Query by Strings and Return Ranking Word Regions with Only One Look

Peng Zhao[0000−0001−8176−9230], Wenyuan Xue[0000−0001−7398−5785], Qingyong Li[0000−0002−3860−4809], and Siqi Cai[0000−0002−8012−6820]

Beijing Key Lab of Transportation Data Analysis and Mining,
Beijing Jiaotong University, China
{18120456,wyxue17,liqy,18120339}@bjtu.edu.cn

Abstract. Word spotting helps people like archaeologists, historian and internet censors to retrieve regions of interest from document images according to the queries defined by them. However, words in handwritten historical document images are generally densely distributed and have many overlapping strokes, which make it challenging to apply word spotting in such scenarios. Recently, deep learning based methods have achieved significant performance improvement, which usually adopt two-stage object detectors to produce word segmentation results and then embed cropped word regions into a word embedding space. Different from these multi-stage methods, this paper presents an effective end-to-end trainable method for segmentation-free query-by-string word spotting. To the best of our knowledge, this is the first work that uses a single network to simultaneously predict word bounding box and word embedding in only one stage by adopting feature sharing and multi-task learning strategy. Experiments on several benchmarks demonstrate that the proposed method surpasses the previous state-of-the-art segmentation-free methods. [1]

Keywords: Word spotting; Query-by-String; Segmentation-free; Multi-task learning;

1 Introduction

Word spotting [1] is an image retrieval task, which provides a fast way to find regions of interest from document images for people like archaeologists, historian and internet censors. Intuitively, given a set of document images and a query (usually a word image or a word string), the purpose of this task is to find word areas related to the query in document images, and then to return all the retrieved word areas ranked by a certain criterion. For such a task, one possible retrieval method that can be easily come up with is the full-page text detection and recognition. However, word spotting is more efficient, which directly locates keyword regions from document images without extra text recognition and post-processing.

[1] The code is available at https://github.com/zhaopeng0103/WordRetrievalNet.

When using machines to automatically process handwritten historical documents, we face more challenges than modern printed documents due to various writing style, changeable visual appearance, and uneven background. Moreover, there have special characteristics in handwritten historical documents, such as dense words distribution and overlapping strokes, which make word segmentation difficult.

There are two ways for the word spotting task classification. Firstly, according to whether the query is a cropped word area or a word string, the word spotting task can be classified as Query-by-Example (QbE) [2,3,4,5,6] and Query-by-String (QbS) [2,3,7,4,8,5,6,9]. Generally, QbS is closer to the requirements in real scenes because you do not have to find a real word area and crop it from document images every time. Secondly, word spotting methods can be divided into segmentation-based [2,3,7,4] and segmentation-free [8,5,6,9] methods by whether it needs to segment word areas in advance of the matching process. The method proposed in Sudholt et al. [2] is the first work to use a deep convolutional neural network for segmentation-based word spotting. Wilkinson et al. [5] proposed a segmentation-free word spotting method, which produces word segmentation results based on Faster R-CNN [10] and then embeds cropped word regions into a word embedding space in which word retrieval is performed. For the reason that segmented word areas are not always available during training, recent works focus more on segmentation-free QbS word spotting.

In this paper we propose a simple and effective end-to-end trainable method for segmentation-free QbS word spotting, which is scale-insensitive and does not need redundant post-processing. To start with, the method extracts and fuses multi-scale features through a deep convolutional neural network [11] embedded with a feature pyramid network (FPN) [12]. After that, based on feature sharing mechanism, the fused features are passed on to three subtasks for multi-task learning. In detail, the first task performs pixel classification by predicting the probability of each pixel belonging to a positive word area. The second task regresses word bounding box by predicting the offsets of a word pixel to its word bounding box boundaries. The third task learns the mapping from the word area to the word embedding. The queries defined by users are retrieved based on the outputs of the three tasks. To the best of our knowledge, this is the first work that utilizes a single network to simultaneously predict word bounding box and word embedding in segmentation-free word spotting. The proposed method achieves state-of-the-art performance on several benchmarks. And the experimental results prove that it is effective to perform word segmentation by directly regressing word bounding box in handwritten historical document images with special characteristics such as dense words distribution and overlapping strokes.

The main contributions of this paper are summarized as follows:

- We propose a novel end-to-end trainable deep model for segmentation-free QbS word spotting in handwritten historical document images, which simultaneously predicts word bounding box and word embedding by adopting feature sharing and multi-task learning strategy in only one stage;

- The proposed method achieves state-of-the-art results in terms of word retrieval performance on public datasets, which demonstrate the effectiveness of segmenting words by directly regressing word bounding box in handwritten historical document images with dense words distribution and overlapping strokes.

The rest of this paper is organized as follows. Section 2 describes some recent approaches in word spotting. Section 3 presents the proposed end-to-end trainable methodology for segmentation-free QbS word spotting. Section 4 demonstrates the effectiveness of the proposed method on several public benchmarks using standard evaluation measures. And conclusions are drawn in Section 5.

