A Probabilistic Retrieval Model for Word Spotting based on Direct Attribute Prediction

Eugen Rusakov, Leonard Rothacker, Hyunho Mo, and Gernot A. Fink
Department of Computer Science
TU Dortmund University
44221 Dortmund, Germany
Email:{eugen.rusakov, leonard.rothacker, hyunho.mo, gernot.fink}@tu-dortmund.de

***Abstract*—In recent years CNNs took over in various fields of computer vision. Adapted to document image analysis, they achieved state-of-the-art performance in word spotting by predicting word string embeddings. One prominent embedding splits a given string in temporal pyramidal regions of character occurrences, namely the *Pyramidal Histogram of Characters (PHOC)*. This string embedding can be interpreted as a binary attribute representation. In this work we present a new approach for ranking retrieval lists originally proposed for zero-shot learning where attribute representations play an important role. Instead of a distance-based matching of the predicted string embedding, we compute the posterior probability of the attribute representation given a word image which can be interpreted as a posterior of the query. We can show that this probabilistic ranking improves word spotting performance, especially in the query-by-string scenario.**

I. Introduction

Given a user defined query, the task of word spotting is to retrieve a list containing word images that are relevant with respect to the query. Typically word spotting methods rank all retrieved word images from a given document collection by a certain criterion and sort them by their similarities. These queries can either be word images, defined by a user cropping a snipped from a document page or defining a word string which needs to be retrieved. Based on these two approaches of query definitions, word spotting distilled two scenarios namely Query-by-Example (QbE) and Query-by-String (QbS).

For QbE, queries are given as word images and retrieval is based on comparing the query word image representation with all other representations. The main challenge in the QbE scenario is to define a suitable measure of similarity for word images. In the literature, several query representations based on visual similarity have been proposed [1], [2], [3]. In the QbS scenario the queries are given as word strings. In contrast to QbE the user is not required to search for a visual example of a word in the document collection, as this search can be exhaustive if there is only one example in between thousands of other words. A drawback for the system is the requirement of a mapping from a textual to a visual (and vice versa) representation. As such appearance models need to be estimated from training data, annotated word images are necessary for obtaning a QbS model.

A very influential approach for learning a mapping from a visual to textual representation was presented in [1]. The authors proposed a word string embedding to project word images into a common space based on a binary attribute representation with a pyramidal partitioning of a word string which they named *Pyramidal Histogram of Characters (PHOC)*. Due to the great success of the PHOC word embedding, recently the focus concerning feature representations shifted towards trainable feature extractions, as this embedding was originally proposed using heuristic feature extractions. Inspired by an outstanding recognition performance of Convolutional Neural Networks (CNNs) in many image classification tasks [4], [5], the concept of embedding word strings into a vector space was extended by replacing heuristically designed feature representation with jointly trainable CNNs. In [6], Sudholt et. al. proposed the TPP-PHOCNet based on the PHOC representation, which achieved state-of-the-art results in word spotting tasks. Next to the TPP-PHOCNet several interesting approaches concerning word image and string embeddings were proposed, achieving comparable results. In [7] the authors applied a discrete cosine transform to a one-hot string encoding namely the *Discrete Cosine Transform of Words (DCToW)*. Another approach was presented in [8], where a deep feature representation is learned using the activations of the second fully connected layer of a CNN as holistic word image representations. In contrast to [7] and [8] the TPP-PHOCNet can be trained in an end-to-end fashion using a binary logistic regression, and thus each attribute classification can be interpreted as a probability of character occurrence. In this work we combine this interpretation with an approach proposed by Lampert et. al. [9], where the authors train a classifier predicting attributes which are shared across all training-classes. Based on this prediction a probability can be estimated to classify test-classes which have not been seen during training. Our approach is based on a probability ranking of retrieval lists, as the method in [9] can be applied to binary attributes of the PHOC representation predicted by the TPP-PHOCNet, which we named *Probabilistic Retrieval*

Model (PRM). We can show that this combination improves the performance of our word spotting system in both QbE and QbS scenarios.

