is obtained as the average width of character models aligned with the training data using the Viterbi algorithm. As the height information is only available on word level, we have to rely on a heuristic. For each character model we consider all the heights of word images from the training data that the character occurs in. An estimate can then be found at a lower percentile (see parameter evaluation in Table I) of the word height distribution. While the height of smaller characters is typically over-estimated, better results can be obtained for larger characters. This is due to the lower frequency of words containing only smaller characters.

C. Segmentation-free query word retrieval

In the word spotting scenario considered, a query word is given as a textual string representation. For the patch-based, segmentation-free retrieval stage the query word's appearance model and the patch size are required. If all characters from the query are available in the inventory of character HMMs, a word model is constituted by concatenating these character models. More specific context dependent models are favored over less specific context independent models. The patch width is determined by adding the widths of the individual character models, while the height is given as the maximum over the individual character models' heights. This way the largest character still fits in the analysis patch.

In order to search for the query word on a document image, the analysis patch is slid over the dense grid of visual words. For each patch position a sequence of BoF representations is extracted. The probability that this sequence was generated with the query word model is computed with the Viterbi algorithm and used as similarity measure. Patches that are most similar to the query model are obtained after applying a non-maximum-suppression filter on the score map. They are sorted by similarity and returned as the final word spotting result.

III. EVALUATION

We evaluate our method on the well-known George Washington benchmark (cf. [3], [4], [5], [9], [13], [10]). The benchmark consist of 20 pages and 4860 words that have been written by George Washington and his associates. It can be considered as a single writer scenario, due to the script's relatively homogeneous appearance. We mainly follow the evaluation process described in [9] and [10]. We divide the 20 pages in 4 folds, each consisting of 5 test pages. The remaining 15 pages constitute the dataset that we use for estimating character models. Every word in the test is used as a query. We do not filter words with respect to stop lists, stemming etc. As customary in word spotting evaluations, we report mean average precision (mAP) and mean recall (mR) over all queries in all folds. Mean average precision is used for evaluating the ranking of relevant items in the retrieval list. Mean recall measures how many relevant items from the ground truth are present in the retrieval list. For our segmentation-free word spotting method a notion of relevance with respect to detected patches is required. A region is considered as relevant if it overlaps with a respectively annotated region in the ground truth by more than a given percentage. This is the basic difference with respect to the evaluation of segmentation-based methods in [9], [10] (see Table II). In this section we will discuss an experimental evaluation followed by a presentation of results for related methods from the literature. Please note, however, that to the best of our knowledge we are not aware of any publications reporting performance measures for segmentation-free query-by-string word spotting on the George Washington dataset. For that reason a direct comparison is impossible. Recognition rates only allow for a rough placement by also taking the methods' prerequisites into account.

In the method's evaluation we focus on the model parameters discussed in Section II-B. Table I shows the recognition accuracies where the patch overlap relevance threshold is 50 %. The best result obtained is marked with a bold font. The first experiment investigates the difference between context dependent and context independent models. Because local image descriptors usually cover small groups of characters, this context dependence is also reflected in the choice of character models. Please note the substantial improvement despite the limited amount of training material. In the context independent case 36 character models are estimated while we estimate in average 1706 context dependent character models over all training sets in the cross validation. This shows the robustness of BoF-HMMs and their suitability for the given task.

Also the model topology has a substantial influence. Bakis models are much more flexible due to their ability to skip states. They are, therefore, able to model more appearance variants than models using a linear topology.

The next experiments investigate parameters that require a trade-off between specificity of the models and their ability to generalize to unseen data. The results show that 6 states per model yield optimal recognition rates (# States). The trade-off for training with the Baum-Welch algorithm is found after 7 iterations (# Iter.). For the sake of completeness we also added an experiment using a linear topology. Due to the limited flexibility, less states are rather likely to have a positive effect in contrast to a Bakis topology. This can also be observed in the results.

Finally, we investigate the height percentile (Height Perc.) parameter used for estimating a characters typical height. Results show that using the percentile over the distribution

Table II. QUERY-BY-STRING WORD SPOTTING ON GW

Method	Evaluation	Segmentation	mAP	mR
Proposed	15-5	free (50% overlap)	76.5 %	92.6 %
Proposed	15-5	free (25% overlap)	80.1 %	98.8 %
Proposed	5-15	free (50% overlap)	54.6 %	88.8 %
Proposed	5-15	free (25% overlap)	58.1 %	97.5 %
Aldavert et al. [9]	15-5	word-level	56.5 %	
Almazan et al. [10]	15-5	word-level	91.1 %	
Aldavert et al. [9]	15-5 (IVW)	word-level	76.2 %	
Almazan et al. [10]	15-5 (IVW)	word-level	93.9 %	
Frinken et al. [5]	10-5-5 (IVW)	line-level	71.0 %	
Fischer et al. [6]	10-5-5 (IVW)	line-level	60.0 %	(cf. [5])

of word heights is very robust. Recognition rates have only little variance with respect to the changes of the parameter.