2 Related Work

2.1 Traditional Word Spotting Methods

In document analysis and recognition literature, most traditional methods for handwritten word spotting are based on Hidden Markov Model (HMM) [13,14], Dynamic Time Warping (DTW) [15,16], RNN [17], Bidirectional long short-term memory (BLSTM) [18]. These methods mainly consist of three steps. The first step is the preprocessing of document images, including image binarization, segmentation and normalization. Afterwards features such as SIFT [19] and HoG [20] extracted from segmented word or line images are embedded into a common representation space. Lastly, word image retrieval lists are acquired by distance measurement criteria such as cosine distance, Euclidean distance and edit distance. Rath et al. [15] presented an algorithm for matching handwritten word images in historical document images, which extracts feature representations from segmented word images and uses DTW for comparison. Rath et al. [16] extended the above work and used DTW to compare variable-length feature sequences for word matching. Frinken et al. [17] proposed to locate words based on the combination of BLSTM and CTC token passing algorithm.

However, these traditional methods generally use hand-crafted features, which typically have poor robustness. The method proposed in this paper utilizes deep learning and convolutional neural network to extract and concatenate low-level texture features with high-level semantic features of images, which can improve accuracy of locating word targets with variable sizes and help to achieve excellent performance.

2.2 Deep Learning Based Word Spotting Methods

In recent years, deep learning based methods have achieved significant performance improvement in handwritten word spotting, which are crucial for promoting the research of word spotting. They are typically classified into segmentation-based and segmentation-free methods.

Segmentation-based word spotting approaches [2,3,7,4] have witnessed major advancements with further research on word embedding and extensive application of deep learning. The method proposed in Sudholt et al. [2] is the first work to use a deep CNN architecture for word spotting, which can handle word images with arbitrary size and predict Pyramidal Histogram of Characters (PHOC) [21] representation. Wilkinson et al. [3] employed a triplet CNN to extract word image representation and subsequently embedded it into a novel word embedding, called Discrete Cosine Transform of Words (DCToW). Gomez et al. [7] learned a string embedding space in which distances between projected points are correlated with the Levenshtein edit distance between the original strings based on a siamese network. Finally, Serdouk et al. [4] learned similarities vs differences between word images, then used Euclidean distance for word matching. However, these methods require lots of segmented word areas. Because segmented word areas are not always available during training, this limits the application of handwritten word spotting. Therefore, this paper proposes a segmentation-free word spotting method, which can be applied in any unconstrained scenarios.

Most of the previous segmentation-free word spotting methods [8,5,6,9] are based on sliding windows or connected components or combination of both, depending on how to generate the word image retrieval regions. In the method proposed by Rothacker et al. [8], regions are generated based on sliding windows and queries are modeled by BoF-HMM, where the size of the region for a given query string has to be estimated. Wilkinson et al. [5] predicted word candidate regions based on Faster R-CNN [10], and then embedded clipped candidate regions into a word embedding space in which word retrieval is performed according to the cosine distance from the query. Three different word detectors are adopted to generate word hypotheses in the method proposed by Rothacker et al. [6]. Then the authors used convolution neural network to predict word embedding and performed word spotting through nearest neighbor search. Vats et al. [9] presented a training-free and segmentation-free word spotting method based on document query word expansion and relaxed feature matching algorithm. These methods generate a large number of candidate regions during word segmentation process, resulting in slow processing speed and too many false positives. Our method directly predicts word bounding boxes based on pixel-level segmentation without redundant post-processing processes.

2.3 Scene Text Detection and Recognition Methods

In the last few years, scene text detection methods [22,23] have attracted extensive attention. EAST [22] adopts fully convolutional network (FCN) [24] to directly produce text regions without unnecessary intermediate steps. PSENet [23] proposes to merge text instances through progressive scale expansion algorithm, which can precisely detect texts with arbitrary shapes. Scene text recognition methods [25] predict character sequences from extracted features. CRNN [25] is the first approach to treat text recognition as a sequence-to-sequence task by combining CNN and RNN in an end-to-end network.

Uniformly, one possible method that can be easily come up with is the full-page text retrieval method, which combines the above text detection and recognition methods into a pipeline and then performs word searching. This method can also achieve word spotting task in historical document images. However, it needs to compare the query with recognition results one-by-one according to whether the content is exactly the same. Different from the above framework, word spotting only needs to label coordinates of query words without recognizing word contents, and then outputs word area retrieval lists ranked by similarity, which is more efficient and more like a tool specifically designed for keyword search tasks. Inspired by scene text detection methods [22,23], the method proposed in this paper combines deep convolutional neural network [11] with feature pyramid network (FPN) [12] to extract image features, and then directly regresses word bounding box and predicts word embedding without complicated post-processing.

3 Method

The proposed method is illustrated in Fig. 1. The input image is first fed into the backbone network to extract multi-scale features and fuse them. Then the fusion features are passed on to three subtasks that predict pixel categories, word bounding boxes and word embeddings, respectively. We present the details of each part in the following subsections.

