Query-by-Online Word Spotting Revisited: Using CNNs for Cross-Domain Retrieval

Sebastian Sudholt, Leonard Rothacker, Gernot A. Fink
Department of Computer Science
Technische Universität Dortmund University
44221 Dortmund, Germany
Email: {sebastian.sudholt, leonard.rothacker, gernot.fink}@tu-dortmund.de

***Abstract*—A word spotting system is in large parts characterized by the query modalities it is able to process. The most common modalities here are Query-by-Example and Query-by-String. However, recently a new query type has been proposed: In Query-by-Online-Trajectory (QbO) the query is presented as a set of online-handwritten trajectories. In this work we devise a cross-domain word spotting framework using CNNs which is able to accomplish the QbO task. In particular, we design two different QbO systems which we evaluate in a number of experiments. We are not only able to outperform the current state of the art in QbO word spotting but also show that a system using a single CNN for both online and offline data achieves superior results compared to a system that uses a CNN for each domain individually.**

I. Introduction

Word spotting has garnered a large amount of research attraction over the recent past. The goal in word spotting is to retrieve parts of a document collection which are relevant with respect to a certain query. Typically, the sought after elements are word or line images.

A current trend in word spotting is to create a common, holistic representation of queries and test documents. Very influential work in this regard was presented in [1] in the from of *Pyramidal Histograms of Characters (PHOC)* which is an embedded attribute representation for text strings. Learning this representation from document images can successfully be achieved with either an ensemble of SVMs [1], [2] or *Convolutional Neural Networks (CNN)* [3], [4]. Performing *Query-by-Example (QbE)* word spotting (query is a document image) boils down to a simple nearest neighbor search within the predicted representations. *Query-by-String (QbS)* (query is a text string) can be as easily achieved as the query string can directly be mapped into the PHOC space. In general, a wide variety of query modalities is possible under this framework as long as the query can be mapped to a PHOC representation.

This work will focus on *Query-by-Online-Trajectory (QbO)*. QbO is an emerging paradigm in the field of word spotting which enables a natural interface for running word spotting applications on either touchscreen-equipped devices or smart boards [5]. For these devices, the human machine interaction is vastly improved by QbO as a user can intuitively define the desired query without the need for additional input devices such as keyboards. In the context of this work, online-handwritten trajectories are sequences of points (pen positions) contrasting offline document images which are pixel representations of scanned documents.

If a query to a word spotting system can be represented as an image, CNNs achieve superior results to other approaches [2]–[4]. It is desirable to transfer this success to sequential online trajectories as well. However, it is not obvious how to feed online trajectories into a CNN. While recurrent neural networks such as *Long Short Term Memories (LSTM)* are capable of processing online-handwritten data [6], it is unclear how an LSTM can be incorporated into the embedded attribute framework or how it can generate a PHOC representation. Moreover, processing online trajectories with LSTMs requires pre-processing techniques which are largely based on heuristics [6], [7]. Some of the pre-processing could be remedied by Convolutional-LSTMs [8] but the question of how to generate a PHOC representation from these networks still prevails.

A CNN is inherently designed to process offline images. Thus a simple solution for a CNN to process online trajectories is to render them into offline images. These images can then be feed to the CNN as input. This approach was already used in [9] for allowing an offline HMM-based classifier to be trained and tested on online trajectories.

Our contribution in this work is to devise a unified framework for word spotting with online-handwritten trajectories based on CNNs. Rendered online trajectory images are used for training CNNs and generating an embedded attribute representation from a given query trajectory. We evaluate both, a system with two separate CNNs for rendered online trajectories and offline word images and one with a single CNN for data from both domains. We show empirically that results for cross-domain word spotting can be vastly improved by using a single CNN for predicting the PHOC representation for both rendered online trajectories and offline document images.

