Neural Word Search in Historical Manuscript Collections

Tomas Wilkinson, Jonas Lindström, and Anders Brun

Abstract—We address the problem of segmenting and retrieving word images in collections of historical manuscripts given a text query. This is commonly referred to as “word spotting”. To this end, we first propose an end-to-end trainable model based on deep neural networks that we dub Ctrl-F-Net. The model simultaneously generates region proposals and embeds them into a word embedding space, wherein a search is performed. We further introduce a simplified version called Ctrl-F-Mini. It is faster with similar performance, though it is limited to more easily segmented manuscripts. We evaluate both models on common benchmark datasets and surpass the previous state of the art. Finally, in collaboration with historians, we employ the Ctrl-F-Net to search within a large manuscript collection of over 100 thousand pages, written across two centuries. With only 11 training pages, we enable large scale data collection in manuscript-based historical research. This results in a speed up of data collection and the number of manuscripts processed by orders of magnitude. Given the time consuming manual work required to study old manuscripts in the humanities, quick and robust tools for word spotting has the potential to revolutionise domains like history, religion and language.

Index Terms—Word spotting, Historical Manuscripts, Deep Convolutional Neural Network, Region Proposals

1 INTRODUCTION

Since the invention of writing over 5000 years ago, people have been recording a great variety of things in written form. Up until the 20th century, writing has been the primary way of storing information across generations. Consequently, manuscripts are a vital source of knowledge about the past, and for that reason of great interest to contemporary historical research. Once digitized, manuscripts are typically collected in archives [2], [3], [4]. Some are also transcribed [5], allowing for text-based searching and processing [6]. Others are collections of published data used in research [4], [6], though they are most commonly simply photographed [2]. Research in historical documents often consists of manually searching for small and scattered pieces of information in large amounts of digitized manuscripts. Finding where to look, or even which book to examine can be time consuming, as it is commonplace for a researcher to spend several months with a single book of a few hundred pages. It is a matter of looking for a black cat in a coal cellar. A need for a scalable solution exists as large part of the writings produced throughout history are yet to be studied.

Word spotting [7], [8] is a way to address the problem of finding where to look. The task consists of locating and retrieving images of words given a user supplied query, much like a regular text word search found in common text processors.

Since its introduction, two principal modes of variation on the task have been proposed. The first is input data format, where images can either be of full manuscript page, known as segmentation-free word spotting; or pre-cropped individual word images, known as segmentation-based word spotting. The second variation is the type of query. It may either be a cropped word image, also called Query-by-Example (QbE); or a text string, also called Query-by-String (QbS) or out-of-vocabulary word spotting. All else being equal, QbS
The contributions of this paper include:

1. Contributions do not directly map their visual BoVW and textual representations to tent Semantic Analysis [34] to perform multi-modal fusion, whereas [30], [31] use SIFT descriptors. In [32], Fisher Vectors [33] were built on top of SIFT descriptors.

Moreover, [32] and [31] both allow for QbS word spotting using similar approaches. In [31], the authors create a textual descriptor based on character n-grams and use Latent Semantic Analysis [34] to perform multi-modal fusion, mapping their visual BoVW and textual representations to

1.1 Contributions
The contributions of this paper include:

1) Two models for segmentation-free query-by-string word spotting are introduced: An end-to-end trainable model based on Faster R-CNN [18] and previous work [19], [20], and a simplified version that performs equally well or better in certain situations.

2) Two novel data augmentation strategies for full manuscript pages, crucial for preventing model overfitting.

3) Ablation studies that evaluate a set of model and training choices.

4) An investigation into performance of the Region Proposal Network [18] as applied to manuscript images.

5) State-of-the-art results on three benchmarks across all experiments, including some very limited data settings.

6) A case study conducted in collaboration with historians where we apply our model to a collection of 64 volumes of court records used for contemporary historical research. The result is an increase in processing speed and data size by orders of magnitude, compared to manual work.

This paper is an extension of [1], where an initial version of this work was presented. Compared to the previous incarnation we have: introduced a simplified model (Section 3.2), carried out an ablation study (Section 5.3); improved upon previous results (Section 5.4); and extended the previous case study from 1 volume of court records to 64 (Section 6), significantly increasing its scope, historical value, and complexity.

2 RELATED WORK

The work in this paper is based on, and related to, a few different fields that we will briefly review in this section.

2.1 Word spotting
Since its introduction over 20 years ago, word spotting [7] in manuscript images has come a long way. The initial approach uses template matching, with the image itself as a feature descriptor. Subsequent work introduced the idea of viewing the images as sequences of column features and applying sequence matching methods, in particular Dynamic Time Warping (DTW) [21], [22], [23]. Hidden Markov Models (HMMs) [24], [25], [26], and to a lesser extent, Recurrent Neural Networks (RNNs) [27].

However, with the size of data ever increasing, the inefficiencies of sequence-based methods were becoming prohibitive. As a result, there was renewed interest in compact, fixed-length representations that allow for fast Euclidean distance calculations. [28] build Bag-of-Visual-Words (BoVW) [29] features on top of HOG descriptors, whereas [30], [31] use SIFT descriptors. In [32], Fisher Vectors [33] were built on top of SIFT descriptors.

Moreover, [32] and [31] both allow for QbS word spotting using similar approaches. In [31], the authors create a textual descriptor based on character n-grams and use Latent Semantic Analysis [34] to perform multi-modal fusion, mapping their visual BoVW and textual representations to
the same space. In a similar vein, [32] use an attribute representation [35], [39] called Pyramidal Histogram of Characters (PHOC) as textual descriptor. They use Canonical Correlation Analysis (CCA) to learn a common subspace for the textual PHOC descriptor and visual Fisher Vectors. As it turns out, the approach in [32] proved to be the stronger system and the PHOC attribute representation has been widely adopted by the word spotting community, and has even been put to good use in lexicon-based text recognition [37].