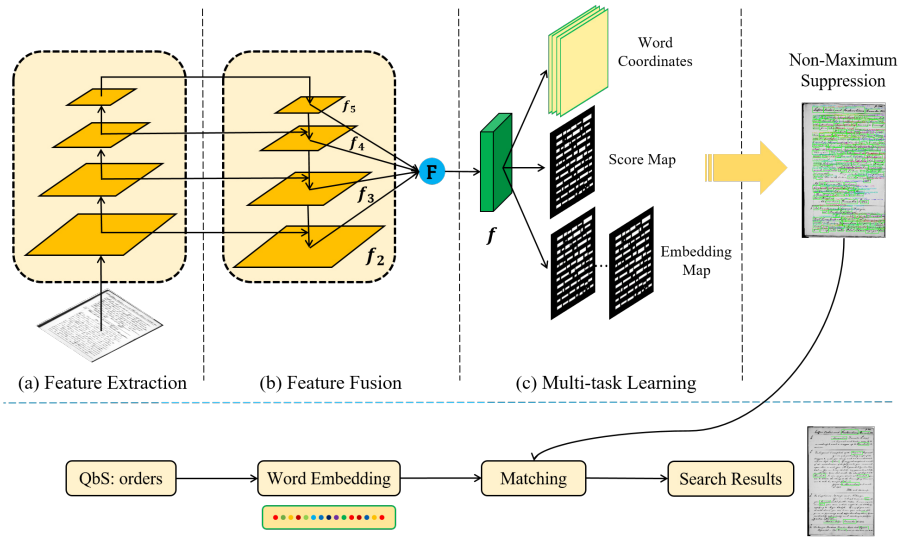

Fig. 1. The pipeline of the proposed method.

3.1 Feature Extraction and Fusion

Words in handwritten historical document images are usually densely distributed and have many overlapping strokes, so it is important to extract appropriate and powerful features. In the proposed method, the ResNet50 [11] pre-trained on ImageNet [26] is adopted as the backbone for feature extraction. Inspired by FPN [12], merging feature maps of different layers may help improve the performance of detecting word areas with various sizes. Therefore, four feature maps are extracted from the ResNet50: the last layer of block1, block2, block3 and block4, whose sizes are $\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{16}$ and $\frac{1}{32}$ of the input image, respectively. Afterwards we concatenate low-level texture features with high-level semantic features to get four feature maps (f_2, f_3, f_4, f_5), whose dimensions are $\left(N, 256, \frac{H}{4}, \frac{W}{4}\right)$, $\left(N, 256, \frac{H}{8}, \frac{W}{8}\right)$, $\left(N, 256, \frac{H}{16}, \frac{W}{16}\right)$ and $\left(N, 256, \frac{H}{32}, \frac{W}{32}\right)$, respectively. N is the batch size. H and W are the height and width of the input image. In order to encode information with various receptive fields, based on the feature fusion part in [23], these four feature maps are further fused to obtain fusion feature map f with dimension $\left(N, 1024, \frac{H}{4}, \frac{W}{4}\right)$. The above fusion process is defined by function $\mathbb{F}(\cdot)$ as follows:

$$f = \mathbb{F}(f_2, f_3, f_4, f_5) = f_2 \| Up_{\times 2}(f_3) \| Up_{\times 4}(f_4) \| Up_{\times 8}(f_5) \ , \tag{1}$$

where "$\|$" represents the fusion operation and $Up_{\times 2}(\cdot)$, $Up_{\times 4}(\cdot)$, $Up_{\times 8}(\cdot)$ represent 2, 4, 8 times upsampling, respectively.

3.2 Multi-task Learning

After feature extraction, the proposed method conducts three subtasks simultaneously for joint supervised learning. The first task classifies word pixels by computing the probability of each pixel belonging to a positive word area. The second task regresses word bounding boxes by predicting the offsets of a word pixel to its word bounding box boundaries. The third task predicts the embeddings of word areas.

Word Pixel Classification For the first task, we feed f into a series of stacked convolutional layers and a Sigmoid layer to produce a single-channel word pixel classification score map with dimension $\left(N, 1, \frac{H}{4}, \frac{W}{4}\right)$, which predicts the probability of each pixel belonging to a positive word area on the resized input image. When building the classification ground truth, we shrink the initial word regions by 0.2 times along the short side of the word boundaries. During training, only the shrinking word regions are treated as positive areas. The areas between the shrinking regions and the bounding boxes are neglected and do not contribute to the classification loss.

