Word Spotting and Recognition using Deep Embedding

Praveen Krishnan, Kartik Dutta and C. V. Jawahar
Center for Visual Information Technology, IIIT Hyderabad, India.
{*praveen.krishnan, kartik.dutta*}*@research.iiit.ac.in and jawahar@iiit.ac.in*

***Abstract*—Deep convolutional features for word images and textual embedding schemes have shown great success in word spotting. In this work, we follow these motivations to propose an End2End embedding framework which jointly learns both the text and image embeddings using state of the art deep convolutional architectures. The three major contributions of this work are: (i) an End2End embedding scheme to learn a common representation for word images and its labels, (ii) building a state of art word image descriptor and demonstrating its utility as off-the-shelf features for word spotting, and (iii) use of synthetic data as a complementary modality to further enhance word spotting and recognition. On the challenging IAM handwritten dataset, we report a mAP of 0.9509 for query-by-string based retrieval task. Under lexicon based word recognition, our proposed method report a 2.66 and 5.10 CER and WER respectively.**

***Keywords*-Handwritten Word Images, Word Spotting, Word Recognition, Deep Features.**

I. Introduction

Learning efficient representation for word images is one of the key problems to address for successful retrieval and recognition of handwritten documents. Unlike printed documents where variations are limited, handwritten images contain a diverse set of writings which makes the problem challenging. More recently, there has been a significant progress in this field with the adoption of newer deep learning frameworks such as [1]–[8] for addressing the challenges associated with handwritten images. In terms of performance, these methods report a significant error reduction of more than 50% as compared to previous non-deep methods for the task of word spotting on popular datasets such as IAM [9] and George Washington (GW) pages [10]. Under recognition, similar trends have been observed where the methods [5]–[7] report a significantly low word error rates of around 10% on IAM dataset.

In the domain of word spotting, the concept of word attributes [11], [12] using the pyramidal histogram of characters (PHOC) has been a key contribution to the community which enabled label embedding techniques to learn robust and discriminative features. PHOC attributes are calculated by dividing the word into multiple pyramid levels and at each level, the histogram of characters and bi-grams are computed and concatenated for the final representation. Here each attribute or the dimension denotes the presence/absence of a character at a particular spatial position. The concept of learning word attributes has been recently demonstrated successfully using convolutional neural networks [4], [13], [14] which validates the generality of this representation. In this work, we further improve the attribute-based representation using robust image level features extracted from the HWNet v2 [15] architecture.

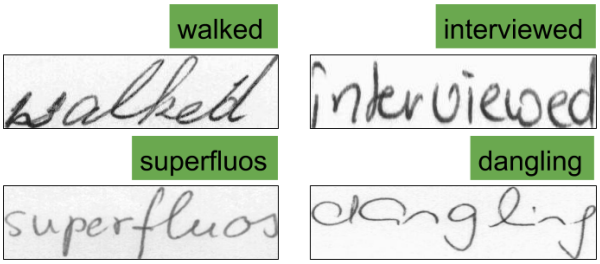

Figure 1. Few sample qualitative results for word recognition using the proposed method along with its ground truth.

In the domain of word recognition, convolutional recurrent neural networks (CRNN) [16] is the most popular architecture used in many methods. The basic idea is to formulate a sequence-to-sequence transcription task where the input sequence of features is calculated using a deep CNN architecture and is further given to a recurrent network such as LSTM [17] or MDLSTM [7], [18].

One of the greatest advantages while training deep architectures for learning word image representation is the availability of huge synthetic data such as IIIT-HWS [19] with practically no additional cost. Although there exists a significant domain gap between synthetic and real-world images, the rendered images still play an important role in pre-training huge CNN networks with millions of parameters. In addition to pre-training, in this work, we present novel ways in which synthetic data can be used for word embedding tasks which gives a significant boost in the performance by enriching the holistic representation of both image and its textual labels.