II. Related Work

One of the earliest works on word spotting in handwritten documents performs word spotting through the use of XOR-maps and Euclidean Distance Mapping [10]. Ensuing works made heavy use of techniques that had proven effective for handwriting recognition tasks. In [11] the authors use a Dynamic Time Warping approach for comparing contour features of different word images. The approaches in [12] and [13] both

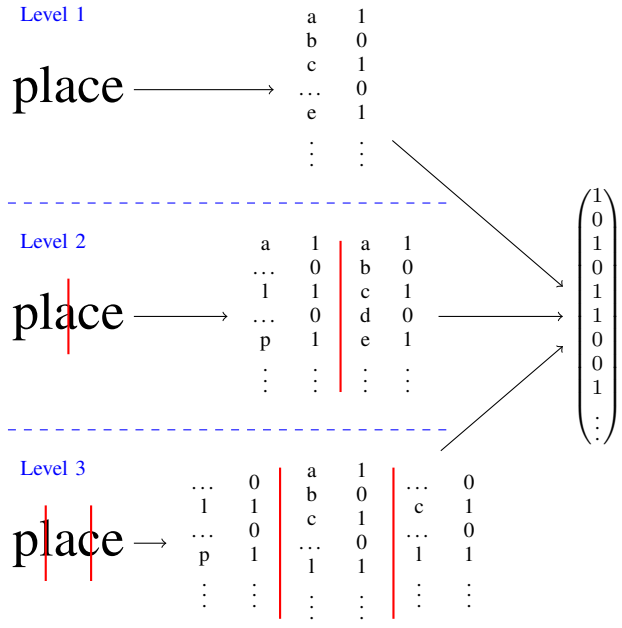

Fig. 1. The figure exemplarily visualizes the creation of a three-level PHOC from a word string.

use Hidden Markov Models (HMM). In [14] a Bidirectional Long Short Term Memory is used to spot query words in segmented line images.

Alongside the sequential models, holistic models have been successfully used for word spotting as well. In [15] the authors make use of a Spatial Pyramid representation on top of a Bag-of-Features (BoF) representation of quantized SIFT-features in order to perform segmentation-free word spotting. This concept is extended in [16] by incorporating product quantization for efficient retrieval. Spatial Pyramids and SIFT-features are used in [17] as a visual representation as well. Using Latent Semantic Analysis, the visual representation together with a textual representation of the sought after query are projected into a common subspace in which QbS word spotting is possible. Combining the BoF principle and sequential models, Bag-of-Feature HMMs are used for segmentation-free word spotting in [18]. The approach of BoF-HMMs is further extended to segmentation-free QbS in [19].

One of the most recent trends in word spotting has been learning an embedded attribute representation for a given word image. Very influential work in this regard was presented in [1] in the form of *Pyramidal Histograms of Characters (PHOC)*. A PHOC is a binary histogram representing presence or absence of certain characters in a string. A three-level PHOC is visualized in figure 1. Except for the first layer, a binary histogram is created for certain splits of the word string. E.g., the word string is split into halves at the second level and a binary histogram is created for each split representing presence or absence of characters in the specific split. Finally, all histograms from all levels are concatenated to form the final PHOC. One of the PHOC's main traits is that it can directly be generated from a string. If a model is able to predict a PHOC representation for a given word image, this prediction can directly be used for QbE and QbS. Moreover, new query modalities can be plugged into the PHOC framework if they can be mapped to this attribute embedding. This is exactly the approach in [5] where a PHOC representation is learned from online-handwritten trajectories through the AttributeSVM framework.

Due to the current success of Deep Learning in other fields of computer vision, recent works in word spotting have focused on learning representations through neural networks as well. In [2] a CNN is used as feature extractor. These features are then used as input to the AttributeSVM framework, effectively replacing the Fisher Vector in [1]. Another recent approach makes use of Triplet-CNNs as feature extractors [4]. The generated features are forwarded to a 3-layer MLP which is trained by applying the Cosine Embedding Loss. This way, the system is not only able to predict binary attribute representations like the PHOC, but also real valued ones. The authors compare the results achieved with a PHOC compared to those using a novel holistic word image representation called *Discrete Cosine Transform of Words*. The evaluation suggest that both representations achieve quite similar results.