There is a strong imbalance between the number of pixels in the foreground and background, because word instances generally occupy only a small region in word areas. In order to prevent predictions of the network biasing to background pixels, we adopt dice coefficient loss [27,23]. The dice coefficient $D(\hat{y}_{cls}, y_{cls})$

between word classification predictions $\hat{y}_{cls}$ and ground truth y_{cls} is formulated as:

$$D\left(\hat{y}_{cls}, y_{cls}\right) = \frac{2\sum_{i,j} \hat{y}_{cls}^{i,j} \times y_{cls}^{i,j}}{\sum_{i,j}\left(\hat{y}_{cls}^{i,j}\right)^2 + \sum_{i,j}\left(y_{cls}^{i,j}\right)^2} \ , \tag{2}$$

where $\hat{y}_{cls}^{i,j}$ and $y_{cls}^{i,j}$ refer to the values of pixel (i,j) in $\hat{y}_{cls}$ and y_{cls}. Thus the word pixel classification loss is defined as:

$$\mathcal{L}_{cls} = 1 - D\left(\hat{y}_{cls}, y_{cls}\right) \ . \tag{3}$$

Word Bounding Box Regression The second task is to obtain the word coordinate map with dimension $\left(N, 4, \frac{H}{4}, \frac{W}{4}\right)$ by feeding f into stacked convolutional layers and a Sigmoid layer. The four channels predict the offsets of a word pixel to the top, bottom, left and right sides of the corresponding word bounding box.

GIoU loss [28] can accurately represent the coincidence degree of two bounding boxes. However, when the target box completely covers the predicted box, it can not distinguish their relative positional relationship. To solve the above shortcomings, DIoU loss [29] tries to predict more accurate word bounding box by adding center point normalized distance. Considering the situation of dense words distribution and overlapping strokes in handwritten historical document images, we adopt the DIoU loss as word regression loss, which can be written as:

$$\mathcal{L}_{bbox} = \frac{1}{|C|}\sum_{i \in C} DIoU\left(\hat{y}_{bbox}, y_{bbox}\right) \ , \tag{4}$$

where C denotes the set of positive elements in the word pixel classification score map, $DIoU\left(\hat{y}_{bbox}, y_{bbox}\right)$ refers to the DIoU loss between the predicted bounding box $\hat{y}_{bbox}$ and the ground truth y_{bbox}.

Word Embedding Prediction The third task aims to learn the mapping from the word area to the word embedding. We train and evaluate our model using Discrete Cosine Transform of Words (DCToW) introduced in [3], which is a distributed representation of a word and has achieved state-of-the-art results in segmentation-based and segmentation-free word spotting methods [2,5]. The calculation process from a word string to the corresponding word embedding is shown in Fig. 2. Given a word of length l and an alphabet of length k [2], each character in the word is first transformed to a one-hot encoding vector. These vectors are concatenated into a matrix $M \in R^{k \times l}$ for the whole word. Secondly, a matrix $N \in R^{k \times l}$ is obtained by applying a Discrete Cosine Transform along

[2] We use the digits $0-9$ and the lowercase letters $a-z$ in our experiments, that is $k = 36$.

the dimension l. Thirdly, the matrix N is cropped and keeps only r first low-frequency components, which denotes as $P \in R^{k \times r}$. Finally, the matrix P is flattened into a vector R with dimension $k \times r$. Specifically, r is set to 3 in our experiments, so the dimension of the word embedding is 108. For words with less than r characters, we pad zeros to get vectors of the same dimension.

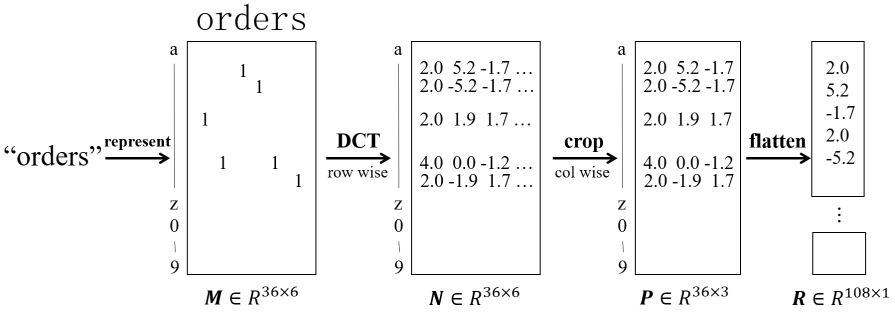

Fig. 2. Word embedding with DCToW. The word string is first represented as matrix M by one-hot encoding. Secondly M is transformed into matrix N through DCT. Thirdly, N is cropped to matrix P. Finally, P is flattened into a 108 dimensional vector R.

The embedding map with dimension $\left(N, E, \frac{H}{4}, \frac{W}{4}\right)$ is obtained by feeding f into stacked convolutional layers and a Sigmoid layer, where E corresponds to the dimension of the word embedding, namely 108. For the generation of the word embedding ground truth, all pixels in the positive word area defined by the first task are assigned to the corresponding word embedding values.

