Word Spotting Using Convolutional Siamese Network

Berat Kurar Barakat, Reem Alasam and Jihad El-Sana
The Department of Computer Science
Ben-Gurion University of the Negev
Beer Sheva, Israel
berat@post.bgu.ac.il, rym@post.bgu.ac.il, el-sana@cs.bgu.ac.il

***Abstract*—We present a method for word spotting using convolutional siamese network. A convolutional siamese network employs two identical convolutional network to rank similarity between two input word images. Once the network is trained, it can then be used to spot not just words with varying writing styles and backgrounds but also to spot out of vocabulary words that are not in the training set. Experiments on the historical Arabic manuscript dataset VML, and on the George Washington dataset shows comparable results with the state of the art.**

***Keywords*-Historical document image analysis; word spotting; deep learning; convolutional siamese network**

I. Introduction

Respected library and archives around the world are digitizing their collections to reach large audience. Efficient search in digital documents is an important prerequisite for retrieving information. Using Optical Character Recognition (OCR) for handwritten historical documents is inefficient due to the degradation of historical documents and style variation in handwriting. Word spotting provides a more efficient alternative for accessing the contents of historical manuscripts. Word spotting aims to determine the appearances of a query word in a document image. It recognizes the words as a whole rather than character by character as in OCR. Spotting words from a possibly large range of vocabulary is challenging since training discriminative model would require many positive samples per word, which in turn makes the model hard to scale to large vocabularies.

In this context we propose an out of vocabulary (OOV) word spotting method for historical manuscripts based on convolutional siamese networks. Siamese network can rank similarity between two input images. Once the network is trained to learn the image representations in a supervised manner, it can generalize to predict OOV words. Results on George Washington (GW) dataset fall behind the compared work. Results on VML historical Arabic manuscript dataset outperform the state of the art. The performance decreases with the number of OOV words but the results are still comparable with the literature.

The rest of the paper is organized as follows: Section II briefly overviews the related research, Section III explains the method, Section IV describes the datasets and presents the experimental results and Section V draws some conclusions.

II. Related Work

Recognition based retrieval can be at word level or at character level. OCR is the character level recognition and has no vocabulary limitation. Word level recognition on historical documents has been successful with very limited vocabulary. Recognition based methods are often rely on learning models by neural networks [1], [2] and Markov models [3], [4], [5].

OCR free retrieval was initially proposed by Manmatha et al. [6] and is called word spotting. It avoids recognition of words by deciding whether two given words are similar. It first segments a collection of document images into word images. Each word image is matched against all the other word images, by calculating pairwise distance among word images. Matching is the difficult aspect of word spotting because of the variations in handwriting and degradation in historical documents. Matching techniques fall into two categories: pixel by pixel matching and feature based matching [7].

Pixel by pixel matching compares two images pixel by pixel using XOR [8], [9], Sum of Squared Differences (SSD) [8] or Euclidean Distance Mapping (EDM) [8], [9]. Feature based matching extracts image features and compares two images using Scott and Longuet Higgins (SLH) [10] algorithm [8], [9], Shape Context (SC) algorithm [11], [12], Dynamic Time Wrapping (DTW) [12] or corner feature correspondence algorithm (CORR) [13]. In general feature based matching and in specific DTW performs best among these set of algorithms surveyed in [14].

In early word spotting applications, image features fall in two broad categories: Global features and local features. Global features are extracted from the whole word image and include width, height, aspect ratio, number of foreground pixels or moments of black pixels distribution. Global features alone are obsolete in the literature and often used in combination with local features [15], [16], [17]. Local features are extracted from each column of the word image and concatenated into variable length feature vectors. Initially proposed local features are upper and lower word profiles, number of foreground pixels or number of transitions from foreground to background [12], [14].

Histogram of Gradients (HOG) and Scale Invariant Feature Transform (SIFT) features have also been used for word spotting [18], [19]. These gradient based features

978-1-5386-3346-5/18 $31.00 © 2018 IEEE
DOI 10.1109/DAS.2018.67

can represent the directions of the strokes and are not constrained to single writer collections.