The major contributions of this work are built on top of HWNet v2 network [15] for learning efficient word image representation. In this work, we use these learned features for word attribute embedding which enables both word spotting and recognition. We also present a novel End2End embedding framework which learns a common subspace of image and text representation using a multi-task loss function. Finally, we also validate the role of synthetic data for improving word recognition at both line and word levels using a CRNN based architecture. In almost all cases, we report the state of the results on standard handwritten datasets. On the challenging IAM and GW handwritten datasets, we report an mAP of 0.9509 and 0.9898 respectively for query-by-string based retrieval task. Under lexicon based word recognition, our proposed

method report a 2.66 and 5.10 CER and WER respectively. We now briefly present some prominent related works using deep learning based methods for building word image representation.

With the advancements in deep learning, there is a paradigm shift in feature engineering where features are learned on the fly while training. In [20], Jaderberg et al. proposed three different architecture models (char, nGram and dictionary words) for scene text recognition. In [4], Poznanski et al. adapted VGGNet [21] for the recognition of PHOC attributes by having multiple parallel fully connected layers, each one predicting PHOC attributes at a particular level. In similar directions, different architectures [2], [3], [14] were proposed using CNN networks which embeds features into different textual embedding spaces. In [3], Sudholt et al. propose an architecture to directly embed image features to PHOC attributes by having sigmoid activation in the final layer and avoiding multiple fully connected layers. It is referred as PHOCNet, which uses the final layer activation to derive a holistic representation for word spotting. In [14], the deep CNN features obtained from the penultimate layer of HWNet architecture are embedded into word attribute space by training attribute based SVM classifiers and projecting both image and textual attributes to a common subspace. In [2], the authors propose a two stage architecture where a triplet CNN network is trained using SoftPN loss function. The learned image representation is embedded into a word embedding space (either PHOC, DCTOW, ngram etc) using a fully connected neural network and finally, the loss is defined using a cosine based embedding function. Most of the above works use the output activation from the penultimate layer of the CNN network as the word features to perform spotting and retrieval. More recently, Sudholt et. al. [13] extended the PHOCNet network by adding temporal pooling layer (TPP-PHOCNet) and evaluated different textual embedding and loss functions. An another recently published work which is close to ours is from Gomez et. al. [22] which learns an embedding space which respects Levenshtein distance between the samples.

II. HWNet Embedding

Label embedding [11], [12] for text images presents a generic framework to recast a recognition problem as retrieval. The basic idea is to embed both images and its labels (text) into a common subspace which respects lexical similarity across domains/modalities (image and text) and, finding the nearest neighbors in this space, enables both word spotting and recognition. The key question for optimum label embedding lies in three parts:- (i) finding a good representation of images and (ii) deriving a similar representation for text and (iii) finding the common subspace to learn the similarity metric across modality. In this section, we present the embedding scheme adapted from [11], [14] using robust word image features obtained from the improved version of HWNet architecture originally proposed in [1].

A. Image Embedding

We use the improved HWNet architecture (HWNet v2 [15]) as our image level features for the label embedding task. It consists of a ResNet34 network with four blocks where each block contains multiple resnet modules, an ROI pooling layer and two fully connected networks. Here each ResNet module consists of two convolutional layers and a shortcut connection to enable residual learning. We also use two layers of fully connected networks towards the end instead of global average pooling (as originally proposed for ResNets) in order to better capture the features learned in the penultimate layer. To improve generalization and also converge faster, we use batch normalization after each convolutional and fully connected layer (FC) except the last one. The key distinction of HWNet v2 architectures with other deep word embedding networks is that it uses multi-scale training which is enabled using an ROI pooling layer before its FC layers. The input word images are resized at random multiple scales on a padded image of size 128×384. While placing the image, we tend to avoid distorting the aspect ratio of word images except in few rare cases where the image width is greater than 384. This scheme enables the ability to train at multiple scales which is typically observed for multiple writers scenario and thereby the network remains invariant to different scales while testing. The use of ROI pooling after the last convolutional network preserves only the valid activations coming from the region where the input word image is rendered and extracts a fixed dimensional representation which is further given to the FC layers. The training objective is formulated as word classification problem using a multinomial logistic regression loss and the weights are updated using a mini-batch gradient descent with momentum. To improve generalization, we train the network from scratch using IIIT-HWS synthetic dataset [19] and later fine tune it on a real dataset to reduce the domain gap between synthetic and real-world data. The activations from the penultimate layer of the network are taken as word image embedding after performing the L_2 normalization.