To minimize the error between the predicted word embedding $\hat{y}_{embed}$ and the ground truth y_{embed}, we use the cosine loss introduced in [30], which can be formulated as follows:

$$\mathcal{L}_{embed} = 1 - \cos\left(\hat{y}_{embed}, y_{embed}\right) \ . \tag{5}$$

Overall, the whole loss function can be written as:

$$\mathcal{L}_{all} = \mathcal{L}_{bbox} + \lambda_{cls}\mathcal{L}_{cls} + \lambda_{embed}\mathcal{L}_{embed} \ , \tag{6}$$

where $\mathcal{L}_{bbox}$, $\mathcal{L}_{cls}$ and $\mathcal{L}_{embed}$ represent the losses for word bounding box regression loss, word pixel classification loss and word embedding loss respectively. λ_{cls} and λ_{embed} balance the importance among these losses, and we set $\lambda_{cls} = 1.0$, $\lambda_{embed} = 1.0$ in our experiments.

Inference Stage At the stage of inference, the dense predictions are filtered by Non-Maximum Suppression (NMS) to yield final word bounding boxes, as shown in Fig. 1. Then, for each predicted word bounding box, we locate the

corresponding area on the multi-dimensional word embedding map and calculate the mean of this area to get the word embedding. Furthermore, the given query string is embedded into the same word embedding space to calculate the cosine distance with predicted word embeddings. The smaller the cosine distance, the greater the similarity. The top N query results with the largest similarity are considered as retrieval results.

3.3 Offline Data Augmentation

Facing the fact that training data is insufficient, we adopt the two offline data augmentation strategies introduced in [5], i.e. the in-place and the full-page augmentation. They help the model to improve the ability of learning word embedding and the accuracy of predicting word bounding box. A comparison of the two strategies is shown in Fig. 3. Given the bounding box for each word in document images, the word area is augmented as follows: random affine transformation, and random morphological dilation or erosion to fatten or thin the ink. The in-place augmentation directly iterates through each word bounding box and augments each word region in-place using the above augmentation, as shown in the left of Fig. 3. The full-page augmentation firstly crops all word areas in document images for the same basic word-level augmentation, then places them row-by-row on a background image without words, as shown in the right of Fig. 3.

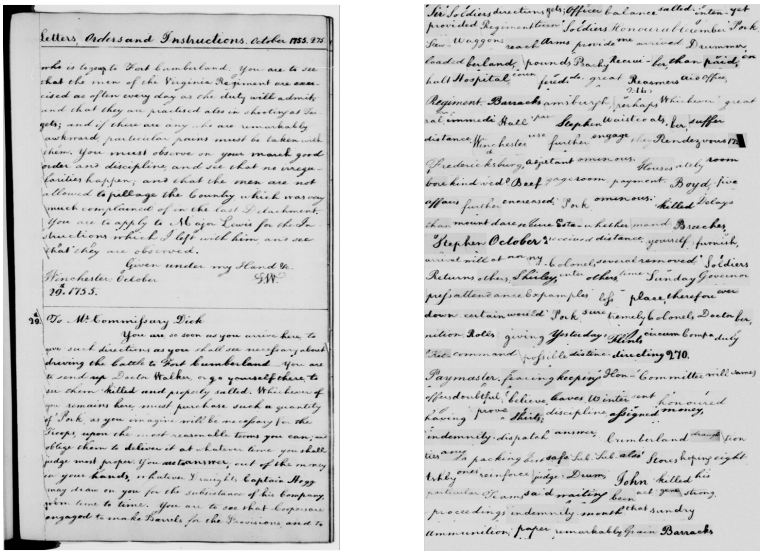

Fig. 3. A visual comparison of the in-place (left) and the full-page (right) augmentation.

4 Experiments

In this section, the datasets used in our experiments and the experimental details are first described. Next, we evaluate the proposed method on the three public benchmarks, and compare it with state-of-the-art methods. Finally, the ablation studies are presented for the proposed method.

4.1 Datasets and Experimental Setup

The proposed method is evaluated on three public benchmarks:

- George Washington Dataset (GW): The George Washington dataset [31] is written by George Washington and his secretaries in the middle of the 18th century. It comprises of 20 pages and 4860 annotated words. Due to the lack of an official partitioning into training and test pages, the 20 pages are split into a training set of 15 pages and a test set of 5 pages according to the common evaluation procedure used in [8], and take the average of four cross validations as the final results.
- Konzilsprotokolle Dataset: The Konzilsprotokolle dataset contains approximately 18000 pages in good preservation state, which includes equal copies of handwritten minutes from formal meeting held by the central administration of Greifswald University between 1794-1797. This dataset is a part of the ICFHR 2016 Handwritten Keyword Spotting Competition (H-KWS2016) [32], which contains 85 document images for training and 25 document images for testing.
- Barcelona Historical Handwritten Marriages Dataset (BH2M): The BH2M dataset [33] consists of 550,000 marriage records stored in 244 books, with marriages held between the 15th and 19th century. A subset of the dataset is used as the IEHHR2017 [34] competition dataset, where 100 images are annotated for training and 25 images for testing.