Recently Convolutional Neural Networks (CNN) are used as feature extractor [20], [21], [22]. A raw word image is fed through the trained CNN and the activations of last conventional layer is used as the feature vector. CNNs are able to be successful on multiple writer collections if the training set is large enough.

III. Method

Word spotting is given a query word, retrieving all its instances in a document image [14]. Search space could be either segmented word images or the whole document image. Query word representation may be extracted in two ways. Query-by-example, uses an example query word image cropped from the document image and query-by-string uses a textual input. In this work we apply word spotting on segmented word images using query-by-example.

The conventional way for comparing two images is to use descriptors and a distance metric. Commonly used descriptors are SIFT descriptors [23] and HOG descriptors [24]. However, in computer vision tasks such hand-crafted features are outperformed by learned features [25]. In this work we use convolutional siamese neural network to learn word image descriptors and compare them for word spotting.

A. Siamese Network

A siamese network consists of two branches that share the same Convolutional Neural Network (CNN) architecture and the same weights [26]. The input is a pair of images and the output is a distance in $[0, 1]$ range, between their corresponding labels. Each branch takes one of the input pairs and applies a set of convolutional, ReLU and max-pooling layers. The output of each branch is the descriptor of corresponding input image. Branches are joined by a distance metric layer. This layer computes the distance between the descriptors of input images. Siamese network finds a function that maps input images into a space in which the distance metric approximates the difference between the corresponding labels. It searches for a weights vector θ such that the distance metric is small if the input pairs are from same class and large if they are from different classes.

B. Architecture

Architecture of a CNN contains hyperparameters that can not be directly learned from the training process. Manual hyperparameter tuning can work very well when we have a good starting point applied to similar tasks [27]. Hence we based our architecture on the CNN proposed in [28] for text spotting in natural images. The texts in natural images, similar to historical document manuscripts, suffer from having orientations, noise, large number of fonts and styles. We measured the distance between the outputs of CNN branches using euclidean distance with the objective to minimize contrastive loss [29]. Let $\mathbf{x}_1, \mathbf{x}_2$ be a pair of word images. Let y be the label of this pair, where $y = 0$ if their distance is close and $y = 1$ if their distance is far. Then the network learns the parameters θ of distance function such that:

$$D_\theta(\mathbf{x}_1, \mathbf{x}_2) = \parallel h_\theta(\mathbf{x}_1) - h_\theta(\mathbf{x}_2) \parallel_2 \tag{1}$$

where h_θ is the output function of the CNN. Then the contrastive loss function is defined as:

$$L(\theta, y, \mathbf{x}_1, \mathbf{x}_2) = \frac{1}{2}[(1-y)(D_\theta)^2 + (y)[max(0, m - D_\theta)]^2] \tag{2}$$

where $m > 0$ is a margin. Far pairs contribute to the loss only if they are closer than the margin. Hadsell et al. [29] paired a sample only with its 5 nearest neighbours to create the similar pairs in a toy dataset. However we paired a sample with all other samples with the same label because data scarcity. Therefore we made a small change in the original contrastive loss such that:

$$\begin{aligned} L(\theta, y, \mathbf{x}_1, \mathbf{x}_2) = \frac{1}{2}[(1-y)[min(m, D_\theta)]^2 + \\ y[max(0, m - D_\theta)]^2] \end{aligned} \tag{3}$$

This assures that same labeled pairs can not increase the loss more than the margin. However the results with both equations were similar.

Using the combination of [28]'s architecture and [29]'s contrastive loss function we made a forward search in the following hyperparameter ranges. Input size $\in \{100 \times 100, 50 \times 110, 60 \times 120\}$, number of units in fully connected layer $\in \{512, 1024, 2048, 4096\}$, optimization algorithm $\in$ {SGD, RMSProp, AdaDelta, Adam} with learning rate $\in \{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$, regularization with dropout rate $\in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ after the fully connected layer and batchsize $\in \{16, 32, 64, 128\}$.