Since then, many methods extending it have been proposed. In [38], [39], the approach in [32] is extended to the segmentation-free setting. The method proposed in [40] replaces the Fisher Vectors by a convolutional neural network (CNN) and the PHOC representation is learned using SVMs. Two methods were simultaneously proposed where the two step Fisher Vector and CCA approach is replaced with end-to-end trainable CNNs [20], [41]. This strand of work has been consolidated and improved upon in [42], [43].

A majority of the proposed handwritten word spotting methods assume that the words or text lines have been segmented, or that this is easily achieved. It turns out that for many manuscripts, especially historical ones, this is not a valid assumption. To remedy this, an increasing number of segmentation-free word spotting methods have been proposed [26], [28], [30], [39], [44], [45], [46].

Segmentation-free word spotting approaches can typically be placed into two broad categories based on how they generate region from within an image. The first are methods based on a sliding window approach [29], [28], [30], [39], [46], where regions are generated positions along a regular grid over a manuscript page. The grid is usually either on the pixel level or on top of densely extracted features. This method is common in the QbE setting as the size of the query image in pixels known, allowing for constraints on the sizes of generated regions [26], [28], [30], [39]. For QbS, the width and height of the region needs to be estimated, given the query [46]. The main drawback with these methods is the large amounts of regions generated, resulting in false positives and long processing times [45]. Additionally, due to a lack of attention, sliding window techniques are not robust against small shifts of the input, as this would lead to a misalignment av regions compared to ones extracted from the un-shifted input.

The second category are methods using connected components [38], [41], [43], [47], [48]. In [44], connected components in the shape of vertical strokes are extracted by performing mathematical morphology to separate characters. A popular approach is to binarize the image, extract connected components, and group them in a bottom-up fashion using heuristics and finally extract bounding boxes [38], [45]. A similar approach is used in [49] for matching entire documents using distributions of word images. A combination of the two approaches is used in [47]. Here, extremal regions are extracted on top of text presence scores computed using a sliding window. A related approach is applied to word segmentation in [19], where morphological closing is used using a variety of different kernel sizes to connect characters into words and extract bounding boxes. These methods still over-segment the manuscript, but the number of proposals is typically fewer than for sliding window based methods. They are also more robust to small input shifts as the connected components provide a kind of attention. The downsides include a sensitivity to physical degradations like ink blots and difficulty with densely written manuscripts.

A recent approach that can be seen as hybrid between sliding window and connected components is presented in [50]. Here, the authors use a Resnet encoder-decoder network to produce a heatmap denoting each pixels probability of being inside, outside, or near a bounding box. The heatmap is smoothed using a smoother network before being fed to a Proposal Generation network that produces bounding box coordinates, which are subsequently filtered using a proposal Filter network.

2.2 Attribute Representations and Label Embeddings

Modern word spotting methods are closely related to two common tasks in computer vision. The widely adopted PHOC representation [32] is an attribute representation where each dimension corresponds to the presence or absence of a character in a part of a word. In the case of word spotting, the attributes are then used to retrieve words with similar attributes. Attribute representations have been successfully used in zero-shot learning [35], [36].

The recently introduced embedding for word spotting DCToW [20] is similar to PHOC in that it is hand-engineered, but it does not consist of binary attributes, but a low-frequency, real-valued representation of a text string. It shares a greater similarity with the Spatial Pyramid of Characters introduced for text recognition in [31], which is similar to PHOC except that character occurrences are counted, not a binary presence/absence.

The attribute and label embedding representations allow seamless retrieval of words not present in the training data, known as zero-shot learning. This same technique is used for multi-label image classification [52], and text recognition [51]. Moreover, word spotting shares similarities with approaches for multi-modal embeddings for zero-shot learning [53], [54], with the difference that the text modality has a fixed embedding.

2.3 Scene Text Recognition

The model proposed in this paper is similar to work in end-to-end scene text detection and recognition, which has been receiving increasing attention [55], [56], [57], [58], [59]. In [55], an end-to-end system for text localization, recognition and retrieval based on region proposals and CNNs is proposed. Another approach is made in [56], where characters are detected and grouped together in a bottom-up fashion to build words and text lines. Two similar approaches [57], [58] use region proposal networks (RPNs) to generate candidates for text regions and then transcribe them. The difference lies in that [58] uses the Connectionist Temporal Classification (CTC) loss [60] to decode a region whereas [57] employs a Recurrent Neural Network (RNN). Similarly, [59] makes use of an RPN and a CTC loss, but with a novel ROI-Rotate operation that maps arbitrarily oriented region proposals to
Fig. 2. The Ctrl-F-Net model at test time. Given an input image, it is fed through the first CNN of the model and Dilated Text Proposals (DTP) are extracted. These are then fed into the localization layer, where additional text proposals are extracted using a Region Proposal Network (RPN), followed by non-max suppression. The RPN proposals are added to the DTP proposals and fed through the Bilinear Interpolation layer, giving a wordness score, and a descriptor is extracted. Finally, a second non-max suppression is applied resulting a large number of region proposals, typically 2-3 times the number of ground truth boxes.

axis-aligned feature maps. Our proposed models are similar to the RPN-based scene text recognition models, where the greatest difference lies in that we learn to embed word images in a word embedding space where they perform text recognition.

3 The Models

In this section we introduce the two models used in this paper. The first is the Ctrl-F-Net that we introduced in the previous version of this paper [1]. The second model is a simplified version of Ctrl-F-Net we call Ctrl-F-Mini. In section 3.3, we provide results from a series of ablation studies investigating the performance of the two models.