The current state of the art for a wide range of segmentation-based word spotting benchmarks is set by the *PHOCNet* [3]. This CNN is able to learn the PHOC in an end-to-end fashion through a non-linear from of logistic regression: The output of each neuron in the last layer is forwarded through a sigmoid activation functions which squashes the output to the range $(0, 1)$. As each neuron is processed individually, the network is able to predict the n out of k encoding of the PHOC.

III. QbO with the PHOCNet

The goal in Query-by-Online word spotting is to retrieve a list of offline word images from a given document image collection with respect to an online-handwritten trajectory. Here, an online-handwritten trajectory is assumed to be a sequence of trajectory points and the document image collection a set of images. Note that QbO differs from other retrieval scenarios such as [20] where online-trajectories are retrieved from textual representations such as ASCII. Informally speaking, we want to retrieve offline images from online data. In the following, we will refer to the word images to be retrieved simply as test images.

In order to solve the problem at hand, we draw inspiration from [5] and project online and offline data into a common attribute space. For this, we first render the online trajectories into offline images which can then be fed to a CNN. This approach allows us to neglect pre-processing techniques such as skew and slant correction or removal of delayed strokes [6], [7]. Much rather than applying these heuristics, we let the CNN learn an appropriate normalization of the unwanted variability during training.

In order to predict the PHOC representation from the rendered online images, we employ the recently proposed PHOCNet [3]. The PHOCNet is a 19-layer CNN which can be

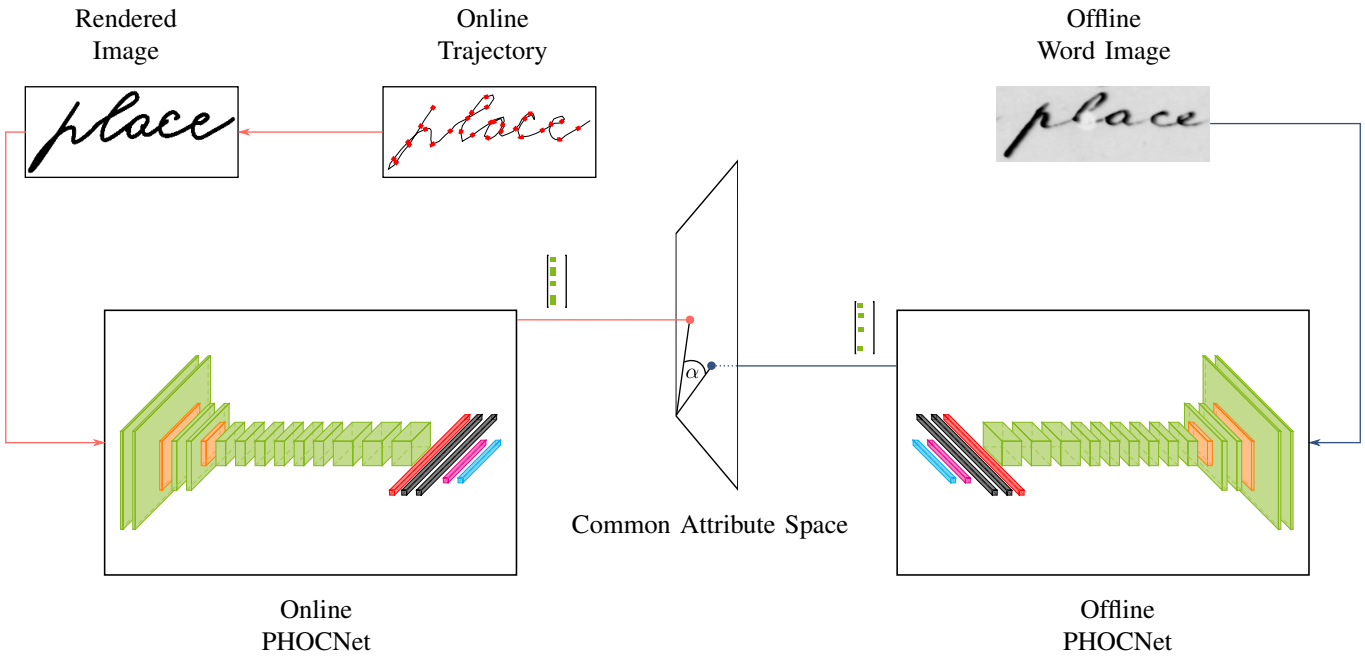

Fig. 2. Outline of 2-PHOCNet System: The online trajectory is first rendered and then passed through the PHOCNet trained on rendered online images. The offline document image is passed through a separate PHOCNet which was previously trained on offline data. The predicted PHOC representations can then be exploited for word spotting by ranking them with a suitable distance metric.