The proposed method is implemented in PyTorch framework [35], and run on a server with 2.40GHz CPU, Tesla P100 GPU, Ubuntu 64-bit OS. For the three datasets used in our experiments, we adopt the two data augmentation techniques introduced in section 3.3 to create 1000 augmented document images respectively, resulting in a total of 2000 document images. Our models are initialized with ResNet50 [11] pre-trained on ImageNet [26]. The whole network is trained end-to-end by using ADAM [36] optimizer and the learning rate is initially set to 1e-3. We train each model for 50 epochs with batch size 4, and evaluate the performance on validation dataset every 10 epochs. The model with the highest validation MAP is used for testing.

The online data augmentation for training data is listed as follows: 1) The long sides of the input images are scaled to 2048 pixels, and the short sides are scaled proportionally. 2) 512×512 random samples are cropped from the transformed images.

For the three datasets used in the experiments, the models are evaluated by adopting the standard metric used for segmentation-free word spotting, Mean

Average Precision (MAP) [5]. For the GW and BH2M datasets, the QbS evaluations use all unique transcriptions from the test set as queries. For Konzilsprotokolle we use the list of queries for QbS which are defined by the competition [32]. And we use a word classification score threshold of 0.9, a word bounding box nms overlap threshold of 0.4 and a query nms overlap threshold of 0.9.

4.2 Comparisons with State-of-the-Art Methods

As in previous work, we compare the performance of our proposed method with the state-of-the-art methods with the 25% and 50% overlap thresholds, respectively. Table 1 shows the evaluation results on the three datasets, which use the same evaluation protocol of MAP. Different from the previous two-stage method [5], the proposed method combines multi-task learning strategy with end-to-end optimization mechanism, which is the first work to utilize a single network to do segmentation-free QbS word spotting in only one stage. On GW dataset, when the overlap threshold is 50%, our method achieves a MAP of 94.06%, surpassing the state-of-the-art result (91.00%) by more than 3%. Notably, on Konzilsprotokolle dataset, the MAP (73.67%) achieved by our method is lower than [6] with a 50% overlap thresholds. The reason may be that the method in [6] adopts statistical prior knowledge to quantify the heights of word hypotheses while we do not apply this dataset-dependency strategy. The special characteristics of the dataset, such as excessive stroke overlap, also lead to performance degradation. However, when using a 25% overlap thresholds, our method achieves 98.77%, outperforming [6] over 2.77%, which clearly demonstrates that our method can detect more word regions. Fig. 4 shows the visualization results of several queries for the proposed method on the GW dataset, which proves that the proposed method can obtain precise word segmentation results.

Table 1. MAP comparison in % with state-of-the-art segmentation-free QbS methods on the GW, Konzilsprotokolle and BH2M datasets. "GW 15-5" means using the 15-5 page train/test split on GW. "25%" and "50%" are the word bounding box overlap thresholds.

Method	GW 15-5		Konzilsprotokolle		BH2M	
	25%	50%	25%	50%	25%	50%
BoF HMMs [8]	80.10	76.50	-	-	-	-
Ctrl-F-Net DCToW [5]	95.20	91.00	-	-	-	-
Rothacker et al. [6]	90.60	84.60	96.00	**89.90**	-	-
Vats et al. [9]	-	-	-	50.91	-	85.72
Resnet50 + FPN (ours)	**96.46**	**94.06**	**98.77**	73.67	**95.30**	**95.09**

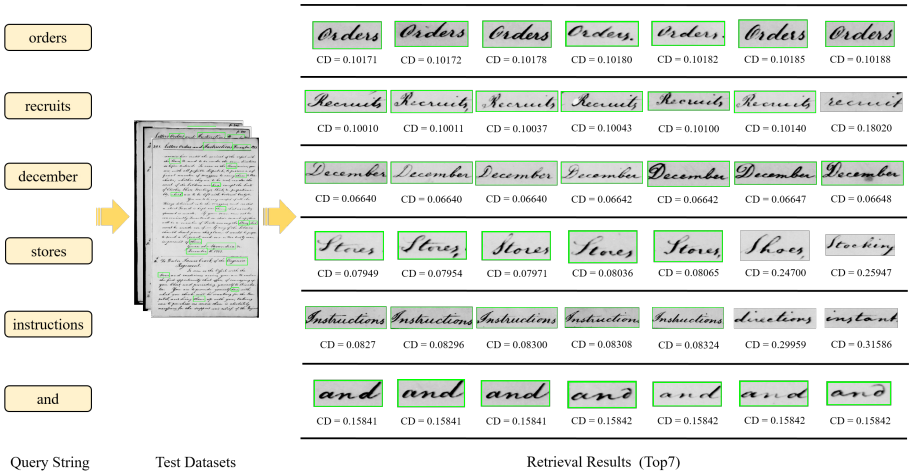

Fig. 4. The visualization results of several queries for the proposed method on GW. The figure shows the top 7 results starting from the left. The correct search results are highlighted in green. "CD" means the cosine distance between the predicted word embedding of the word area and the ground truth. The smaller the cosine distance, the greater the similarity.