B. Synthetic Attribute Embedding

In [14], it was shown that deep features can be used for word attribute embedding [11], which projects both image and text into a common Euclidean space where spotting and recognition tasks are treated as a nearest neighbor search. For attribute embedding, we use the Pyramidal Histogram of Characters (PHOC) where a word is divided into multiple levels and at each level, the histogram of characters and bigrams are computed and concatenated. For more details on PHOC representation, the authors can refer to [11]. The PHOC based attribute representation for a word label (text) can be computed directly, while for a word image, it can be learned by building attribute level binary classifiers using the training data as demonstrated in [11], [14]. Given the attribute representation for both images and text, we project them into a common subspace using subspace regression (CSR),

to capture the correlation between the attributes present in both modalities. In the common subspace where image and its corresponding labels lie close to each other, one can search using text (QBS) or images (QBE), to perform word spotting and recognition. In this work, we propose to use an additional synthetic modality to enrich the attribute embedding process and observe that it gives complementary information. Synthetic data for word images [19] is easy to generate and is available for most of the languages. In this particular setting, we only demand word images rendered in one synthetic font for each language. The basic idea is to compose the query representation using both textual and visual modality, and thereby exploiting the complementary information. Therefore while testing, the textual embedding computed using PHOC, is fused with its corresponding synthetic image embedding in a weighted manner. We refer to this as synthetic attribute embedding which we found to be much better for query-by-string retrieval.

III. End2End Embedding

The idea of synthetic attribute embedding, discussed in the previous section is a two-stage approach where the image embedding is learned in the first pass while the projection into the attribute space happens in the second pass. We now propose an End2End trainable deep word embedding network with two main motivations:- (i) replacing the attribute classifiers using multi-layer fully connected networks and (ii) to validate the learning of attribute space rather than fixing it to be PHOC. Note that, although PHOC has been shown the optimum attribute representation of word images and text, in this work, we investigate the possibility of automatic learning of such representation.

Figure 2 presents the proposed End2End deep convolutional network for simultaneous learning of both textual and image embeddings. It consists of three streams of feature extraction layers: one for the real word images and two for the label associated with it. The label information is given in two streams, which includes the synthetic image rendered in one single font and the textual stream which is given using PHOC representation. The features of both real images and synthetic images are captured using convolutional layers, where we use ResNet34 architecture for real images and a simpler network similar to AlexNet is used for synthetic image stream. The choice is made by understanding the complexity level of data variations of individual streams. Both these streams are given variable sized input images placed in a fixed padded image of size 128×384. Here also, we use ROI pooling after the last convolutional layer similar to the network described in the previous section. The textual stream consists of the network comprising of a PHOC extractor which is appended to the synthetic stream and is treated as a conditional label. This is achieved by concatenating the vectorized features activation of ($batchSize \times \#features$) last convolutional layer of the synthetic stream and the PHOC based textual representation. After concatenation,

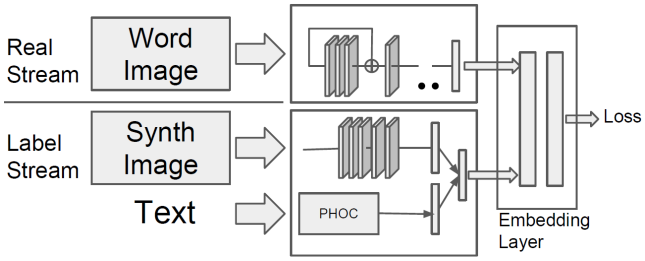

Figure 2. End2End convolutional network for learning both image and textual embedding using a multi-task loss function.

we use a fully connected network to merge both information which should optimize to preserve valid information from both modalities. Note that the weights of individual streams are not shared because of two main reasons:- (i) we need both real and synthetic (along with its text) stream to learn complementary features and (ii) to learn real stream without any conditional label information.