trained with a set of document images and their corresponding PHOC annotation in an end-to-end fashion. We chose this CNN over other architectures as it is able to achieve state of the art results on even small data sets without the need for extensive additional amounts of data as are needed for e.g. Deep Feature Embedding [2] or Triplet-CNNs [4].

Using the PHOCNet for the rendered online images and the offline document images, we devise two different word spotting systems.

For the first system, one PHOCNet is trained on the rendered online trajectory images (ROTI) and another PHOCNet is trained to predicting PHOCs from offline word images. In the following, we refer to this system as *2-PHOCNet System*. Figure 2 gives an overview over the first system. It is important to note here, that the two different PHOCNets do not share any parameters and are only trained on their respective data (either ROTI or word images). At query time, we render the query trajectory and predict the corresponding PHOC representation from the PHOCNet trained only on the ROTIs. Likewise, we predict PHOC representations for the offline word images to be retrieved through the CNN trained on offline data. Word spotting is then performed as usual in the PHOC framework: compute the Cosine distance between predicted query PHOC and predict test PHOCs and rank the test set according to these distances.

The second system makes use of a single PHOCNet only which is trained on both ROTIS and offline word images. We refer to this as the *1-PHOCNet System*. Figure 3 visualizes this approach. Here, the single PHOCNet is able to predict PHOCs for both ROTIs and offline word images in a joint fashion. The rest of the pipeline is very similar to the first approach: Predict PHOCs for the rendered query image and offline test images and run a nearest neighbor search.

IV. Experiments

We evaluate the presented systems in a number of different experiments. In this section, we first introduce the data sets used for the experiments. Then we present the protocols used for assessing the performance of the two system and finally present the results and an accompanying discussion.

A. Data Sets

The **George Washington (GW)** data set is a collections of letters and correspondences written by George Washington and his associates in the 18th century. The data set is made up of 20 scanned pages with the ground truth containing annotation and word-level segmentation for 4860 words. The ground truth does not contain an official partitioning into training and test set. However, it is common to use a cross validation approach with the exact same cross validation partitions as were chosen in [1][1] (e.g. [2], [4]). In our experiments, we follow this cross validation approach as well.

The **George Washington Online (GWO)** is a data set of online trajectories. It was created for the experiments in [5] in order to be able to perform QbO word spotting on the GW data set. The GWO data set contains the online-handwritten trajectories for every word in the GW data set. All trajectories were created by a single writer.