II. Direct Attribute Prediction

The concept of attribute-based representations was introduced to the field of computer vision in 2009 [10], [11]. Lampert *et al.* leverage attributes to tackle the problem of classifying classes which are not part of the training set also known as *zero-shot* learning [10], [9]. They propose a method called *Direct Attribute Prediction (DAP)*. The key idea of this method is the use of an attribute representation that allows to uniquely define pattern classes with corresponding attribute configurations. Instead of predicting class labels directly from image data, classifiers are trained for predicting the individual attributes. Based on the attribute prediction, the DAP method then computes the posterior of an object class z – irrespective whether this was seen during training or not – by computing its posterior given an image $\mathbf{x}$ (see [9, Sec. 2.2.1]):

$$p(z|\mathbf{x}) = \sum_{\mathbf{a}} p(z|\mathbf{a})\, p(\mathbf{a}|\mathbf{x}) \tag{1}$$

The most important part of this probabilistic classification model is the attribute prediction. Lampert *et al.* [9] use a model that combines several well known heuristic feature representations for natural images and Support Vector Machines (SVM) as classifiers. Each SVM is trained to predict one attribute. In order to be able to interpret the result as a probability prediction, Platt scaling is used [12].

III. Word Spotting with Attribute-Like Representations

In analogy to [9], approaches to classify word classes with CNNs are not completely feasible in the field of text recognition. For example if a text recognition task contains $100\,000$ different word instances, treating each single word as an independent class results in a one-hot encoding with $100\,000$ classes. In [13], Jaderberg et. al. demonstrated the problem in so-called dictionary learning. Here, the authors trained a model to classify over $90k$ different word classes in a one-hot encoding manner and found that an iterative training is required for convergence. Furthermore, a database containing many samples for each word class is necessary. As text recognition is an important part of the methodology concerning word spotting, several attribute-like representations were proposed.

In [6], several word string embeddings were evaluated. The first embedding is the PHOC [1] representation. Here, the characters are encoded as binary attributes in a histogram of character occurrences at different splits. The PHOC consists of levels representing a spatial pyramid, where the string is partitioned in splits equivalent to the level. In the first level the string is not split, instead the whole string is considered to build the first level histogram. The second level splits the given string into two halves (e.g. the word "home" results in "ho" and "me") and builds two histograms of character occurrences, and so on. Hence the PHOC levels $[1 + 2 + 3 + 4 + 5]$ are concatenated to a vector of 15 splits. Another evaluated embedding is the Spatial Pyramid of Characters (SPOC) [6]. This embedding can be seen as a multinomial generalization of the PHOC, where the correspondence of characters is counted [6]. Similar to the PHOC a SPOC representation contains pyramidal levels, where each level is split into partitions equivalent to the level (e.g. one partition in the first level, two partitions in the second level and so on). In contrast to the PHOC each split is built on a *Bag-of-Characters (BoC)* meaning a histogram of character counts. Finally, the levels are concatenated to a SPOC representation. In [6] the SPOC is a PHOC with counts instead of binary character occurrences. The last embedding is the DCToW originally proposed in [7]. Here, each character is represented in a one-hot encoded vector with respect to the alphabet. The vectors are stacked to a $K \times m$ matrix, where K is the size of the alphabet and m is the length of the given word. A discrete cosine transform is applied to each row of the matrix, afterwards only the highest three values are extracted per row.

Both the SPOC and DCToW constitute real-valued representations and hence can not be used in combination with the DAP method, as DAP requires an attribute prediction which can be interpreted as a vector of individual attribute probabilities. On the other side the PHOC representation is suitable for DAP using the *Binary Cross Entropy (BCE)* as loss function.

IV. Method

In this section, we describe our proposed method used to evaluate two benchmarks. At first, a brief introduction to the TPP-PHOCNet is given, describing the architecture and design choices. Afterwards, we introduce the PRM which computes probabilities for user defined queries, based on attribute predictions.