3.1 Ctrl-F-Net

The model we propose is a deep convolutional neural network inspired by previous work on object detection [18], dense image captioning [61], and segmentation-based word spotting [20]. We call it Ctrl-F-Net, named after the well known shortcut for word search in many word processors. It is an end-to-end trainable model that simultaneously proposes and scores word candidate region proposals and embeds them into a word embedding space, wherein a search can be performed. The input to the model is a full manuscript page. The output is set of bounding box region proposals: \( N_1 \times 4 \) CNN CNN \( 128 \times H/8 \times W/8 \) 1 x H x W N1+N2 x 512 box reg embed NMS

ResNet [62] is used as the CNN architecture due to its small memory footprint while still achieving high performance. The feature maps are then fed through a localization module, that consists of: i) an RPN that generates region proposals and corresponding scores; ii) A non-max suppression (NMS) step to remove redundant, overlapping proposals; iii) A resizing layer produces fixed sized outputs given variable sized inputs; iv) loss functions for the region proposals coordinates and scores.

The input feature maps are first fed to an RPN, with one branch that regresses K=15 anchor boxes using a convolutional layer with \( 4K \) output channels. To accommodate the varying aspect ratios of words, the anchor boxes have the sizes \( \{20, 40, 60\} \times \{30, 90, 150, 210, 300\} \), in pixels. A second parallel branch predicts wordness scores for each box, that represent the probability that a box is situated atop a word. The boxes and scores are then combined and NMS is applied. The RPN can be seen to fall into the category of sliding window based segmentation-free word spotting methods. Proposals with an Intersection-over-union (IoU) overlap with a ground truth bounding box greater than 0.75 are considered as positives and negatives are defined as proposals with an IoU lower than 0.4. Proposals that fall between are ignored. A total of 256 proposals, 128 positives and negatives, are sampled and used to calculate the mid-network wordness and regression losses.

The boxes of varying size sampled from the RPN are then resized using Bilinear Interpolation [61], [63] to a fixed output size of \( 8 \times 20 \) pixels. They are then fed through the rest of the CNN and used as input to three parallel branches. The first branch is fully connected (FC) layer with 4 outputs that refines the box coordinates by regressing them once again. Similarly, the second branch is also a FC layer with a single output that predict the final wordness scores. The third branch is a small FC embedding network with 2 hidden layers, that does the final embedding.

For the mid-network scores and output wordness scores, we use a binary logistic loss. The bounding boxes are parameterized according to [64], both for the anchor box regression
and the output box regression. The boxes are represented as the quadruples \((x_c, y_c, w, h)\), where \(x_c\) and \(y_c\) are the center of a box and \(w\) and \(h\) is its width and height. The functions to learn are normalized translation offsets for \(x\) and \(y\) and log-space scaling factors for \(w\) and \(h\). The loss is a smooth \(L_1\) loss, also known as a specialized version of the Huber loss \([25]\):

\[
L_{\text{reg}}(x_i, t_i) = \begin{cases} 
\frac{1}{2}(x_i - t_i)^2 & \text{if } |x_i - t_i| < 1 \\
|x_i - t_i| - \frac{1}{2} & \text{if } |x_i - t_i| \geq 1
\end{cases}
\]

where \(x_i\) is one of \(\{x_c, y_c, w, h\}\) and \(t_i\) is its corresponding target.

The embedding branch is a fully-connected network with two hidden layers of size 4096, with batch normalization \([66]\) after each layer followed by the hyperbolic tangent activation function. The final layer is an \(l^2\)-normalization layer. It only receives the regions labelled positive as input.

As loss function for the embedding network, we use the Cosine Embedding loss that has successfully been used in segmentation-based word spotting \([20]\). It is defined as

\[
L_{\text{emb}}(u, v, y) = \begin{cases} 
1 - \frac{u^T v}{\|u\|\|v\|} & \text{if } y = 1 \\
\max(0, \frac{u^T v - \gamma}{\|u\|\|v\|}) & \text{if } y = 0
\end{cases}
\]

where \(v\) is an embedding of a positive region proposal and \(u\) is a ground truth embedding. If \(y = 1\), \(v\) and \(u\) match, and they are moved closer together. If \(y = 0\), they do not match and \(v\) and \(u\) are moved further apart.

The total loss function is a weighted linear combination of the five losses.

\[
L_{\text{tot}} = 10^{-2}\cdot(L_{\text{rpn-reg}} + L_{\text{rpn-score}}) + 10^{-1}\cdot(L_{\text{reg}} + L_{\text{score}}) + 3\cdot L_{\text{emb}}
\]

### 3.2 Ctrl-F-Mini

In order to evaluate different model choices, in particular the source of region proposals, we introduce the Ctrl-F-Mini model. The main difference to the full model is the removal of the region proposal network, which in turn leads to other parts that are no longer relevant. From the localization module, only the bilinear interpolation remains. Towards the end, the end box scoring and embedding branches are kept. Ctrl-F-Mini is trained using external region proposals. The mid-network box scoring and regression losses and the end box regression loss is removed. The reduced model can be seen in Figure 4. This results in a greatly simplified model, which takes about a quarter of the time to train. Furthermore, inference becomes a lot faster as fewer region proposals are used, and the computations related to the RPN (in particular the non-max suppression over several hundred of thousand proposals) are not performed.