[1]partitions available at https://github.com/almazan/watts/tree/master/data

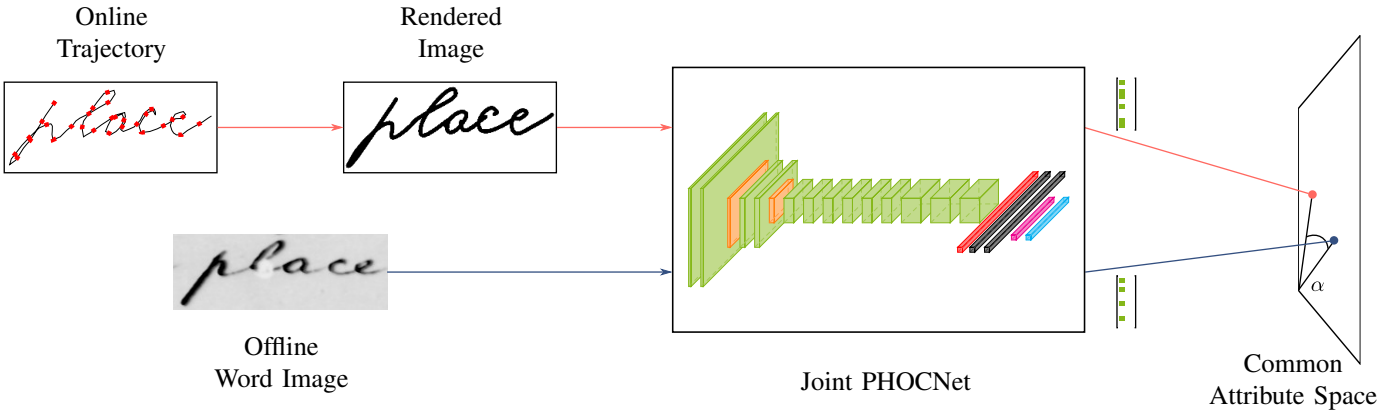

Fig. 3. Outline of the 1-PHOCNet System: The online trajectory is first rendered and then passed through a PHOCNet which has been jointly trained on both rendered online images and offline document images. Word spotting is then performed as in the first approach.

The **Unipen** data set is a collection of online-handwritten trajectories from the Unipen foundation [21]. The ground truth comes with annotations and segmentations at line, word and character level. In order to be able to directly compare our evaluation to the one reported in [5], we use the same subset of trajectories (*sta0*) for our experiments. The subset contains on average 400 trajectories for each of the 62 writers. The total amount of word-trajectories is 27 112.

The **IAM Handwritten Database (IAM-DB)** [22] consists of 115 320 words written by 657 writers. For our experiments, the official partition available for writer independent text line recognition is used for generating the training and test splits. These splits yield a training and test set of 60 453 and 13 752 words respectively. Please note that a single writer does only contribute to either the training or the test split in this official partitioning.

The last data set used is the **IAM On-Line Handwriting Database (IAM-OnDB)** [23]. It is made up of 13 049 trajectories of online-handwritten text lines, contributed by 221 writers. The IAM-OnDB does only come with line-level segmentations for the trajectories. As our PHOCNets need a word-level segmentation, we created it through a forced alignment by an HMM. The word-level segmentation will be made publicly available. The exact specifications for running the forced alignment are presented in the next section. As test partition, we use the official *testset_f* that comes with the annotation. All other word images are used for the training partition. Similar to the IAM, a single writer does only contribute to either the training or the test split in this setup.

B. Word-Level Segmentation of IAM-OnDB

In order for the PHOCNets to be trained on rendered online trajectories, a word-level segmentation of the training images is required. This word-level segmentation already exists for the GWO but is missing for the IAM-OnDB. Thus, we create such a segmentation by means of forced alignment with a semi-continuous BoF-HMM using an Exponential Dirichlet Compound Multinomial (EDCM) output model [24].

BoF can automatically be adapted to the application domain and have shown excellent performance for word spotting and handwriting recognition [24]. In this regard, BoF-based representations have also shown high robustness if no preprocessing is applied for normalizing word images, cf. [1].

In order to integrating BoF with HMMs, the generation of sequences of BoFs is modeled. BoF are column-wise extracted from a dense grid of quantized SIFT descriptors (*visual words*). The probability for generating a BoF in a specific HMM state can then be modeled with a multinomial distribution. The EDCM is an extension of a multinomial model that is suitable when the observations are very sparse. Within a line image, columns in the dense grid contain only few visual words. Due to our large visual vocabulary, sparse BoF are our standard scenario.

For computing the forced alignment we closely follow the configuration as reported in [24]. Rectangular SIFT descriptors containing 4×2 cells (*rows* $\times$ *columns*) at a cell size of 13×13 pixels are extracted in a dense grid of 5×5 pixels. After quantizing SIFT descriptors with respect to 4096 visual words we model BoF with an EDCM mixture model consisting of 1024 mixture components. Finally, we estimate 83 character models with a Bakis topology and 6 states per model.