A. STPP-PHOCNet

In this section, we briefly revisit our design choices and important aspects concerning the STPP-PHOCNet. Originally the PHOCNet [14] was proposed based on the idea to replace the AttributeSVM [1] by a CNN. The architecture is inspired by the VGGnet [5]. One year later the PHOCNet was extended, replacing the Spatial Pyramid Pooling (SPP) layer [15] by a Temporal Pyramid Pooling (TPP) layer [6]. The TPP layer takes arbitrarily sized input feature maps and subdivides these in horizontal bins as defined by the TPP level. For example, the first level is equal to global pooling and the second level subdivides the feature map in two halves (left and right side). The third level subdivides a given feature map in three splits (left, middle, and right part), and so on. A global pooling is performed over each split, obtained by the temporal pooling. Higher levels behave equivalently and subdivide feature maps in as many splits as specified by the corresponding level. Sudholt et al. [6] show that splitting a feature map in temporal, i.e., horizontal, positions suites the prediction of PHOC attributes, as this resembles the structure of the PHOC embedding which splits the string horizontally as well. As in [6] we use TPP

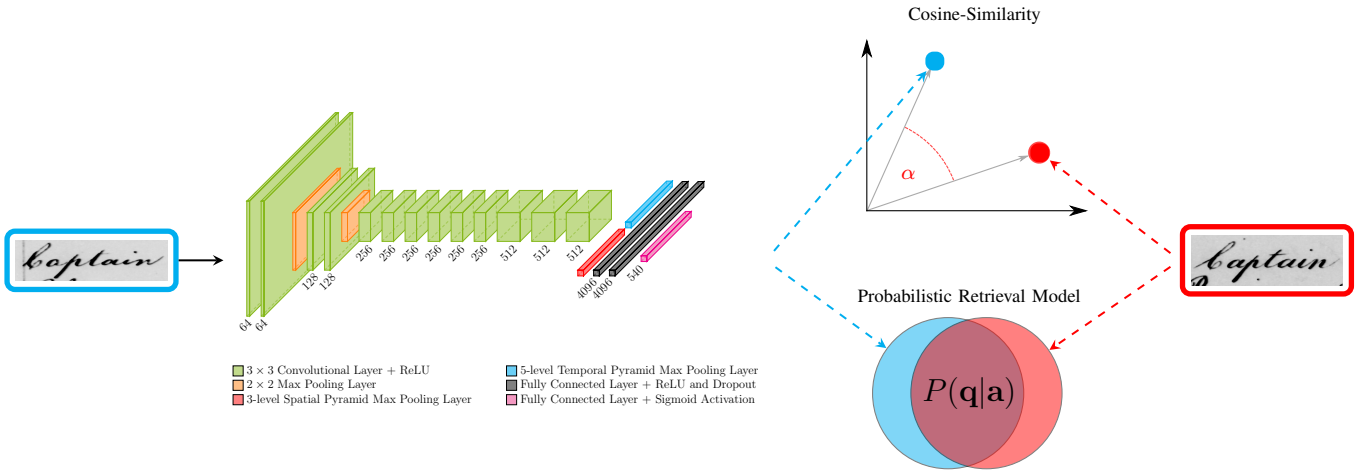

Figure 1. Overview of the attribute prediction framework using a CNN. The CNN is trained to predict a desired attribute prepresentation generated from the annotation. Word spotting can then be performed either by the Probabilistic Retrieval Model obtaining probabilities for each test image given a query or in the embedding space through a simple nearest neighbor search using the cosine similarity as the distance metric.

levels $[1+2+3+4+5]$. In addition to the TPP layer we keep the *SPP* layer. Similar to the temporal pooling, the SPP layer subdivides a feature map into quadratic grids. At level one a global pooling is performed, whereas on level two the feature map is divided into 2×2 splits resulting in 4 bins. At level three the feature map is subdivided into 4×4 splits, and so on. Here, we use SPP levels $[1+2+3]$ in ordner to obtain $[1^2+2^2+4^2] = 21$ bins. The TPP and SPP layers are both used in *parallel* after the convolutional layers and are concatenated before the fully connected layers. Due to the combination of the TPP and SPP layers we named our CNN *Spatial Temporal Pyramid Pooling PHOCNet*, i.e. *STPP-PHOCNet*.