In the resulting siamese network, a CNN (Figure 1) in one of the branches has five convolutional layers and two fully connected layers. The input is 50×110 and 60×120 gray scale image obtained by resizing the word image and then normalizing it, in VML and GW datasets respectively. Rectified Linear Unit (ReLU) is used after each layer with weights. The convolutional layers have 64, 128, 256, 512 and 512 square filters with edge sizes of 5, 5, 3, 3 and 3 respectively. Convolutions are performed with the same padding to preserve spatial dimensions. 2×2 max pooling follows all the convolutional layers except the fourth. The fully connected layers have 4096 neurons.

The gradient of the cost function with respect to the parameters is computed using back propagation and stochastic gradient descent with a learning rate of 0.001. The siamese CNN is implemented with high level python library Keras [30] using a single NVIDIA 1080GTX.

C. Evaluation

Following the work of Rath and Manmatha [12], we used an information retrieval approach for evaluation. Each image in the dataset is used as a query to retrieve similar images from the entire dataset. This produces a ranked list of retrieved images sorted by their distance to the query image. We computed the average precision of each query and reported the mean of average precision

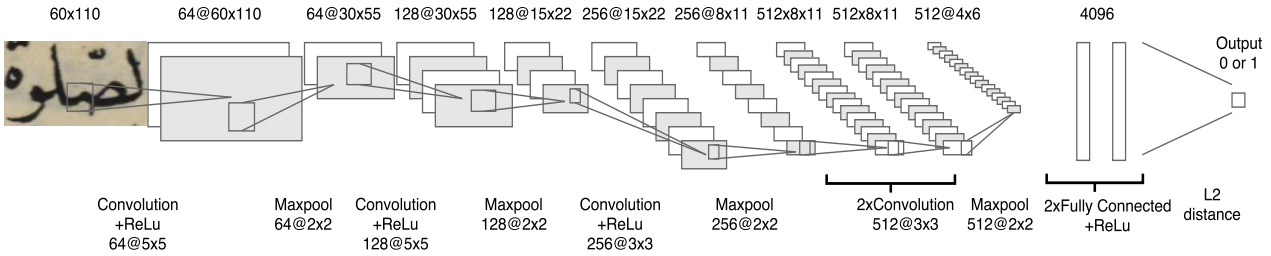

Figure 1. Architecture of the convolutional network. Siamese network is the joint of two architecture like this with L2 distance.

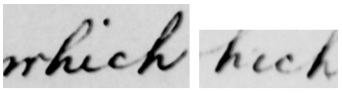

Figure 2. A segmentation error on the word "which" in GW dataset.

(mAP) of all queries. We also reported precision at the top k retrievals (P@k). We did not consider queries as candidates. Hence the words that appear once in the dataset were not used as queries. But they still exist in the dataset as distractors.

IV. Experiments

A. George Washington dataset

It is worth to note that this dataset is different from the one used in [31] and [22] which contains around 5000 words. We will refer to their dataset as GW5000 and our one as GW. GW was first used in [12] and contains images of words segmented from 10 pages of George Washington collection. The segmentation is performed automatically using the algorithm in [32], which led to to the existence of segmentation errors (Figure 2). The downloaded test set[1] contains 2381 images from which 9 images without words were removed. We select GW, because GW5000 has a predefined cross validation split [31] in word level whereas we use a class level split to ensure complete OOV.

1) Training: We split dataset into five folds in class level, where test set is completely OOV. Three folds are used for training, one fold is used for validation and the remaining fold is used for testing. In this way we avoid tuning the parameters on the test data. We repeated the experiment for all 5 different combination of train, validation and test folds and report the average test results.

We created a train set of pairs with images taken from a train set. A pair from the same class is labeled (0), and a pair from different classes is labeled (1). We included all possible positive pairs and a negative pair for each positive pair, produced by randomly pairing images of different words. Hence the train set contains equal number of 0 labeled pairs and 1 labeled pairs. However we recognized that an evaluation set with the nature of test set leads to better results (Figure 3). The difference between test set and training set is in the ratio of pair labels. Training set contains equal number of 0 labeled pairs and 1 labeled

[1]https://ciir.cs.umass.edu/download

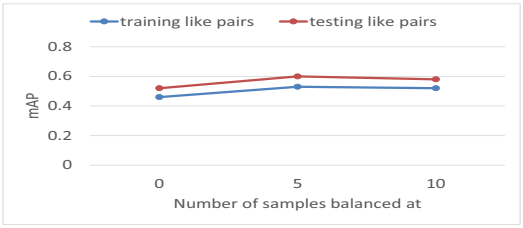

Figure 3. mAP values of first fold on the raw GW dataset and balanced datasets at 5 and 10 samples. Using validation pairs in the nature of testing pairs improves the results.