C. Evaluation Protocol

In this section we are going to outline the protocol used for the word spotting experiments. In order to simplify explanations, we only point out the training set, query set and test set used in each experiment. For the 2-PHOCNet system, the ROTIs and offline word images of the respective training set are used to train two different PHOCNets (cf. figure 2) while for the 1-PHOCNet system both sets are combined to train one PHOCNet (cf. figure 3). The query set is always a set of online-handwritten trajectories for which PHOCs are computed by first rendering them and then forwarding them through the PHOCNet trained on the ROTIs or the joint PHOCNet. The test set is always a list of offline word images for which PHOCs are predicted through the PHOCNet trained

on offline images or the joint PHOCNet. For each experiment, word spotting is performed following the protocol used in [1] (this protocol was used in [5] as well): Each query PHOC is used to rank the test PHOCs through the Cosine distance. A query is only considered if it has at least one relevant occurrence in the test set. For each query and ranked retrieval list, the Average Precision is calculated. Finally, the overall performance of the system is assessed by computing the *mean Average Precision (mAP)*.

In our experiments, we evaluate three different QbO word spotting scenarios. The training, query and test sets for respective experiments are generated as follows:

Exp. 1: Single Known Writer (SKW): For the first experiment, the queries originate from a single writer, who already contributed other trajectories for training the word spotting system. The data sets used are the GWO and GW. First, we render all online trajectories of the GWO with a stroke width of 10 pixels. The stroke width was determined by taking five trajectories from the training partition and creating a visually appealing rendered word image. The GWO and GW are then divided into four cross validation splits. For both GWO and GW, we use the splits defined in [1].

The training set for this experiment is then the combination of both training splits for GWO and GW. The query set is the test split of the GWO while the test set is the test split of the GW.

Exp. 2: Single Unknown Writer (SUW): For the second experiment, the queries originate from a single writer, who did not contribute other trajectories for training the word spotting system. The data sets used are the Unipen, GWO and GW. First, the GWO is rendered with a stroke width of 10 pixels while for the Unipen we chose a stroke width of 20 pixels. The stroke widths were determined as in the first experiment. Afterwards, GWO and GW are again split up into the four cross validation batches.

For the training set, we use the entire Unipen data set as well as the GW training split. The query set is again the test split of the GWO data set. During pre-experiments it became evident that simply using the query set this way does not allow the CNN to reliably predict PHOCs. This is due to the vastly different scales of the Unipen and GWO images. In order to cope with this size mismatch, a scaling factor is determined which is applied to the GWO images. For this, we first compute all common words from the Unipen and the respective training partition of the GWO. Then the average height of these common words is calculated for both data sets. The scaling factor is then the quotient between the average height of the Unipen and the average height of the GWO. Finally, the test set is again the test split of the GW data set.

Exp. 3: Multiple Unknown Writer (MUW): For the third experiment, the the queries originate from multiple writers, who did not contribute other trajectories for training the word spotting system. For this, we use the IAM and IAM-OnDB data sets. The IAM-OnDB is rendered with a stroke width of 20 pixels (stroke width was determined as for Exp. 1 and 2). The training set for this experiment is the union of the two training splits of IAM and IAM-OnDB. The query set is the official test split *f* of the IAM-OnDB. Please note that we use all words from the *f* split as queries. All other trajectories in the IAM-OnDB are accounted to the training set. Finally, the test set is the test split of the IAM.

TABLE I
SUMMARY OF THE DIFFERENT SETS FOR THE THREE QBO EXPERIMENTS

Experiment	Training Set	Query Set	Test Set
Exp. 1: SKW	GWO Train + GW Train	GWO Test	GW Test
Exp. 2: SUW	Unipen Train + GW Train	GWO Test	GW Test
Exp. 3: MUW	IAM Train + IAM-OnDB Train	IAM-OnDB Test	IAM Test

Table I summarizes the training, query and test sets for the three experiments. Note that exp. 1 (SKW) and 2 (SUW) are following the same protocol as was presented in [5].

D. Experimental Setup Details

For rendering online-handwritten trajectories, we let a sphere of the diameter explained in the previous section slide along the trajectory. Each image is rendered as a binary image with no post-processing whatsoever.