For training we use the *BCE* loss, which is also known as binary logistic loss [6]. The BCE is computed to

$$l_{BL}(\mathbf{a}, \mathbf{y}) = -\frac{1}{D} \sum_{i=1}^{D} [y_i \log(a_i) + (1 - y_i) \log(1 - a_i)] \quad (2)$$

where $\mathbf{a}$ is the predicted attribute vector, $\mathbf{y}$ represents the desired attribute vector and D its dimensionality. a_i and y_i represent the i-th component in vector $\mathbf{a}$ and $\mathbf{y}$, respectively. The BCE loss is an important ingredient of our method as it can interpreted from a probabilistic point of view [6]: Let $\mathbf{x}^{(i)}$ be the i-th input vector in a set of n samples, $\mathbf{y}^{(i)}$ the corresponding desired attribute representation and θ the hyperparameters of the CNN, then using the BCE loss is equivalent to finding parameters $\hat{\theta}$ that maximize the likelihood of predicting $\mathbf{y}$ from $\mathbf{x}$ [6]:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \prod_{i=1}^{n} p(\mathbf{y}^{(i)} | \mathbf{x}^{(i)}, \theta) \quad (3)$$

In order to derive this interpretation of the BCE loss (Eq. 2 from 3), the assumption of pairwise independent elements in label vector $\mathbf{y}^{(i)}$ is made. Hence the attribute predictions of the PHOCNet, using BCE as loss function, can be interpreted as probabilities. Therefore, the CNN with BCE loss is better suited than an Attribute-SVM ensembles for interpreting outputs as probabilities. In the SVM framework values are merely scaled in the range $[0, \cdots 1]$ with a Sigmoid during Platt scaling [9]. The distribution of the data is considered only marginally because the SVM hyperplane is defined by the support vectors only.

B. Probabilistic Retrieval Model

When using the *cosine similarity* for ranking retrieval results, the score assigned to each retrieved hypothesis is based on the angle between the PHOC vector encoding of the query q and the PHOC representation predicted for some word image $\mathbf{x}$. In contrast to this, our proposed *Probabilistic Retrieval Model* (PRM), which is based on the DAP method described in Sec. II, exploits the fact that the attribute vector predicted for a word image can be interpreted as a vector of attribute probabilities. As each attribute can be considered as a binary random variable A_i, its behavior can be described by a Bernoulli distribution. Each output of the STPP-PHOCNet now computes an estimate $\hat{a}_i = p(A_i = 1|\mathbf{x})$ for the probability of the i-th attribute being present in word image $\mathbf{x}$. Assuming conditional independence among attributes, we obtain

$$p(\mathbf{a}|\mathbf{x}) = \prod_{i=1}^{D} p(A_i = a_i|\mathbf{x}) = \prod_{i=1}^{D} \hat{a}_i^{a_i} \cdot (1 - \hat{a}_i)^{(1-a_i)}. \quad (4)$$

For a string query q, the attribute vector is given in a deterministic way according to the construction principle of the PHOC. Therefore, we obtain attribute probabilities ${}^{q}a_i$ for

a query q as follows:

$$ {}^{q}a_i = p(A_i = 1|q) = \begin{cases} 1 & \text{if PHOC}(q)_i = 1 \\ 0 & \text{otherwise.} \end{cases} $$

In order to decide whether image $\mathbf{x}$ is relevant for a query q, we need to compute the probability of producing q from $\mathbf{x}$. Using the DAP method, this posterior can be derived in the following way (see Eq. 1):

$$ p(q|\mathbf{x}) = \sum_{\mathbf{a}} p(q|\mathbf{a})\, p(\mathbf{a}|\mathbf{x}) \tag{5} $$

The posterior of q given an attribute vector $\mathbf{a}$ can be rewritten using Bayes rule:

$$ p(q|\mathbf{a}) = \frac{p(q,\mathbf{a})}{p(\mathbf{a})} = p(\mathbf{a}|q)\,\frac{p(q)}{p(\mathbf{a})} \tag{6} $$