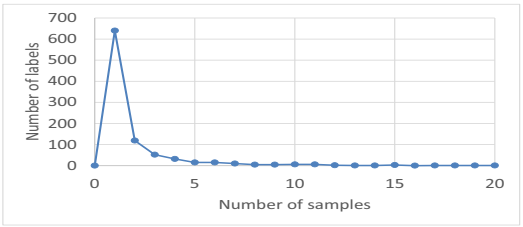

Figure 4. Number of samples in every label of GW dataset.

pairs whereas test set contains 0 labeled pairs in the number of query word and 1 labeled pairs in the number of non-query words. Consequently test set does not contain equal number of 0 labeled pairs and 1 labeled pairs.

GW dataset contains 929 classes with 2372 words. It is a highly imbalanced dataset where the classes are not nearly equally represented (Figure 4). Therefore we balanced the training set. For balancing at k samples, minority classes oversampled to k samples by copying random samples and majority classes are undersampled to k samples by deleting random samples. We showed mAP values of first fold on the raw dataset (represented with 0 in the graph) and the datasets balanced at 5 and 10 samples in Figure 3. Accordingly, training on a dataset balanced at 5 samples, using validation pairs in the nature of testing pairs leads to the best result.

2) Results: Table I shows the mAP and P@k values of the proposed method on the GW dataset. Results are compared with the work of Rodriguez and Perronin [33]. Their method is also based on 5-fold cross validation and spots the words QBE on the segmented word images. Although the GW words have a very consistent writing style, proposed method could not outperform [36]'s result. This can be because the proposed method was tested on completely OOV, whereas Rodriguez and Perronin did not

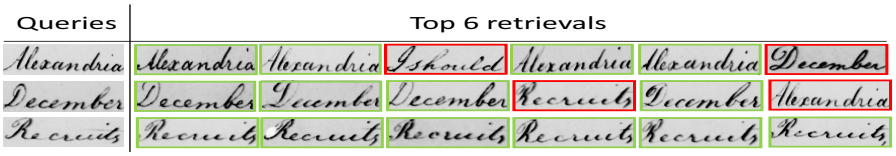

Figure 5. Sample queries and top 6 relevant retrievals to the query from the GW dataset.

Table I
MAP AND P@K VALUES ON GW DATASET. FOR THE PROPOSED METHOD ALL QUERIES ARE OOV.

Metric	Proposed	Rodriguez&Perronin[33]
mAP	0.49	**0.53**
P@1	0.71	-
P@2	0.72	-
P@3	0.73	-
P@4	0.69	-
P@5	0.67	-

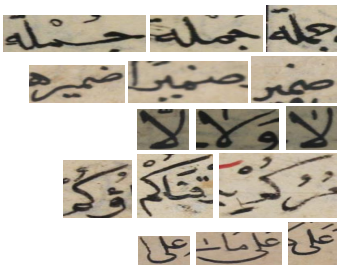

Figure 6. Each row shows the variation of a word in a book's test set.

point out that the test set is completely OOV.

Finally, Figure 5 shows some qualitative results on the GW dataset. Word images with distortions can still be retrieved successfully by the proposed method.

B. Visual Media Lab Dataset

Visual Media Lab (VML) [34] dataset is a new dataset for historical handwritten Arabic word spotting and recognition. It contains five books written by five writers in the years 1088-1451. An example document image from each book is shown in Figure 7. The books are fully annotated in word level. The downloaded dataset[2] has 680 pages in total and on average 136 pages from each book. It contains a total of $121,636$ words from 1731 classes. The variation in a word may be any combination of noise, writing style and font size (Figure 6).