All PHOCNets are trained with the exact same parameters as had been chosen in [3]: The CNNs are trained with *stochastic gradient descent (SGD)*. The learning rate is set to 10^{-4}, momentum to 0.9 and weight decay to $5 \cdot 10^{-5}$. After $70\,000$ iterations the learning rate is set to 10^{-5}. An iteration here means computing the forward and backward pass through the CNN for a single batch and updating the weights with respect to the gradient. The batch size is set to 10. The training is run for a maximum of $80\,000$ iterations.

Each layer of the PHOCNet is initialized with the weights randomly drawn from a zero-mean Gaussian distribution with variance $\frac{2}{n_l}$ where n_l is the number of inputs in layer l [25]. For example, if a convolution layer with a kernel size of 3×3 is presented with a feature map of 512 channels, n_l computes to $3^2 \cdot 512 = 4608$. The layer biases are initialized with 0.

We use the Caffe framework [26] in order to train our PHOCNets. Please note that Caffe by default scales the gradients calculated for training with the computed loss. This has to be taken into consideration when recreating the experiments.

E. Results & Discussion

The results for the three experiments and two word spotting systems are reported in table II. As can be seen in the table, the approach of using CNNs instead of AttributeSVMs leads to a considerable performance gain.

Another interesting observation is that the 1-PHOCNet System is able to consistently outperform the 2-PHOCNet System across all experiments. This result might seem counter intuitive at first as using individual CNNs for the rendered

TABLE II
RESULTS FOR THE THREE EXPERIMENTS IN MAP [%]

Method	SKW	SUW	MUW
2-PHOCNet System	96.91	83.04	55.57
1-PHOCNet System	**97.35**	**90.96**	**77.73**
AttributeSVM [5]	86.49	21.71	–

online images and offline document images should enable each CNN to focus better on the intricacies of the respective data.

However, as the CNNs essentially only see offline images, combining both data sets lets the CNN experience more intraclass variability during training. This enables it to learn a much more robust representation compared to seeing data from one domain only. What is really interesting about this is that the online trajectories are rendered as binary images while the offline word images are gray-scale images. Yet, the CNN is able to draw knowledge from combining both binary and gray-scale images. This further demonstrates the excellent generalization capability of the PHOCNet.

V. CONCLUSION

In this work, we present a unified framework for running cross-domain word spotting. Using online trajectories of handwriting, we are able to retrieve offline word images from historic as well as contemporary document images. We evaluate two different systems for this Query-by-Online word spotting approach. Using the recently proposed PHOCNet, both systems project offline document images or online-handwritten trajectories into a PHOC space in which the QbO retrieval can be solved through a simple nearest neighbor approach.

In a number of experiments, we showed that both systems are able to outperform the current state of the art in QbO. Additionally we found that the system using a single PHOCNet for both rendered online images and offline document images consistently performed better than the system using an individual PHOCNet for the two domains. This observation holds true for different amounts of known and unknown writers as well as historic and contemporary data sets.

As part of our experiments, we created a new word-level segmentation for the IAM-OnDB. This segmentation will be made available to the research community.

REFERENCES

[1] J. Almazán, A. Gordo, A. Fornés, and E. Valveny, "Word Spotting and Recognition with Embedded Attributes," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 12, pp. 2552–2566, 2014.

[2] P. Krishnan, K. Dutta, and C. Jawahar, "Deep Feature Embedding for Accurate Recognition and Retrieval of Handwritten Text," in *International Conference on Frontiers in Handwriting Recognition*, 2016, pp. 289–294.

[3] S. Sudholt and G. A. Fink, "PHOCNet : A Deep Convolutional Neural Network for Word Spotting in Handwritten Documents," in *International Conference on Frontiers in Handwriting Recognition*, 2016.

[4] T. Wilkinson and A. Brun, "Semantic and Verbatim Word Spotting using Deep Neural Networks," in *International Conference on Frontiers in Handwriting Recognition*, 2016, pp. 307–312.

[5] C. Wieprecht, L. Rothacker, and G. A. Fink, "Word Spotting in Historical Document Collections with Online-Handwritten Queries," in *International Workshop on Document Analysis Systems*, 2016.

[6] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, "A Novel Connectionist System for Unconstrained Handwriting Recognition," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 5, pp. 855–868, 2009.