Plugging Eq. 6 into Eq. 5 and noting that $p(\mathbf{a}|q)$ is nonzero – i.e. equal to 1 – only for a single ${}^{q}\mathbf{a}$, namely the PHOC representation of q, we obtain

$$ p(q|\mathbf{x}) = \frac{p(q)}{p({}^{q}\mathbf{a})} p({}^{q}\mathbf{a}|\mathbf{x}). $$

In order to evaluate this equation, in addition to the probability of predicting ${}^{q}\mathbf{a}$ from $\mathbf{x}$, the prior probabilities $p(q)$ of the query and the attribute representation thereof, $p({}^{q}\mathbf{a})$, are principally required. It was, however, already observed in [9] that simply ignoring these priors or using uniform priors did not noticeably affect the results obtained. As uniform priors avoid any possibly wrong assuptions, and as it is not obvious how a prior over the potentially open set of string queries could be defined, we decided to not use any priors in our PRM. Therefore, it eventually boils down to evaluate Eq. 4 for ${}^{q}\mathbf{a}$.

As the PHOC representation is quite high-dimensional and the evaluation of the PRM via Eq. 4, consequently, requires to evaluate a product of several hundreds of probabilities, we compute the PRM score in the logarithmic domain in order improve numerical stability:

$$ \begin{aligned} \log p({}^{q}\mathbf{a}|\mathbf{x}) &= \log \prod_{i=1}^{D} \hat{a}_i^{\,{}^{q}a_i} \cdot (1-\hat{a}_i)^{(1-{}^{q}a_i)} \\ &= \sum_{i=1}^{D} {}^{q}a_i \log \hat{a}_i + (1-{}^{q}a_i)\log(1-\hat{a}_i) \end{aligned} $$

In Figure 1 an overview over the attribute prediction framework is given. The CNN (here STPP-PHOCNet) predicts the attributes of the PHOC representation. Afterwards, word spotting can be performed using either the PRM or a nearest neighbor search with cosine similarity in the word embedding space.

V. Experiments

For the experiments we used two benchmark datasets described in Sec. V-A and a evaluation protocol (Sec. V-B) for segmentation-based word spotting commonly used in the literature. In Sec. V-C we describe the training setup with all hyper-parameter used for training. Afterwards we discuss the retrieval results achieved by both ranking methods and compare them in Sec. V-D.

Table I
Number of training and test images in both datasets George Washington and IAM off-line dataset.

Data Set	George Washington				IAM
	$Split_1$	$Split_2$	$Split_3$	$Split_4$	
# Train Images	3696	3568	3677	3639	60453
# Test Images	1164	1292	1183	1222	13752

A. Datasets

We evaluate our method on two publicly available data sets. The first is the **George Washington (GW) data set**. It consists of 20 pages that are containing 4,860 annotated words. The pages originate from a letterbook and are quite homogeneous in their visual appearance. However, particularly for smaller words the annotation is very sloppy. As the GW data set does not have an official partitioning into training and test pages, we follow the common approach and perform a four-fold cross validation. Thus, the data set is split into batches of five consecutive documents each.

The second benchmark is the large **IAM off-line dataset** comprising 1,539 pages of modern handwritten English text containing 115,320 word images, written by 657 different writers. We used the official partition available for writer independent text line recognition. We combined the training and validation set in order to obtain 60,453 word images for training, and 13,752 word images in the test set for evaluation. We exclude the stop words as queries, but kept them as distractors for both QbE and QbS. Table I shows the number of training and test images for both datasets.

B. Evaluation Protocol

We evaluate the TPP-PHOCNet for the data sets GW and IAM in the segmentation-based word spotting standard protocol proposed in [1]. One training partition of each data set is used to train a single TPP-PHOCNet. For the QbE scenario in GW and IAM all test images, which occur at least twice in the test set, are considered as queries. A word string embedding is predicted from the STPP-PHOCNet for a given query as well as all other word images in the test set. Retrieval is then performed in two scenarios. In the first scenario we run a nearest neighbor search in the string embedding space, using the cosine similarity as recommended in [6]. The second scenario computes probabilities for all test set strings given a query, using the DAP method. Afterwards the retrieval lists are obtained by sorting the list items by their cosine similarities and probabilities, respectively. For QbS the retrieval is performed equivalently to QbE except that only unique strings from the test set are considered as queries.