### 3.3 Querying

During inference, a manuscript pages and \(N_1\) optional external region proposals are fed through the model that outputs: an \(N \times 4\) matrix of region proposals; an \(N \times D\) matrix of descriptors, where \(D\) is the dimensionality of the word embedding that is used; and an \(N\)-dimensional vector of wordness scores, where for Ctrl-F-Net \(N = N_1 + N_2\). We typically set \(N_2 = N_1\) on a page-by-page basis (\(N = N_1\) Ctrl-F-Net mini). We then threshold the wordness scores, only keeping proposals with a score \(> t_{\text{th}}\), followed by an NMS step using an overlap threshold \(t_{\text{nms}}\).

Once a query is selected, either by cropping a part of an image for QBE, or providing a search query for QBS, it is first transformed to the word embedding space. Then the cosine distance is used to compare the query to each region proposal and they are sorted w.r.t. their similarity to the query. Using the similarity the query as a score, we perform a final NMS step with an overlap threshold set to 0.

### 3.4 Word Embeddings

In the recent word spotting literature based on using word embeddings, two have been most successful, see Figure 4. The first, and by far most popular, is the Pyramidal Histogram of Characters, or PHOC \([32], [42], [43], [48]\). Provided a text string, the number of pyramid levels, and an alphabet of length \(K = 36\) (we use the digits 0-9 and lower-case letters for all experiments), construct a binary occurrence vector for each sub-word in each level of the
The resulting vector $u_p$ is a $36 \cdot (1 + 2 + 3 + 4 + 5) = 540$ dimensional binary vector, i.e., $u_p \in \{0, 1\}^{540}$. The earliest papers using PHOC augmented the alphabet with the most common bi-grams for a particular language. Subsequent work have achieved better results without them while keeping the embedding language agnostic, and we do the same.

The second embedding is the Discrete Cosine Transform of Words (DCToW), recently introduced in [20]. It is a low frequency, distributed representation of a word, that has recently achieved state-of-the-art results in segmentation-based word spotting. Given a word of length $m$ and an alphabet of length $K$, first build a $m \times K$ one-hot matrix representation. Then apply the Discrete Cosine Transform (DCT) to each row of the matrix. Finally, keep the first $r$ components of the DCT, and flatten the $m \times K$ matrix into an $r \cdot K$ dimensional vector, $u_d$. Following [20], we set $r = 3$ making $u_d \in \mathbb{R}^{108}$. Words that are shorter than $r$ characters are padded with zeros to get the correct length.

### 3.5 Dilated Text Proposals

The ideal case for a segmentation-free word spotting system is maximizing the recall while keeping the number of proposals as low as possible. Referring back to the distinctions made in section 2.1, the RPN would fall into the category of sliding window approaches. As such, a likely improvement of the recall can be achieved by using a complementary external region proposal method based on connected components. We use the approach from [19], which we call Dilated Text Proposals (DTP) for sake of clarity. Given a grayscale image, DTP first creates a set of $j$ binary images by thresholding at $j$ different multiples of the image mean value. Then applies morphological closing to each binary image using a set of $l$ generated rectangular kernels. For each of the $j \cdot l$ images, find the connected components, then extract bounding boxes for each connected component and remove duplicate boxes.

### 4 Data Augmentation

As we are operating in a small data setting (as few as 5 manuscript pages for training), data augmentation is crucial to prevent severe overfitting on the training data. We propose two complementary ways of augmenting the entire manuscript pages that we call full-page and in-place augmentation. The two techniques are visually compared in Figure 5.

Full-page augmentation allows to have control over the distribution of classes, which is important for learning a discriminative word embedding. It works by uniformly sampling word images from the training set, augmenting them, and placing the row-by-row on background canvas. We adopt the affine and grayscale morphology augmentation from [20]. The canvas is created by uniformly sampling a background colour from an interval centered on the median of all images in the training set and adding on some Gaussian noise. The finished augmented page looks like left-aligned manuscripts of randomly sampled word images.

In-place augmentation is designed to keep the overall look of the page intact, while still providing some useful variation in writing style. Ideally, this helps the model generate and score region proposals, while still providing variation for the learning word embeddings, although without control of class distributions. For a given manuscript page, we iterate through the ground truth bounding boxes and augment each word image in-place. We apply a shearing transform and followed by grayscale morphological dilation or erosion while ensuring the output has the same size as the input so that it can be slotted back into place.

### 5 Experiments

Here, we perform the main quantitative evaluation of our models, including an ablation study, on two widely used benchmarks for word spotting.

The George Washington (GW) Dataset [67] was written in English the middle of the 18th century by George Washington and his secretaries. It consists of 20 pages, or 4860 words. We follow the evaluation procedure used in [46], by using two different splits of the pages into training, validation and test sets. For the first split called GW 15-5, has a training set of 15 pages and 5 for testing. The second is a 5-15 split with 5 training and 15 test pages. In both cases, we use 1 page as a validation set. The reported results
are the average of four cross validations. The bounding boxes for the GW are manually annotated, resulting in a large amount of extra space around 1-2 character words, causing a significant decrease in recall for higher overlap thresholds. To counteract this annotation issue, we pad the DTP proposals with 10 pixels for this dataset.

The IAM Offline Handwriting Dataset [68] is a modern cursive dataset consisting of 1539 pages, or 115320 words, written by 657 writers. We use the official train/val/test split for writer independent text line recognition, where there is no writer overlap between the different splits. Following standard protocol, we remove stop words from the set of queries, and in line with [32], queries that come from lines that are marked as containing segmentation errors are removed. Ground truth boxes that are so small that they collapse to a width or height of zero when downsampled by a factor 8 are also removed.

5.1 Training

The models are trained end-to-end in a single phase. We first train a model (with weights initialized randomly) using the synthetic IIIT-HWS-10k dataset [49]. Since it only consists of word images, we use the full-page augmentation technique to create 3000 synthetic document images. This model was used to initialize all other models.