Finally, we have two sets of features, one from the real stream and another from label stream (synth+text). We now perform label embedding by projecting both these features into a common subspace. We achieve this using the embedding layer as shown in the figure, which is a typical Siamese style network implemented as multi-layer perceptron. The weights are shared since we want to identify the common subspace where the correlation among similarly paired data is maximized. We use a multi-task loss function as given below:-

$$\mathcal{L}(X, Y, \bar{X}, \bar{Y}) = \mathcal{L}_1(\hat{X}, \bar{X}) + \mathcal{L}_2(\hat{Y}, \bar{Y}) + \mathcal{L}_3(X, Y) \quad (1)$$

Here, X, Y are the features obtained from real and label stream respectively while $\bar{X}, \bar{Y}$ are the ground truths represented using one hot representation. The first two components of the loss function are the classification loss ($\hat{X}, \hat{Y}$ are the soft-max scores for real and label embedding respectively) which is a cross entropy based loss function while the third component is similarity loss function, which is defined using the cosine embedding between the pairs of features belonging to the same label. The loss is given as:-

$$\mathcal{L}_3(X, Y) = 1 - \cos(E(X), E(Y)) \quad (2)$$

Here $E(.)$ denotes the embedding function. The choice of multi-task loss was done following our experience with learning HWNet, which convinced us that the features learned while training a word classification network are robust enough to perform word spotting. Secondly, we chose a simpler cosine based loss function instead of contrastive loss [23], because in our experiments, we found such a network slow to train and the selection of pairs (positive and negative) is extremely crucial for optimum training. Note that using cosine loss, we achieved slightly better performance than contrastive loss, however, we believe with careful selection of hard negative samples, one can have better learning using the contrastive loss itself. More details on the implementation are discussed in Section V-D.

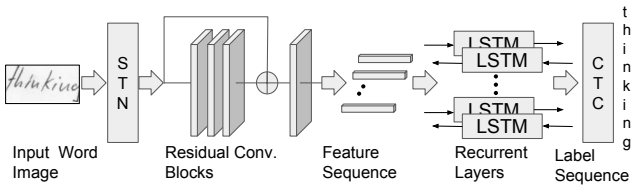

Figure 3. A CRNN style network pre-trained using synthetic data for performing word recognition.

IV. Word Recognition using CRNN Network

To perform word recognition, we use a CRNN [16] style architecture with a spatial transformer layer (STN) [24] as shown in Figure 3. The role of the STN layer is to correct geometric transformations, while the convolutional layers (ResNet18 [25]) is used for learning a sequence of feature maps. These feature maps are then used as input to the recurrent layers, which consist of stacked bi-directional LSTM's (BLSTM) [17]. It takes each frame from the feature sequence generated by the convolutional layers and provides probability distribution over the class labels. Finally, the CTC layer [26] decodes target label sequence from the set of probabilities computed from all the frames. In our experiments, we pre-train the network using IIIT-HWS dataset [19] for better generalization. Once the model converges on the synthetic dataset, we fine-tune the network on real handwritten images. We refer this model as CRNN-STN$_{synth}$.

V. Experiments

We evaluate the proposed embedding schemes for the task of word spotting and recognition. In word spotting, we design the protocol similar to [11] and also follow the train-val-test splits given along with each dataset. We perform our evaluation in a case-insensitive manner and also remove stopwords from the test query set, however, stopwords are kept in retrieval set to act as outliers. For query-by-example (QBE) setting, we test with each exemplar image from the test query set while for query-by-string (QBS) scenario, we take the unique strings in the test set as queries. We evaluate word spotting using mean average precision (mAP), which is the standard evaluation method under retrieval problems.

The second major experiment is on word recognition, which is evaluated using mean character error rate (CER) and mean word error rate (WER). Both CER and WER are computed using the Levenshtein distance between the predicted sequence of characters/words with actual ground truth respectively. To compare a wide spectrum of existing methods, we present our recognition results under two different scenarios:- (i) lexicon-based and lexicon-free and (ii) recognition at the line and word levels.

A. Datasets

The IAM Handwriting Database [9]: It contains a total of 1,539 handwritten forms written by 657 writers and is categorized as part of a modern collection. The database is labeled at the sentence, line and word levels. We use the official partition for writer independent text line recognition that splits the pages into training, validation, and testing sets, which are writer independent.

Table I
THE LIST OF DATASETS USED IN THIS WORK. HERE GW, BOTANY AND KONZILSPROTOKOLLE DATASETS ARE HISTORICAL DOCUMENTS WRITTEN PRIMARILY BY A SINGLE AUTHOR AND PROBABLY ALONG WITH A FEW ASSISTANTS(*).