Table II
RESULTS FOR THE QBE AND QBS EXPERIMENTS IN MAP [%].

Architecture	Loss	Similarity	George Washington		IAM	
			QbE	QbS	QbE	QbS
STPP-PHOCNet	BCE	Cosine	97.47	96.50	88.49	93.03
STPP-PHOCNet	BCE	DAP	97.76	96.89	**89.27**	**95.40**
TPP-PHOCNet [16]	BCE	Cosine	97.90	96.73	84.80	92.97
TPP-PHOCNet [16]	Cosine	Cosine	97.96	**97.92**	82.74	93.42
PHOCNet [16]	BCE	Cosine	97.58	95.58	85.50	92.38
PHOCNet [16]	Cosine	Cosine	97.72	97.44	75.85	91.12
Deep Features [8]			94.41	92.84	84.24	91.58
Triplet-CNN [7]			**98.00**[1]	93.69	81.58	89.49
AttributeSVM [1]			93.04	91.29	55.73	73.72

As in [6], we use the interpolated *Average Precision (AP)* as performance metric for each single query:

$$AP = \frac{\sum_{i=1}^{n} Prec(i) \cdot R(i)}{t} \tag{7}$$

Where $Prec(i)$ is the precision if we cut off the retrieval list after i elements. $R(i)$ is an indicator function, which computes to 1 if the i-th position of the retrieval list is relevant with respect to the query and 0 otherwise. The length of the retrieval list is given with n. And finally, t is the total amount of relevant elements. As customary for segmentation-based word spotting, the retrieval list contains all word images from the respective test set. Afterwards, the overall performance is obtained by computing the *mean Average Precision (mAP)* over all queries.

C. Training Setup

For training we use hyperparameters as in [6]. For the PHOC vectors we used the levels $[1 + 2 + 4 + 8]$. As attributes we choose a case insensitive latin alphabet with 26 characters and 10 digits resulting in 36 attributes per PHOC split. Based on the PHOC levels we get $1 + 2 + 4 + 8 = 15$ splits each containing 36 attributes and in summation a 540 dimensional PHOC vector. The TPP-PHOCNet was trained using a *BCE* loss and the *Adaptive Moment Estimation (Adam)* [17] optimizer. For the momentum the mean value β_1 is set to 0.9 and for the variance value β_2 is set to 0.999 while the variance flooring value is set to 10^{-8} as recommended in [17]. For the George Washington Database (GW) the initial learning rate is set to 10^{-4} and is divided by 10 after 70 000 training iterations, while running 80 000 iterations in total. As the initial learning rate for IAM Handwritten Database (IAM) is equal to GW, we decrease the learning rate by a factor of 10 after 100 000 training iterations, while running the training for 240 000 iterations . As parameter initialization, we use the strategy proposed in [18]. The weights are sampled from a zero-mean normal distribution with a variance of $\frac{2}{n_l}$, where n_l is the total number of trainable weights in layer l.

[1]results obtained with additional annotated training data [7]

As both databases are relative small, we use some augmentation techniques to extend the samples in both databases. At first we balance the training data, so every word instance appears at least 30 times. Followed by augmentation methods like perspective transformation, shearing, rotation, translation, scaling, lightness changing and noise generating techniques. In total we obtain 500 000 training images for GW and 1 100 000 for IAM.

D. Results & Discussion

Table II lists the results for the QbE and QbS experiments on two benchmarks. For the GW dataset the DAP method slightly improves the performance of the attribute prediction using the binary cross-entropy as loss function. As the results are already close to 100%, there is not much space for improvement. Furthmore, [6] shows that the TPP-PHOCNet trained with the cosine loss (CPS) achieves better results on the GW dataset compared to BCE. Using the Spatial Temporal Pyramid Pooling (STPP) layer does not result in any performance improvements on the GW dataset. As the GW dataset is relatively small (see Table I), using only a TPP layer is already sufficient in order to obtain a powerful feature representation after the finally convolutional layer.