4.3 Ablation Study

Influence of Feature Fusion The effect of the feature fusion is studied by extracting a single feature map at different layers and exploiting features of f_2, f_3, f_4 and f_5. The models are evaluated on GW and Konzilsprotokolle datasets. Table 2. shows that the MAP on the test datasets drops when only a single feature map is extracted. When only the low-level texture feature f_2 is extracted, the network can not learn deep information due to the lack of high-level semantic features, resulting in poor retrieval performance. When only extracting the high-level semantic feature f_5, the training can not converge very well because of the lack of low-level texture features. Considering various sizes of words in historical document images, fusing feature maps of different layers helps the network to handle word targets with different scales, which further improves the performance of word segmentation and retrieval.

Influence of the Backbone To further analyze the performance of our method, we investigate the effect of the backbone on the experimental results. Specifically, the following two network architectures are compared with the backbone used in the proposed method, Vgg16 [37] + FPN [12] and Resnet50 [11] + FCN [24]. The models are evaluated on GW and Konzilsprotokolle datasets. Table 3 shows the experimental results, from which we can find that the model with Resnet50 + FPN achieves the best performance than other backbones. This demonstrates the importance of a better backbone network for feature extraction and representation.

Table 2. MAP (%) performance evaluation of feature fusion on GW and Konzilsprotokolle. "GW 15-5" means using the 15-5 page train/test split on GW. "25%" and "50%" are the word bounding box overlap thresholds.

Method	Feature map	GW 15-5		Konzilsprotokolle	
		25%	50%	25%	50%
Resnet50 + FPN	f_5	92.47	75.59	97.54	54.43
Resnet50 + FPN	f_4	92.77	89.82	97.88	65.83
Resnet50 + FPN	f_3	91.35	87.96	97.92	66.48
Resnet50 + FPN	f_2	92.21	89.18	97.10	67.85
Resnet50 + FPN	**fusion**	**96.46**	**94.06**	**98.77**	**73.67**

Table 3. MAP (%) performance evaluation of different backbones on GW and Konzilsprotokolle. "GW 15-5" means using the 15-5 page train/test split on GW. "25%" and "50%" are the word bounding box overlap thresholds.

Backbone	GW 15-5		Konzilsprotokolle	
	25%	50%	25%	50%
Vgg16 + FPN	93.02	86.65	97.77	68.25
Resnet50 + FCN	93.00	90.19	98.38	70.78
Resnet50 + FPN (ours)	**96.46**	**94.06**	**98.77**	**73.67**

4.4 Robustness Analyze

Previous methods evaluate the experimental results only with the queries entirely from unique words in the original test set. In order to explore the robustness of the proposed method, we conduct experiments with another two query test sets, in which only the queries that appear in the training set or not in the training set are preserved. Because the queries used on the Konzilsprotokolle dataset almost totally appear in the corresponding training set, the robustness of the model is analyzed only on GW and BH2M datasets. As shown in Table 4, it can be seen from the results in the last row that the model still achieves high MAP when query words never appear in the training set, which proves the generalization and robustness of our method.

Table 4. Robustness analyze of the proposed method on GW and BH2M. "all" means using the queries totally from unique words in the test set. "only in train" means using the queries only appear in the training set. "not in train" means using the queries not appear in the training set.

Query set	GW 15-5		BH2M	
	25%	50%	25%	50%
all	96.46	94.06	95.30	95.09
only in train	97.70	94.98	96.56	96.40
not in train	93.57	91.71	92.73	92.43

5 Conclusion and Future Work

In this paper, we present an efficient end-to-end trainable model for segmentation-free query-by-string word spotting. Based on feature sharing and multi-task learning strategy, for the first time, our method simultaneously predicts word bounding box and word embedding through a single network. Experiments on word spotting benchmarks demonstrate the superior performance of the proposed method, and prove the effectiveness of segmenting words by directly regressing word bounding box in handwritten historical document images with dense words distribution and overlapping strokes.

Since labeling of training data is time-consuming, in the future, we will consider using weak supervised learning to perform word spotting task on handwritten historical document images, and applying this method to other scenarios such as natural scene images.

References

1. Giotis, A.P., Sfikas, G., Gatos, B., Nikou, C.: A survey of document image word spotting techniques. Pattern Recognition **68** (2017) 310–332