[7] S. Jaeger, S. Manke, J. Reichert, and A. Waibel, "Online Handwriting Recognition: The NPen++ recognizer," *International Journal on Document Analysis and Recognition*, vol. 3, no. 3, pp. 169–180, 2001.

[8] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-k. Wong, and W.-c. Woo, "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting," *Neural Information Processing Systems*, pp. 802–810, 2015.

[9] T. Plötz, C. Thurau, and G. A. Fink, "Camera-based Whiteboard Reading: New Approaches to a Challenging Task," in *International Conference on Frontiers in Handwriting Recognition*, 2008, pp. 385–390.

[10] R. Manmatha, C. Han, and E. Riseman, "Word Spotting: A New Approach to Indexing Handwriting," *Computer Vision and Pattern Recognition*, pp. 1–29, 1996.

[11] T. M. Rath and R. Manmatha, "Word Spotting for Historical Documents," *International Journal on Document Analysis and Recognition*, vol. 9, pp. 139–152, 2007.

[12] J. A. Rodríguez-Serrano and F. Perronnin, "Handwritten word-spotting using hidden Markov models and universal vocabularies," *Pattern Recognition*, vol. 42, no. 9, pp. 2106–2116, 2009.

[13] A. Fischer, A. Keller, V. Frinken, and H. Bunke, "HMM-Based Word Spotting in Handwritten Documents Using Subword Models," in *International Conference on Pattern Recognition*, 2010, pp. 3416–3419.

[14] 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, pp. 211–224, 2012.

[15] M. Rusiñol, D. Aldavert, R. Toledo, and J. Lladós, "Browsing Heterogeneous Document Collections by a Segmentation-Free Word Spotting Method," in *International Conference on Document Analysis and Recognition*, 2011, pp. 63–67.

[16] ——, "Efficient segmentation-free keyword spotting in historical document collections," *Pattern Recognition*, vol. 48, no. 2, pp. 545–555, 2015.

[17] D. Aldavert, M. Rusinol, R. Toledo, and J. Llados, "Integrating Visual and Textual Cues for Query-by-String Word Spotting," in *International Conference on Document Analysis and Recognition*, 2013, pp. 511–515.

[18] L. Rothacker, M. Rusinol, and G. A. Fink, "Bag-of-Features HMMs for Segmentation-Free Word Spotting in Handwritten Documents," in *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, 2013, pp. 1305–1309.

[19] L. Rothacker and G. A. Fink, "Segmentation-free Query-by-String Word Spotting with Bag-of-Features HMMs," in *International Conference on Document Analysis and Recognition*, 2015, pp. 661–665.

[20] C. V. Jawahar, A. Balasubramanian, M. Meshesha, and A. M. Namboodiri, "Retrieval of Online Handwriting by Synthesis and Matching," *Pattern Recognition*, vol. 42, no. 7, pp. 1445–1457, 2009.

[21] I. Guyon and L. Schomaker, "UNIPEN project of on-line data exchange and recognizer benchmarks," in *International Conference on Pattern Recognition*, vol. 2, 1994, pp. 29–33.

[22] U. V. Marti and H. Bunke, "The IAM-database: An English Sentence Database for Offline Handwriting Recognition," *International Journal on Document Analysis and Recognition*, vol. 5, no. 1, pp. 39–46, 2002.

[23] M. Liwicki and H. Bunke, "IAM-OnDB - An on-line English Sentence Database Acquired from Handwritten Text on a Whiteboard," in *International Conference on Document Analysis and Recognition*, 2005, pp. 956–961.

[24] L. Rothacker and G. A. Fink, "Robust Output Modeling in Bag-of-Features HMMs for Handwriting Recognition," in *International Conference on Frontiers in Handwriting Recognition*, 2016, pp. 199–204.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," in *International Conference on Computer Vision*, 2015, pp. 1026–1034.

[26] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, and U. C. B. Eecs, "Caffe : Convolutional Architecture for Fast Feature Embedding," in *ACM Conference on Multimedia*, 2014, pp. 675–678.