For the other datasets, we create 5000 augmented images, split evenly between in-place and full-page augmentation, and add them to the original data. The input image is rescaled such that its longest side is 1720. We train each model for a maximum of 25000 iterations, and measure the performance on a held out validation set every 1000 iterations. The model with the highest validation MAP score is used for testing. The learning rate is initially set to $10^{-3}$ for all models (except for Ctrl-F-Net on IAM which starts at $2 \cdot 10^{-4}$) and is multiplied every 10000 iterations by 0.1. We use ADAM [69] to update the weights. Our implementation is in Pytorch [70] and training time is approximately 10 hours for Ctrl-F-Net and 3 hours for Ctrl-F-Mini on an NVIDIA Titan GTX.

5.2 Evaluation

For the GW and IAM datasets, we evaluate our model using the standard metric used for word spotting, Mean Average Precision (MAP), where the Average Precision for a collection of size $N$ is defined as

$$AP = \frac{\sum_{k=1}^{N} P_k \cdot r_k}{R}$$  
(4)

where $P(k)$ is the precision measured at cut-off $k$ in the returned list, $R$ is the number of relevant results, and $r(k)$ is an indicator function that is 1 if a returned result at rank $k$ is relevant, and 0 otherwise. A retrieved word is considered relevant if its IoU overlap with a ground truth box is greater than a threshold $t_o \in \{0.25, 0.5\}$ and the label matches the query. The MAP score is the mean of the AP over the set of queries

$$MAP = \frac{\sum_{q=1}^{Q} AP(q)}{Q}$$  
(5)

where $Q$ is the number of queries. For the QbE evaluation, all the ground truth segmented word images in the test set is used. For QbS, all unique ground truth labels are used. Some methods, in particular [35, 39, 48], use a slightly different protocol for the GW dataset. Here all word instances in the dataset are used as queries for QbE, and all unique labels for QbS. The search is performed in all 20 pages. We perform a grid search over the score NMS overlap threshold $t_{nms}$, score threshold, and RPN score NMS overlap threshold when applicable.

5.3 Ablation and analysis

To perform the ablation study, we adopt the methodology of using baseline model settings and changing one setting at a time, and always comparing with the baseline. We investigate the quantitative performance of various model choices, with the most significant being the source of region proposals (RPN, DTP, or both). Other experiments include comparing the PHOC and DCToW and three embedding loss functions. Recent work [71] argues that the Cosine loss [52] has outperformed other common loss functions for segmentation-based word spotting, including the Cosine Embedding loss. We evaluate this loss in the segmentation-free setting. The Cosine loss is defined as

$$L(u, v) = 1 - \frac{u^T v}{||u|| \cdot ||v||}$$  
(6)

where $v$ is an embedding of a positive region and $u$ is the embedding of the ground truth label. It is the part of the Cosine Embedding loss (Equation 2) where $y = 1$. We also evaluate the Binary Cross Entropy (BCE) loss, which models the embedding as a multi-label binary classification problem, and is a common choice of loss function for the PHOC embedding. As the BCE loss requires a binary embedding, it is not applicable to the DCToW. We use the GW 15-5 and IAM datasets to evaluating the different model choices.

The top section of Table 1 contains the results from [1], which are a bit different from the new baselines. Since [1] was published, we discovered a few mostly small bugs in our code, the most notable one is that the margin $\gamma$ for the Cosine Embedding loss was actually 0.2 instead of the reported 0.1, and the learning rate was accidentally multiplied by 0.1 in the first iteration of training, making the initial learning rate $2^{-4}$ instead of $2^{-3}$.

The first variant we evaluated is using the Cosine loss instead of the Cosine Embedding loss. Although the results are not unanimous, the trend suggests that the Cosine loss is superior. This corroborates the findings in [71] that the Cosine loss is better for word spotting, and coupled with the fact that it is simpler makes the Cosine loss a better choice compared to the Cosine Embedding loss. From our experiments, the Binary Cross Entropy loss works very well on the QbE setting, notably getting the highest MAP score on IAM by a good margin, around 3% for 50% IoU. However, it underperforms when it comes to QbS.

The second set of experiments in Table 1 involve using both proposals from the RPN and DTP during training the Ctrl-F-Net. This is implemented as sampling both positive and negative boxes from the proposal pool from the DTP.
TABLE 1
Ablation results for different model variants the GW 15-5 and IAM datasets. Recall is calculated based on the proposals left after the final NMS stage.

| Model Variant | Embedding | GW 15-5 MAP 50% | GW 15-5 MAP 25% | GW 15-5 Recall | IAM MAP 50% | IAM MAP 25% | IAM Recall |
|---------------|-----------|-----------------|-----------------|----------------|-------------|-------------|------------|
| Baselines from [1] | Ctrl-F-Net DCToW | 90.5 | 91.0 | 97.0 | 95.2 | 99.4 | 99.9 | 72.0 | 80.3 | 74.1 | 82.5 | 98.1 | 98.9 |
| Ctrl-F-Net PHOC | 90.9 | 90.1 | 96.7 | 93.9 | - | - | 71.5 | 78.8 | 73.7 | 80.8 | - | - |
| Baselines | Ctrl-F-Net DCToW | 91.9 | 92.9 | 96.8 | 95.7 | 93.8 | 99.5 | 73.9 | 81.9 | 77.2 | 85.3 | 97.6 | 98.5 |
| Ctrl-F-Net PHOC | 92.5 | 93.4 | 96.9 | 96.2 | 95.6 | 99.0 | 73.9 | 83.2 | 75.8 | 85.1 | 93.2 | 94.3 |
| Ctrl-F-Net PHOC | 90.8 | 90.5 | 97.1 | 95.3 | 96.7 | 99.8 | 74.6 | 83.0 | 78.6 | 87.4 | 98.0 | 98.9 |
| Ctrl-F-Mini PHOC | 92.2 | 92.9 | 97.1 | 96.1 | 96.1 | 99.2 | 74.6 | 84.9 | 77.0 | 87.0 | 93.1 | 94.2 |