1) Training: We based our experiments on the first book but evaluated the best model on all the 5 books. We split the first book dataset into training, validation and testing parts in class level. This ensures that test set is completely OOV. First, all the classes with more than 100 samples were put in test part, then all the classes with more than 20 samples were put in training part, and lastly all the classes with more than 10 samples were put in validation part. We did not use all the samples available in a part. For testing we used random 21 classes each with 100 samples for comparing the results with Kassis and El-Sana's work [35]. For validation we used random 20 classes each with 10 samples. Lastly, for training we used random 100 classes each with gradually increasing number of samples. Then the pairs were created as described in IV-A1, paragraph two. We did not remove or correct segmentation or annotation errors. Book 3, 4 and 5 contains a class of mislabels which we did not include in the experiments.

Gradual increment in train set size aims to monitor the affect of train set size on the performance. We showed mAP values of training with set sizes of $100 \times 3, 100 \times 5, 100 \times 10$ and 100×15 in Figure 8. Accordingly, best result with the least training set size is achieved with 500 samples. We made all experiments with this model.

2) Results: We trained a model on the first book and test this model on all the five books. Book 1 test set is completely OOV where book 2, 3, 4 and 5 have 7, 5, 6 and 5 classes in common with training set. We built another test set by combining the test sets of all the books. This test set contains 11 words in common with the training set. Table II shows the mAP and P@k values of the proposed method on the VML dataset. Training on the first book and testing on the same book aims to show our method's performance in spotting words with variations and classes that are unknown at training stage (Figure 6). Obviously results on VML dataset are better than on GW dataset. We think that this is because highly imbalanced nature of GW dataset. Therefore, it is better to annotate nearly equal number of samples from each class.

Training on the first book and testing on the other books aims to expose the proposed method's performance in spotting words with writing styles and backgrounds that are unknown at training stage. A model trained on a book can be used to spot words in another book, with a degraded performance.

Finally, we compare with the results of [35] (Figure 9). Their work is also segmentation and example based. Their results are on the same five books using 21 types of words each with 100 samples. The metric is hit rate at top k retrievals. It is the rate of labels that returned at least one true positive at the top k retrievals to the total number of labels. Using 2100 queries for each book and 10500 queries for the combined test set, word spotting with convolutional siamese network can outperform their results.

V. CONCLUSIONS

This paper introduces convolutional siamese network to represent and compare word images for word spotting. This method inputs a pair of word images, extracts their

[2]https://www.cs.bgu.ac.il/ vml/

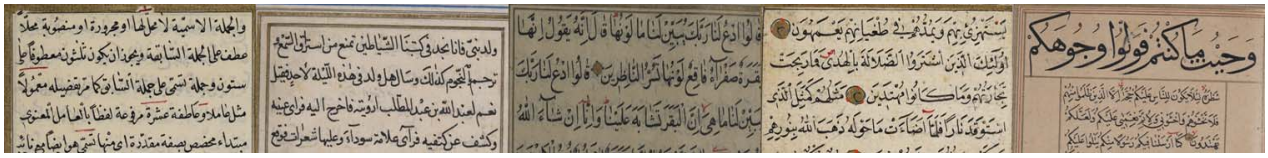

Figure 7. Variety of writing styles and backgrounds in 5 books.

Table II
MAP AND P@K VALUES ON VML DATASET. BOOK 1, 2, 3, 4 AND 5 HAVE 0, 7, 5, 6 AND 5 CLASSES IN COMMON WITH THE TRAINING SET.

Book	mAP	P@1	P@2	P@3	P@4	P@5	#Queries	#Labels	#OOV words
Book1	0.91	1.0	1.0	1.0	1.0	1.0	2100	21	21
Book2	0.75	1.0	1.0	1.0	1.0	0.99	2100	21	14
Book3	0.82	1.0	1.0	0.98	0.99	0.98	2100	21	16
Book4	0.75	1.0	1.0	0.98	0.99	0.98	2100	21	15
Book5	0.80	0.96	0.96	0.95	0.94	0.94	2100	21	16
All	0.62	0.95	0.95	0.93	0.93	0.94	10500	58	47

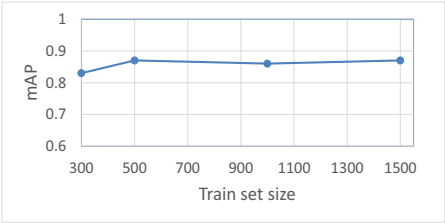

Figure 8. mAP values on the first book of VML dataset with gradually increasing train set size.

representations and outputs the distance between them. Once the model learns the filters to extract descriptors, it can then be used to spot in vocabulary or OOV words. Training on a balanced dataset is more effective than on an imbalanced dataset. We demonstrate that proposed method is robust to variations in writing styles and backgrounds.