Dataset	Historical	#Words	#Writers
IAM	No	1,15,320	657
GW	Yes	4,894	1*
Bentham	Yes	1,54,470	1*
Botany	Yes	20,004	1*
Konzilsprotokolle	Yes	12,993	1*

George Washington (GW) [10]: It contains 20 pages of letters written by George Washington and his associates in 1755 and thereby categorized into a historical collection. The images are annotated at the word level and contain approximately 5,000 words. Since there is no official partition, we use a random set (similar to [11]) of 75% for training and validation, and the remaining 25% for testing.

Botany and Konzilsprotokolle [27]: These two datasets are part of ICFHR 2016 Handwritten Keyword Spotting Competition [27]. We considered only the segmentation based track data which contained cropped word images split into training and test sets. There were also three partitions of training sets: small, medium, and large. Here we took only the largest partition which contains 16, 686 training images for Botany and 9,102 for Konzilsprotokolle. And the test set contains 3,318 word images for Botany and 3,891 for Konzilsprotokolle. These two datasets are also categorized under historical document collection.

B. Word Spotting Results

Table II presents the results of word spotting under various proposed embedding methods and comparisons made with other recent state of the art methods. Most of the compared methods are based on deep neural networks except the first method KSCR [11], which uses Fisher based representation for attribute embedding. As one can observe, the introduction of deep features as presented in HWNet [1], gave a significant push in the performance on all handwritten datasets. Further to it, with the introduction of attribute based label embedding [3], [4], [14], one can now perform both QBE and QBS seamlessly. In terms of performance, we observe that the embedded image and text attributes preserve the notion of distance similarity between image and its corresponding labels. Methods shown from row 2-7 presents the mAP of some prominent methods in this space.

We first compare the performance of HWNet v2 network, which is used as a base network for learning image features in the proposed methods. Here we observe a significant boost in terms performance where we now report an mAP above 0.90 for IAM which is the largest collection of dataset in terms of writers. Here we improve

Table II
QUANTITATIVE EVALUATION OF WORD SPOTTING ON STANDARD HANDWRITTEN DATASETS. WHILE EVALUATING [13], WE USED THE BEST PERFORMING SCHEMES AMONG THE VARIOUS LOSS FUNCTIONS AND TEXTUAL DESCRIPTORS THAT WERE TESTED BY THE ORIGINAL AUTHORS.

Method	IAM		GW		Botany		Konzilsprotokolle	
	QBE	QBS	QBE	QBS	QBE	QBS	QBE	QBS
KCSR [11]	0.5573	0.7372	0.9304	0.9129	0.7577	0.6569	0.7791	0.8291
HWNet [1]	0.8061	-	0.9484	-	0.8416	-	0.7913	-
PHOCNet [3]	0.7251	0.8297	0.9671	0.9264	0.8969	0.7447	0.9605	0.9420
Triplet-CNN [2]	0.8158	0.8949	**0.9800**	0.9369	0.5495	0.0340	0.8215	0.1219
DeepEmbed [14]	0.8425	0.9158	0.9441	0.9284	-	-	-	-
LSDE [22]	-	-	-	0.9131	-	-	-	-
TPP-PHOCNET [13]	0.8274	0.9342	0.9778	0.9802	0.9123	0.9506	**0.9770**	**0.9728**
HWNet v2 [15]	**0.9065**	-	0.9601	-	0.9401	-	0.9427	-
DeepEmbed	0.9038	0.9404	0.9801	0.9886	**0.9546**	0.9717	0.9411	0.9065
Synth+DeepEmbed	-	**0.9509**	-	**0.9898**	-	**0.9718**	-	0.9143
End2End Embed	0.8907	0.9126	**0.9814**	0.9742	0.9482	0.886	0.9296	0.7100

our results by a good margin as compared to the most recent state of the art method [13], which uses PHOC as the target embedding space along with spatial pyramid pooling layers. We see a similar pattern in other datasets except for Konzilsprotokolle where we get comparable results.