TABLE 2
Recall comparison in %, averaged over pages between the region proposal network and dilated text proposals using Ctrl-F-Net with the DCToW embedding. Filtered refers to the score thresholding and non-max suppression steps described in section 3.3.

| Method | GW 15-5 Filtered 50% | GW 15-5 Filtered 25% | GW 5-15 Filtered 50% | GW 5-15 Filtered 25% | IAM Filtered 50% | IAM Filtered 25% |
|--------|-----------------------|-----------------------|-----------------------|-----------------------|------------------|------------------|
| Ctrl-F-Net DCToW | 91.8 | 93.4 | 96.6 | 95.1 | 95.1 | 99.4 |
| Ctrl-F-Net PHOC | 91.7 | 92.9 | 96.9 | 96.0 | 94.5 | 99.3 |
| Ctrl-F-Mini PHOC | 91.6 | 91.8 | 97.0 | 96.2 | 96.5 | 99.2 |

Across all experiments, the DCToW and the PHOC seem to work best on the GW and IAM datasets respectively. This suggests that it is best to evaluate both embeddings for the task at hand. All else being equal, the DCToW is preferable due to its smaller dimensionality, approximately a fifth of the PHOC. Finally, analysing the Ctrl-F-Net and Ctrl-F-Mini, the latter performs at a similar if not higher level compared to the former over all experiments. This comes at the cost of a slightly lower rate of recall, most notably on the IAM dataset.

The small difference in recall between the Ctrl-F-Net (both RPN and DTP) and Ctrl-F-Mini (only DTP) models merits further investigation into the contributions of the RPN and DTP proposals in the recall of Ctrl-F-Net. Table 2 shows the recall rates of the two sources of region proposals, the RPN and the DTP, and their union using the baseline DCToW model on 15-5 and 5-15 GW datasets and the IAM dataset. The recall is the average over pages, and the number of proposals for each method is held the same. We evaluate the recall before and after filtering, which refers to the score thresholding and non-max suppression from Section 3.3.

Before the filtering step, the recall is practically equal on the GW dataset. For the IAM dataset on the other hand, and RPN separately, and concatenating them before continuing the forward pass. Towards the end of the model, both sets of proposals are used for the box scoring and embedding losses, but only the RPN proposals are used for the end box regression loss. Although the results are mixed, this modification seems to give an improvement for the MAP using 25% IoU.

To investigate the quality of the proposals w.r.t. MAP score, we only use RPN proposals to retrieve words from the manuscripts. There is a noticeable drop in performance for both GW and IAM datasets with this setup. Comparing proposal quality, the DTP only model i.e., Ctrl-F-Mini, clearly outperforms the RPN only model. Similarly, adding DTP proposals to the Ctrl-F-Net increases the MAP score by a good amount.

Across all experiments, the DCToW and the PHOC seem to work best on the GW and IAM datasets respectively. This suggests that it is best to evaluate both embeddings for the task at hand. All else being equal, the DCToW is preferable due to its smaller dimensionality, approximately a fifth of the PHOC. Finally, analysing the Ctrl-F-Net and Ctrl-F-Mini, the latter performs at a similar if not higher level compared to the former over all experiments. This comes at the cost of a slightly lower rate of recall, most notably on the IAM dataset.

The small difference in recall between the Ctrl-F-Net (both RPN and DTP) and Ctrl-F-Mini (only DTP) models merits further investigation into the contributions of the RPN and DTP proposals in the recall of Ctrl-F-Net. Table 2 shows the recall rates of the two sources of region proposals, the RPN and the DTP, and their union using the baseline DCToW model on 15-5 and 5-15 GW datasets and the IAM dataset. The recall is the average over pages, and the number of proposals for each method is held the same. We evaluate the recall before and after filtering, which refers to the score thresholding and non-max suppression from Section 3.3.

Before the filtering step, the recall is practically equal on the GW dataset. For the IAM dataset on the other hand,
there is a noticeable gap between the two sources of region proposals. Post filtering, the gap widened considerably for the IAM dataset and a chasm appeared between the RPN and DTP on the GW dataset. Considering that it is the model that is scoring the proposals from the two sources, and the model was only trained on RPN proposals, for the model to favour the DTP heavily suggests that the DTP generates far superior proposals. This is further corroborated by the MAP results in Table 1, where there is a significant drop in score and that the Ctrl-F-Mini is performing on a similar level to Ctrl-F-Net.

In the next section we compare our best models from the ablation study with the state of the art. For those experiments we adopt the Cosine loss instead of the Cosine Embedding loss, and for Ctrl-F-Net we make use of DTP proposals during training.

5.4 State of the art comparison
In this section, we compare the best performing models determined from the ablation study in the previous section to the state-of-the-art in segmentation-free and segmentation-based word spotting for the GW and IAM datasets.

Table 3 shows the results of the top Ctrl-F-Net and Ctrl-F-Mini models with the DCToW and PHOC embeddings, and contrasts them with the state of the art on the GW and IAM datasets. For the GW dataset, we outperform the other methods by a large margin, in both QbE and QbS and across the 25% and 50% overlap thresholds. The difference in MAP score between the GW 5-15 and GW 15-5 datasets are relatively small, considering the number of training and validation pages (5 vs 15) and test pages (15 vs 5). Our results that are marked with an asterisk use the evaluation protocol from [38], [39], [48].