ACKNOWLEDGMENT

The authors would like to thank Gunes Cevik for helping in Arabic dataset preparation. This research was supported by the Lynn and William Frankel Center for Computer Sciences at Ben-Gurion University.

REFERENCES

[1] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 5, pp. 855–868, 2009.

[2] R. Alaasam, B. Kurar, M. Kassis, and J. El-Sana, "Experiment study on utilizing convolutional neural networks to recognize historical arabic handwritten text," in *Arabic Script Analysis and Recognition (ASAR), 2017 1st International Workshop on*. IEEE, 2017, pp. 124–128.

[3] M. Liwicki and H. Bunke, "Combining on-line and off-line systems for handwriting recognition," in *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, vol. 1. IEEE, 2007, pp. 372–376.

[4] A.-L. Bianne-Bernard, F. Menasri, R. A.-H. Mohamad, C. Mokbel, C. Kermorvant, and L. Likforman-Sulem, "Dynamic and contextual information in hmm modeling for handwritten word recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 10, pp. 2066–2080, 2011.

[5] D. Fernández-Mota, R. Manmatha, A. Fornes, and J. Llados, "Sequential word spotting in historical handwritten documents," in *Document Analysis Systems (DAS), 2014 11th IAPR International Workshop on*. IEEE, 2014, pp. 101–105.

[6] R. Manmatha, C. Han, E. M. Riseman, and W. B. Croft, "Indexing handwriting using word matching," in *Proceedings of the first ACM international conference on Digital libraries*. ACM, 1996, pp. 151–159.

[7] R. M. T. M. Rath, "Indexing of handwritten historical documents-recent progress," in *Proceedings 2003 Symposium on Document Image Understanding Technology*. UMD, 2003, p. 77.

[8] R. Manmatha and W. Croft, "Word spotting: Indexing handwritten archives," *Intelligent Multimedia Information Retrieval Collection*, pp. 43–64, 1997.

[9] S. Kane, A. Lehman, and E. Partridge, "Indexing george washingtons handwritten manuscripts," *Center for Intelligent Information Retrieval, Computer Science Department, University of Massachusetts, Amherst, MA*, vol. 1003, 2001.

[10] G. L. Scott and H. C. Longuet-Higgins, "An algorithm for associating the features of two images," *Proceedings of the Royal Society of London B: Biological Sciences*, vol. 244, no. 1309, pp. 21–26, 1991.

[11] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE transactions on pattern analysis and machine intelligence*, vol. 24, no. 4, pp. 509–522, 2002.

[12] T. M. Rath and R. Manmatha, "Word image matching using dynamic time warping," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, vol. 2. IEEE, 2003, pp. II–II.

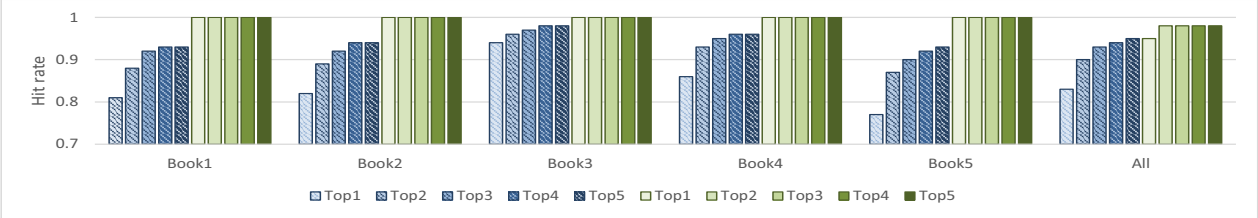

Figure 9. Comparison of proposed method with Kassis and El-Sana's work [35]. Pattern filled bars show their results, solid filled bars show our results.