Table II shows an overview of query-by-string word spotting results on the George Washington benchmark. In the first part of the table we show results for different patch overlap thresholds in our segmentation-free evaluation. Afterwards, we report results with very limited amounts of training material (5 pages training and 15 pages testing) in order to emphasize the robustness of our model. All recognition rates have been obtained with the best parameter configuration from Table I. Our experiments show an interesting property of the segmentation-free evaluation protocol. A lower overlap threshold leads to substantially improved mean recall values and a considerable improvement in mean average precision. Thus, correct detections are not regarded as relevant when using a threshold of 50 %. On the one hand this shows that good similarity scores are possible although the detection patch does not perfectly fit the word in the document. On the other hand this shows that there is still room for improvement with more accurate patch size estimations. With respect to the experiments using reduced training material we are still able to achieve competitive results. This can been seen when looking at the second part of Table II. It contains recognition rates for related methods from the literature. All of them assume a prior segmentation on word or line level. Thus, their retrieval lists always contain all relevant items, recall is always 100 % and we omit the respective column in Table II. Furthermore, some experiments only consider in-vocabulary queries, i.e., query words that appear in the training set. They are labeled with *IVW*. The results are, therefore, not directly comparable.

The first four experiments show evaluation variants of the methods presented in [9] and [10]. While results from [10] are very robust with respect to in- and out-of-vocabulary words, performance from [9] is severely affected. In terms of recognition accuracy we end up with similar rates when reducing the training set to 5 pages. Almazán et al. [10] set the state-of-the-art for segmentation-based query-by-string word spotting. The last two experiments [6], [5] are based on line-level segmentation. They use a 4 fold cross validation consisting of 10 training, 5 validation and 5 test pages. In their experiments they only consider in-vocabulary words. It is worth noting that feature representations from [9], [10] are suitable for a segmentation-free extension, in contrast to features from [6], [5] that are strongly dependent on correct baseline estimates. Results from [5] have been obtained by averaging recognition rates from 50 randomly initialized neural networks. Their best score was 84 % mean average precision. Results from [6] (reported as baseline in [5]) are interesting because they are also using HMMs and their model is, therefore, conceptually related to our BoF-HMM.

IV. CONCLUSION

In this paper we presented a method for segmentation-free query-by-string word spotting. We achieve very accurate results by estimating context dependent character models from comparably very limited amounts of training data. For segmentation-free retrieval, we set the decoding patch size according to character model size estimates. Although this patch size is fixed for a query and is not adapted to word instances in a document image, we achieve very competitive results. In contrast, segmentation-based methods rely on a perfect word or line segmentation and features only represent information that is relevant to a word instance. In terms of absolute recognition rates our method compares extremely favorably, even with the segmentation-based approaches.

REFERENCES

[1] J. Lladós, M. Rusiñol, A. Fornés, D. F. Mota, and A. Dutta, "On the influence of word representations for handwritten word spotting in historical documents," *IJPRAI*, vol. 26, no. 5, 2012.

[2] T. Causer and M. Terras, "Many hands make light work, many hands together make merry work: Transcribe Bentham and crowdsourcing manuscript collections," in *M. Ridge (ed.), Crowdsourcing Our Cultural Heritage (Ashgate)*, 2014.

[3] T. Rath and R. Manmatha, "Word spotting for historical documents," *IJDAR*, vol. 9, no. 2–4, 2007.

[4] 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*, 2011.

[5] V. Frinken, A. Fischer, R. Manmatha, and H. Bunke, "A novel word spotting method based on recurrent neural networks," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 34, no. 2, 2012.

[6] A. Fischer, A. Keller, V. Frinken, and H. Bunke, "Hmm-based word spotting in handwritten documents using subword models," in *Proc. of the Int. Conf. on Pattern Recognition*, 2010, pp. 3416–3419.

[7] Y. Leydier, A. Ouji, F. LeBourgeois, and H. Emptoz, "Towards an omnilingual word retrieval system for ancient manuscripts," *Pattern Recognition*, vol. 42, no. 9, 2009.

[8] J. A. Rodriguez-Serrano and F. Perronnin, "Synthesizing queries for handwritten word image retrieval," *Pattern Recognition*, vol. 45, no. 9, pp. 3270 – 3276, 2012.

[9] D. Aldavert, M. Rusiñol, R. Toledo, and J. Lladós, "Integrating visual and textual cues for query-by-string word spotting," in *Proc. of the Int. Conf. on Document Analysis and Recognition*, 2013, pp. 511–515.

[10] J. Almazán, A. Gordo, A. Fornés, and E. Valveny, "Word spotting and recognition with embedded attributes," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 36, no. 12, pp. 2552–2566, 2014.

[11] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. Journal of Computer Vision*, vol. 60, 2004.

[12] L. Rothacker, S. Vajda, and G. 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.

[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] M. Rusiñol, D. Aldavert, R. Toledo, and J. Lladós, "Efficient segmentation-free keyword spotting in historical document collections," *Pattern Recognition*, vol. 48, no. 2, pp. 545 – 555, 2015.

[15] T. Plötz and G. A. Fink, "Markov models for offline handwriting recognition: a survey," *IJDAR*, vol. 12, no. 4, pp. 269–298, 2009.