2. Sudholt, S., Fink, G.A.: Phocnet: A deep convolutional neural network for word spotting in handwritten documents. In: Proceedings of the 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), IEEE (2016) 277–282
3. Wilkinson, T., Brun, A.: Semantic and verbatim word spotting using deep neural networks. In: Proceedings of the 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), IEEE (2016) 307–312
4. Serdouk, Y., Eglin, V., Bres, S., Pardoen, M.: KeyWord Spotting using Siamese Triplet Deep Neural Networks. In: Proceedings of the 15th International Conference on Document Analysis and Recognition (ICDAR), IEEE (2019) 1157–1162
5. Wilkinson, T., Lindstrom, J., Brun, A.: Neural Ctrl-F: Segmentation-free query-by-string word spotting in handwritten manuscript collections. In: Proceedings of the International Conference on Computer Vision (ICCV). (2017) 4433–4442
6. Rothacker, L., Sudholt, S., Rusakov, E., Kasperidus, M., Fink, G.A.: Word hypotheses for segmentation-free word spotting in historic document images. In: Proceedings of the 14th International Conference on Document Analysis and Recognition (ICDAR). Volume 1., IEEE (2017) 1174–1179
7. Gómez, L., Rusinol, M., Karatzas, D.: Lsde: Levenshtein space deep embedding for query-by-string word spotting. In: Proceedings of the 14th International Conference on Document Analysis and Recognition (ICDAR). Volume 1., IEEE (2017) 499–504
8. Rothacker, L., Fink, G.A.: Segmentation-free query-by-string word spotting with bag-of-features HMMs. In: Proceedings of the 13th International Conference on Document Analysis and Recognition (ICDAR), IEEE (2015) 661–665
9. Vats, E., Hast, A., Fornés, A.: Training-free and segmentation-free word spotting using feature matching and query expansion. In: Proceedings of the 15th International Conference on Document Analysis and Recognition (ICDAR), IEEE (2019) 1294–1299
10. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems. (2015) 91–99
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR). (2016) 770–778
12. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR). (2017) 2117–2125
13. Rodríguez-Serrano, J.A., Perronnin, F.: Handwritten word-spotting using hidden Markov models and universal vocabularies. Pattern Recognition **42** (2009) 2106–2116
14. Rodríguez-Serrano, J.A., Perronnin, F.: A model-based sequence similarity with application to handwritten word spotting. IEEE transactions on pattern analysis and machine intelligence **34** (2012) 2108–2120
15. Rath, T.M., Manmatha, R.: Word image matching using dynamic time warping. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR). Volume 2., IEEE (2003) II–II
16. Rath, T.M., Manmatha, R.: Word spotting for historical documents. International Journal of Document Analysis and Recognition (IJDAR) **9** (2007) 139–152
17. Frinken, V., Fischer, A., Manmatha, R., Bunke, H.: A novel word spotting method based on recurrent neural networks. IEEE transactions on pattern analysis and machine intelligence **34** (2011) 211–224

18. Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H., Schmidhuber, J.: A novel connectionist system for unconstrained handwriting recognition. IEEE transactions on pattern analysis and machine intelligence **31** (2008) 855–868
19. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision **60** (2004) 91–110
20. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR). Volume 1., IEEE (2005) 886–893
21. Almazán, J., Gordo, A., Fornés, A., Valveny, E.: Word spotting and recognition with embedded attributes. IEEE transactions on pattern analysis and machine intelligence **36** (2014) 2552–2566
22. Zhou, X., Yao, C., Wen, H., Wang, Y., Zhou, S., He, W., Liang, J.: East: An efficient and accurate scene text detector. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR). (2017) 5551–5560
23. Wang, W., Xie, E., Li, X., Hou, W., Lu, T., Yu, G., Shao, S.: Shape robust text detection with progressive scale expansion network. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR). (2019) 9336–9345
24. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR). (2015) 3431–3440
25. Shi, B., Bai, X., Yao, C.: An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE transactions on pattern analysis and machine intelligence **39** (2016) 2298–2304
26. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems. (2012) 1097–1105
27. Milletari, F., Navab, N., Ahmadi, S.A.: V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: Proceedings of the 4th International Conference on 3D Vision (3DV), IEEE (2016) 565–571
28. Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., Savarese, S.: Generalized intersection over union: A metric and a loss for bounding box regression. In: Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR). (2019) 658–666
29. Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., Ren, D.: Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression. In: Proceedings of the 34th AAAI Conference on Artificial Intelligence. (2020) 12993–13000
30. Krishnan, P., Dutta, K., Jawahar, C.V.: Word spotting and recognition using deep embedding. In: Proceedings of the 13th International Workshop on Document Analysis Systems (DASW), IEEE (2018) 1–6
31. Lavrenko, V., Rath, T.M., Manmatha, R.: Holistic word recognition for handwritten historical documents. In: Proceedings of the 1st International Workshop on Document Image Analysis for Libraries, IEEE (2004) 278–287
32. Pratikakis, I., Zagoris, K., Gatos, B., Puigcerver, J., Toselli, A.H., Vidal, E.: ICFHR2016 handwritten keyword spotting competition (H-KWS 2016). In: Proceedings of the 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), IEEE (2016) 613–618
33. Fernández-Mota, D., Almazán, J., Cirera, N., Fornés, A., Lladós, J.: Bh2m: The barcelona historical, handwritten marriages database. In: Proceedings of the 22nd International Conference on Pattern Recognition (ICPR), IEEE (2014) 256–261

34. Fornés, A., Romero, V., Baró, A., Toledo, J.I., Sánchez, J.A., Vidal, E., Lladós, J.: ICDAR2017 competition on information extraction in historical handwritten records. In: Proceedings of the 14th International Conference on Document Analysis and Recognition (ICDAR). Volume 1., IEEE (2017) 1389–1394
35. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L.: Pytorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems. (2019) 8026–8037
36. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
37. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)