[13] J. L. Rothfeder, S. Feng, and T. M. Rath, "Using corner feature correspondences to rank word images by similarity," in *Computer Vision and Pattern Recognition Workshop, 2003. CVPRW'03. Conference on*, vol. 3. IEEE, 2003, pp. 30–30.

[14] T. M. Rath and R. Manmatha, "Word spotting for historical documents," *International Journal on Document Analysis and Recognition*, vol. 9, no. 2, pp. 139–152, 2007.

[15] A. Fischer, A. Keller, V. Frinken, and H. Bunke, "Hmm-based word spotting in handwritten documents using sub-word models," in *Pattern recognition (icpr), 2010 20th international conference on*. IEEE, 2010, pp. 3416–3419.

[16] V. Frinken, A. Fischer, R. Manmatha, and H. Bunke, "A novel word spotting method based on recurrent neural networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 2, pp. 211–224, 2012.

[17] A. Fischer, V. Frinken, H. Bunke, and C. Y. Suen, "Improving hmm-based keyword spotting with character language models," in *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*. IEEE, 2013, pp. 506–510.

[18] J. A. Rodriguez and F. Perronnin, "Local gradient histogram features for word spotting in unconstrained handwritten documents," *Proc. 1st ICFHR*, pp. 7–12, 2008.

[19] K. Terasawa and Y. Tanaka, "Slit style hog feature for document image word spotting," in *Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on*. IEEE, 2009, pp. 116–120.

[20] G. Sfikas, G. Retsinas, and B. Gatos, "Zoning aggregated hypercolumns for keyword spotting," in *Frontiers in Handwriting Recognition (ICFHR), 2016 15th International Conference on*. IEEE, 2016, pp. 283–288.

[21] P. Krishnan, K. Dutta, and C. Jawahar, "Deep feature embedding for accurate recognition and retrieval of handwritten text," in *Frontiers in Handwriting Recognition (ICFHR), 2016 15th International Conference on*. IEEE, 2016, pp. 289–294.

[22] S. Sudholt and G. A. Fink, "Phocnet: A deep convolutional neural network for word spotting in handwritten documents," in *Frontiers in Handwriting Recognition (ICFHR), 2016 15th International Conference on*. IEEE, 2016, pp. 277–282.

[23] J. Almazán, A. Gordo, A. Fornés, and E. Valveny, "Word spotting and recognition with embedded attributes," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 12, pp. 2552–2566, 2014.

[24] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.

[25] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *NIPS workshop on deep learning and unsupervised feature learning*, no. 2, 2011, p. 5.

[26] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a" siamese" time delay neural network," in *Advances in Neural Information Processing Systems*, 1994, pp. 737–744.

[27] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[28] 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.

[29] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, vol. 2. IEEE, 2006, pp. 1735–1742.

[30] F. Chollet *et al.*, "Keras," https://github.com/fchollet/keras, 2015.

[31] J. Almazn, A. Gordo, A. Forns, and E. Valveny, "Efficient exemplar word spotting." in *BMVC*, R. Bowden, J. P. Collomosse, and K. Mikolajczyk, Eds. BMVA Press, 2012, pp. 1–11.

[32] R. Manmatha and N. Srimal, "Scale space technique for word segmentation in handwritten documents," *Lecture notes in computer science*, pp. 22–33, 1999.

[33] J. A. Rodríguez-Serrano and F. Perronnin, "A model-based sequence similarity with application to handwritten word spotting," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 11, pp. 2108–2120, 2012.

[34] M. Kassis, A. Abdalhaleem, A. Droby, R. Alaasam, and J. El-Sana, "Vml-hd: The historical arabic documents dataset for recognition systems," in *1st International Workshop on Arabic and derived Script Analysis and Recognition*. IEEE, 2017.

[35] M. Kassis and J. El-Sana, "Word spotting using radial descriptor graph," in *Frontiers in Handwriting Recognition (ICFHR), 2016 15th International Conference on*. IEEE, 2016, pp. 31–35.