The proposed embedding schemes as described in this work are shown in the last three rows of the table. Here we first validate the features of HWNet v2 for attribute level embedding using the framework proposed in [14]. We refer this as DeepEmbed (using HWNet v2). As expected, with better features the performance has improved from previous attribute based methods in both QBE and QBS setting across all datasets. The mAP performance for query-by-string in IAM and Botany datasets is reported at 0.9404 and 0.9717 respectively. We now describe the results from synthetic embedding scheme as presented in Section II-B for QBS setting which combines both PHOC based representation with its corresponding synthetic image attributes computed from HWNet v2 for preparing the query representation. This results in further improvement of the performance, where we report state of the art results on IAM and GW at 0.9509 and 0.9898 respectively. Note that the previous state of the art result [2] on GW uses external real world CVL database for pre-training the network whereas the current method only uses synthetic data which is available almost free of cost.

Finally, we compare the proposed End2End embedding scheme which directly learns the attribute space using the network shown in Section III. Here we notice that, although we improve our results with respect to previous methods, we perform just comparable with our other proposed methods which use synthetic embedding on HWNet v2 features. We believe that this could be because of the increased number of parameters in the overall network which demands more data to generalize well. We further believe that the performance could be further improved with careful selection of hyper-parameters.

C. Word Recognition Results

Table III presents the word recognition results across various methods and compare it with the proposed method CRNN-STN$_{synth}$ under different settings. The popular settings are:- (i) level of segmentation (words/lines), (ii) use of test lexicon for recognition and (iii) use of language models while decoding the output. The first block of experiments compare the results under first setting. Here we observe the proposed DeepEmbed method using HWNet v2 features, to improve both WER and CER as compared to other deep features presented in [4], [14]. The use of CRNN-STN$_{synth}$ architecture gives the state of the art results under the setting of lexicon based word level segmentation which validates the importance of pre-training with synthetic data and the STN layer. Figure 4 shows qualitative results from CRNN-STN$_{synth}$ based word recognition network along with few failure cases. Notice the complexity of different handwriting and the prediction performance.

Table III
WORD RECOGNITION RESULTS ON IAM DATASET UNDER DIFFERENT SETTING OF EVALUATION SUCH AS LEVEL OF SEGMENTATION, USE OF LEXICON AND LANGUAGE MODEL FOR MAKING THE PREDICTIONS.

Method	Seg.	Lexicon	WER	CER
Almazán et al. [11]	Word	Based	20.01	11.27
Arik et al. [4]			6.45	3.44
DeepEmbed (HWNet) [14]			6.69	3.72
DeepEmbed (HWNet v2)			5.46	3.00
CRNN-STN$_{synth}$			**5.10**	**2.66**
CRNN-STN$_{synth}$		Free	16.19	6.34

Method	Seg.	Lang. Model	WER	CER
Bluche et al. [5]	Line	Yes	11.90	4.90
Doetsch et al. [6]			12.2	4.70
Voigtlaender et al. [28]			12.7	4.8
Pham et al. [8]			13.6	5.1
Voigtlaender et al. [7]			**9.3**	**3.5**
Pham et al. [8]		Free	35.1	10.8
CRNN-STN$_{synth}$			**32.89**	**9.78**

The second block of experiments in the table focuses mostly on the line level segmentation and the use of language model while prediction. In this work, we didn't use any language model which resulted in an inferior prediction, however, we still compare our method with the other popular methods in this space. The true comparison of our method in this setting is with [8], which shows results on both line level and without a language model. In this setting, we report our method at 9.78 CER which is better than [8]. The other observation one could directly make from the table is that, using language model significantly boosts the performance of line level prediction.

Figure 4. Qualitative results of word recognition on IAM dataset.

D. Implementation Details

We use extensive data augmentation while training our networks. In addition to usual data jittering techniques such as translation, shear, and resizing the images, we also perform elastic distortion [29] which is found to be very beneficial for handwriting modality. In our experiments using PHOC, we extract unigrams at 10 levels and the bigrams are extracted upto 6 levels. We use NVIDIA GeForce GTX 1080 Ti GPUs for all our experimentation and the codes are written in PyTorch and Torch libraries.

VI. Conclusion

In this work, we presented HWNet v2 architecture for efficient word image representation which enabled state of the art attribute embedding. We demonstrated an End2End embedding framework which efficiently uses synthetic image and textual representation to jointly learn complementary information for embedding of images and text. We further improved the performance of word recognition using a CRNN architecture, utilizing the STN layer and synthetic data. As part of future work, we plan to work with language models for line level recognition and carefully select the hyper-parameters of End2End network for learning much better models.