For the IAM database, the performance increases by about 3% in the QbE scenario. Here, the DAP method shows more influence on the performance for both QbE and QbS. For QbE a mAP of 88.49% is achieved using the BCE loss function. The DAP method improves this performance to 89.27% (+0.78%). Thus, is can be assumed that the STPP layer has more influence on the performance. The layer consists of $36\,(15 + 21)$ bins compared to the 15 bins using only the TPP layer or 21 bins using only the SPP layer, as shown in [16]. One can argue that the more powerful feature representation of the STPP achieves better results, given a more complex dataset like IAM based on handwritings from different writers and hence less visual similarity of the same words.

VI. Conclusion

In this work, we compared two ranking methods for retrieval lists in word spotting for Query-by-Example and Query-by-String. The experiments show that the performance of a CNN predicting attributes given a word image can be improved using the *Direct Attribute Prediction (DAP)* method. This method effects the performance more on larger datasets like IAM, with a higher visual variability of word images. Additionally, the probabilistic model allows for a better interpretation of similarity scores compared to distances. This can be beneficial for users and also for automatic systems that incorporate word spotting results for further automatic analysis. Finally, we use a combination of the Temporal and Spatial Pyramid Pooling layers in order to obtain a more powerful feature representation. This allows us to outperform state-of-the-art results on the IAM database.

References

[1] J. Almazán, A. Gordo, A. Fornés, and E. Valveny, "Word spotting and recognition with embedded attributes," *TPAMI*, vol. 36, no. 12, pp. 2552–2566, 2014.

[2] T. Rath and R. Manmatha, "Word spotting for historical documents," *Int. Journal on Document Analysis and Recognition*, vol. 9, no. 2–4, 2007.

[3] M. Rusiñol, D. Aldavert, R. Toledo, and J. Lladós, "Efficient segmentation-free keyword spotting in historical document collections," *Pattern Recognition*, vol. 48, no. 2, pp. 545 – 555, 2015.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[5] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *Proc. of the Int. Conf. on Learning Representations*, 2015.

[6] S. Sudholt and G. Fink, "Evaluating word string embeddings and loss functions for CNN-based word spotting," in *Proc. of the ICDAR*, Kyoto, Japan, 2017, pp. 493–498.

[7] T. Wilkinson and A. Brun, "Semantic and verbatim word spotting using deep neural networks," in *Proc. of the ICFHR*, 2016, pp. 307 – 312.

[8] P. Krishnan, K. Dutta, and C. Jawahar, "Deep feature embedding for accurate recognition and retrieval of handwritten text," in *Proc. of the ICFHR*, 2016, pp. 289 – 294.

[9] C. H. Lampert, H. Nickisch, and S. Harmeling, "Attribute-based classification for zero-shot visual object categorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 453–465, 2014.

[10] ——, "Learning to detect unseen object classes by between-class attribute transfer," in *Proc. IEEE Comp. Soc. Conf. on Computer Vision and Pattern Recognition*, 2009, pp. 951–958.

[11] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, "Describing objects by their attributes," in *Proc. IEEE Comp. Soc. Conf. on Computer Vision and Pattern Recognition*, Miami, USA, 2009, pp. 1778–1785.

[12] J. C. Platt, *Probabilities for SV Machines*. MIT Press, 2000, pp. 61–74.

[13] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Synthetic data and artificial neural networks for natural scene text recognition," *arXiv preprint arXiv:1406.2227*, 2014.

[14] S. Sudholt and G. A. Fink, "PHOCNet: A deep convolutional neural network for word spotting in handwritten documents," in *Proc. of the ICFHR*, Shenzhen, China, 2016, pp. 277 – 282.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *European Conference on Computer Vision*, pp. 346–361, 2014.

[16] S. Sudholt and G. A. Fink, "Attribute cnns for word spotting in handwritten documents," *Int. Journal on Document Analysis and Recognition*, 2018, to appear.

[17] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. of the Int. Conf. on Learning Representations*, 2015.

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