In Table 4, we compare the best segmentation-free setup with a 25% overlap threshold with state of the art methods for segmentation-based word spotting, that use the same evaluation protocol. We observe that we have competitive results for the GW 15-5 split in both QbE and QbS, whereas for IAM the QbS performance is quite close the best segmentation-based methods, even though they depend on manually segmented bounding boxes.

5.5 Discussion
We make several observations from the experiments. The first is that we get an increase in the MAP score across the board from lowering the overlap threshold to 25%. This suggests that there is further performance to be gained from more accurate proposals. Another possible explanation is that this effect is due to tightness of the annotation for a

### Table 3

| Method         | Embedding | GW 15-5 MAP 50% | GW 15-5 MAP 25% | IAM MAP 50% | IAM MAP 25% |
|----------------|-----------|-----------------|-----------------|-------------|-------------|
| Ctrl-F-Net     | DCToW     | 90.9            | 91.7            | 97.0        | 96.2        |
| Ctrl-F-Net     | PHOC      | 90.7            | 92.0            | 97.3        | 96.4        |
| Ctrl-F-Mini    | DCToW     | 92.5            | 93.5            | 97.2        | 96.3        |
| Ctrl-F-Mini    | PHOC      | 91.6            | 91.8            | 97.0        | 96.2        |
| BoF HMMs       | n/a       | -               | -               | 76.5        | -           |
| AAM+SIFTquant  | PHOC      | 81.6            | 84.6            | 92.0        | 90.6        |
| Encoder-Decoder Net | PHOC    | -               | -               | -           | -           |

### Table 4

| Method                      | GW      | IAM      |
|-----------------------------|---------|---------|
| Ctrl-F-Net PHOC             | 97.3    | 78.0    |
| Embed attributes†           | 93.0    | 55.7    |
| DCToW†                      | 98.0    | 77.0    |
| TPP-PHOCNet (CPS)†          | 98.0    | 82.7    |
| DeepEmbed†                  | 98.0    | 90.4    |

Note that methods marked with † use on pre-segmented word images.
dataset. This means that for single letter words like “I”, the amount of space surrounding the ground truth box makes a region proposal that tightly attends to the ink overlapping less than 50%. This effect is greater for manually labelled datasets like the GW datasets, and less so for the IAM dataset, which is reflected in the results. For word search in manuscripts, 25% overlap would suffice as the user would in any case manually inspect each result.

The results in Table 4 show a minor difference in performance between the segmentation-free and segmentation-based word spotting methods. The segmentation-based method most similar to this work is [20], and the overall performance of the Ctrl-F-Net is higher. This suggests that there might be possible upsides to using a segmentation-free approach when learning a representation for a word image. For example, the increased consistency of automatic region proposal methods compared to manual labeling could be beneficial for learning.

A result that is of great significance for practical adoption of this work in historical research is the high performance on GW 5-15 dataset, where we train on 4 pages, validate on 1 page and test on 15 pages. This suggests that the model is learning efficiently with respect to the amount of training data. This is crucial when working with historical manuscripts, which are very expensive to annotate due to the expert knowledge required. An important aspect of training on such small data is extensive use of data augmentation. To that end we have introduced two complementary data augmentation techniques for full page manuscripts to facilitate the learning of proposed models.

From the experiments presented in the previous section, we conclude that using the RPN as a main source of region proposals without any external region proposals is a suboptimal approach. While there has been some work adapting the RPN for text proposals [72], two of the suggested improvements relate to handling arbitrarily oriented text and increasing recall. Both of these are not issues here, in fact, the RPN has higher recall than the DTP. A third adaptation of the RPN is changing the aspect ratio of the anchor boxes, which has been adopted our formulation. Instead, the results suggests that the RPN, while generic in its current form, could be improved upon or specialized for sub-domains of computer vision. One example relevant for this work would be how to generate proposals that work well for word spotting.

Furthermore, the ablation study showed how the Ctrl-F-Mini performed on par with the full Ctrl-F-Net, and in some instances even outperformed the full model. This held true also in the small data setting of GW 5-15. Considering the reduced model complexity, and decreased training and inference times Ctrl-F-Mini provides over Ctrl-F-Net, it is a recommended alternative to the full model when it comes to easily segmented data. However, the Ctrl-F-Mini with DTP proposals still suffers from the known limitations of connected components based methods for segmenting. So for more densely written, noisy manuscripts where document binarization is difficult and large parts of the text is one connected component, the full Ctrl-F-Net with its sliding window RPN would be recommended.

6 CASE STUDY: EARLY MODERN COURT RECORDS

Court records are often used in historical research. Their usefulness stretches far beyond the study of crime and judicial systems as they offer insights into practices and mentalities of ordinary people that very few sources do. In the words of a well-known introduction to historical methodology, ”court records are probably the single most important source we have for social history of the medieval and early modern periods.” [73]. Not surprisingly, several of the most famous and influential historical studies are based on court record [24], [25], [26]. At the same time, the richness in information and the variety of subjects dealt with in a single volume make the records difficult to work with. Swedish court records mix criminal and civil cases and they often lack even simple search tools such as indexes. Finding relevant information is extremely time-consuming and researchers need often to restrict their empirical research to a very limited number of volumes.