Acknowledgement

This work was partly supported by IMPRINT project, Govt. of India. Praveen Krishnan is supported by TCS Research Scholar Fellowship scheme.

References

[1] P. Krishnan and C. V. Jawahar, "Matching handwritten document images," in *ECCV*, 2016.

[2] T. Wilkinson and A. Brun, "Semantic and verbatim word spotting using deep neural networks," in *ICFHR*, 2016.

[3] S. Sudholt and G. A. Fink, "PHOCNet: A deep convolutional neural network for word spotting in handwritten documents," in *ICFHR*, 2016.

[4] A. Poznanski and L. Wolf, "CNN-N-Gram for handwriting word recognition," in *CVPR*, 2016.

[5] T. Bluche, H. Ney, and C. Kermorvant, "A comparison of sequence-trained deep neural networks and recurrent neural networks optical modeling for handwriting recognition," in *SLSP*, 2014.

[6] P. Doetsch, M. Kozielski, and H. Ney, "Fast and robust training of recurrent neural networks for offline handwriting recognition," in *ICFHR*, 2014.

[7] P. Voigtlaender, P. Doetsch, and H. Ney, "Handwriting recognition with large multidimensional long short-term memory recurrent neural networks," in *ICFHR*, 2016.

[8] V. Pham, T. Bluche, C. Kermorvant, and J. Louradour, "Dropout improves recurrent neural networks for handwriting recognition," in *ICFHR*, 2014.

[9] U. Marti and H. Bunke, "The IAM-database: an English sentence database for offline handwriting recognition," *IJDAR*, 2002.

[10] T. M. Rath and R. Manmatha, "Word spotting for historical documents," *IJDAR*, 2007.

[11] J. Almazán, A. Gordo, A. Fornés, and E. Valveny, "Word spotting and recognition with embedded attributes," *PAMI*, 2014.

[12] J. A. Rodriguez-Serrano, A. Gordo, and F. Perronnin, "Label embedding: A frugal baseline for text recognition," *IJCV*, 2015.

[13] S. Sudholt and G. Fink, "Evaluating word string embeddings and loss functions for cnn-based word spotting," in *ICDAR*, 2017.

[14] P. Krishnan, K. Dutta, and C. V. Jawahar, "Deep feature embedding for accurate recognition and retrieval of handwritten text," in *ICFHR*, 2016.

[15] P. Krishnan and C. V. Jawahar, "HWNet v2: An efficient word image representation for handwritten documents." *arxiv*, 2018.

[16] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *PAMI*, 2016.

[17] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *PAMI*, 2009.

[18] T. Bluche, J. Louradour, and R. Messina, "Scan, attend and read: End-to-end handwritten paragraph recognition with mdlstm attention," *arXiv preprint arXiv:1604.03286*, 2016.

[19] P. Krishnan and C. V. Jawahar, "Generating Synthetic Data for Text Recognition," in *arxiv*, 2016.

[20] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Synthetic data and artificial neural networks for natural scene text recognition," *CoRR*, 2014.

[21] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, 2014.

[22] L. Gómez, M. Rusinol, and D. Karatzas, "Lsde: Levenshtein space deep embedding for query-by-string word spotting," in *ICDAR*, 2017.

[23] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *CVPR*, 2005.

[24] M. Jaderberg, K. Simonyan, A. Zisserman *et al.*, "Spatial transformer networks," in *NIPS*, 2015.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[26] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *ICML*, 2006.

[27] I. Pratikakis, K. Zagoris, B. Gatos, J. Puigcerver, A. H. Toselli, and E. Vidal, "ICFHR2016 handwritten keyword spotting competition (H-KWS 2016)," in *ICFHR*, 2016.

[28] P. Voigtlaender, P. Doetsch, S. Wiesler, R. Schlüter, and H. Ney, "Sequence-discriminative training of recurrent neural networks," in *ICASSP*, 2015.

[29] P. Y. Simard, D. Steinkraus, and J. C. Platt, "Best practices for convolutional neural networks applied to visual document analysis," in *ICDAR*, 2003.