The Snevringe Court Records consists of 64 volumes of newly digitized court records from the magistrate court of Snevringe judicial district, written between 1719 and 1880. The 64 volumes consist of 55k images, each of which contain 2 pages. Figure 6 shows a sample of four images of the dataset. The court records provide several challenges: unlike modern text, there is no standardized spelling; said spelling evolves over time, compounding the problem; and there are hundreds of different writers, adding their personal variation. A final peculiarity of the court records is that they are written during a time in Sweden where a change of alphabet took place. Earlier volumes use Kurrent script (or German cursive), with particular words written in Latin cursive. A gradual, non-linear change of scripts to Latin cursive occurs over the time span that the court records were written. This dual alphabet provides interesting challenges where certain characters are written in two completely different ways in the same dataset. The queries we evaluate are chosen according to their relevance to contemporary historical research.

We have manually annotated 11 pages for training, where 3 are from the Snevringe set of court records (these
The value of word spotting for research in historical manuscripts cannot be overstated. The work often entails manually locating small pieces of information scattered throughout large amounts of texts, and finding only where to look can be very time consuming. In effect, limited sets of elusive data are what a historians’ interpretations are based on, and a limiting factor when it comes to which inquiries can be conducted at all. Speeding up the process of identifying relevant sections in handwritten texts would not only make it possible to gather more data, but also make way for new questions to be researched.

7 Conclusion

We have introduced Ctrl-F-Net, an end-to-end trainable model for segmentation-free query-by-string word spotting. It simultaneously produces region proposals, and embeds them into a word embedding space in which searches are performed. Using an ablation study, we investigate several model choices, most notably the source of region proposals. The ablation leads us to propose the simplified Ctrl-F-Mini, a model suited to manuscripts that easily segmented into words. We outperform the previous state-of-the-art approaches for segmentation-free word spotting, in some cases by a large margin. Moreover, in a case study applying the Ctrl-F-Net to a collection of 64 volumes of court records, spanning the most of the 18th and 19th centuries, we enable a historical study using orders of magnitude greater data than would be possible otherwise.

Acknowledgments

This project is a part of q2b, From quill to bytes, which is a digital humanities initiative sponsored by the Swedish Research Council (Dnr 2012-5743), Riksbankens Jubileumsfond (NHS14-2068:1) and Uppsala university.
[44] Y. Leydier, F. Lebourgeois, and H. Empozy, “Text search for medieval manuscript images,” Pattern Recognition, vol. 40, no. 12, pp. 3552–3567, 2007.

[45] A. Kovalchuk, L. Wolf, and N. Dershowitz, “A simple and fast word spotting method,” in Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on. IEEE, 2014, pp. 3–8.

[46] L. Rothacker and G. A. Fink, “Segmentation-free query-by-string word spotting with bag-of-features hms,” in Document Analysis and Recognition (ICDAR), 2015 13th International Conference on. IEEE, 2015, pp. 661–665.

[47] L. Rothacker, S. Sudholt, E. Rusakov, M. Kasperidus, and G. A. Fink, “Word hypotheses for segmentation-free word spotting of historic document images,” in Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on, vol. 1. IEEE, 2017, pp. 1174–1179.

[48] G. Axler and L. Wolf, “Toward a dataset-agnostic word segmentation method,” in 2018 25th IEEE International Conference on Image Processing (ICIP). IEEE, 2018, pp. 2635–2639.

[49] J. A. Rodriguez-Serrano, A. Gordo, and F. Perronnin, “Label embedding: A frugal baseline for text recognition,” International Journal of Computer Vision, vol. 113, no. 3, pp. 193–207, 2015.

[50] F. Chollet, “Information-theoretical label embeddings for large-scale image classification,” arXiv preprint arXiv:1607.05691, 2016.

[51] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov et al., “Devise: A deep visual-semantic embedding model,” in Advances in Neural Information Processing Systems, 2013, pp. 2121–2129.

[52] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng, “Zero-shot learning through cross-modal transfer,” in Advances in neural information processing systems, 2013, pp. 935–943.

[53] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, “Reading text in the wild with convolutional neural networks,” International Journal of Computer Vision, vol. 116, no. 1, pp. 1–20, 2016.

[54] L. Neumann and J. Matas, “Real-time lexicon-free scene text localization and recognition,” IEEE transactions on pattern analysis and machine intelligence, vol. 38, no. 9, pp. 1872–1885, 2016.

[55] H. Li, P. Wang, and C. Shen, “Towards end-to-end text spotting with convolutional recurrent neural networks,” in The IEEE International Conference on Computer Vision (ICCV), Oct 2017.

[56] M. Bista, L. Neumann, and J. Matas, “Deep textspotter: An end-to-end trainable scene text localization and recognition framework,” in The IEEE International Conference on Computer Vision (ICCV), Oct 2017.

[57] X. Liu, D. Liang, S. Yan, D. Chen, Y. Qiao, and J. Yan, “Fots: Fast oriented text spotting with a unified network,” in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018.

[58] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” in Proceedings of the 23rd international conference on Machine learning. ACM, 2006, pp. 369–376.

[59] J. Johnson, A. Karpathy, and L. Fei-Fei, “Densecap: Fully convolutional localization networks for dense captioning,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016.

[60] K. He, X. Zhang, S. Ren, and J. Sun, “Identity mappings in deep residual networks,” in European Conference on Computer Vision. Springer, 2016, pp. 630–645.

[61] M. Jaderberg, K. Simonyan, A. Zisserman et al., “Spatial transformer networks,” in Advances in Neural Information Processing Systems, 2015, pp. 2017–2025.

[62] R. Girshick, “Fast r-cnn,” in Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1440–1448.

[63] P. J. Huber et al., “Robust estimation of a location parameter,” The annals of mathematical statistics, vol. 35, no. 1, pp. 73–101, 1964.

[64] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” arXiv preprint arXiv:1502.03167, 2015.