# Graph Convolutional Neural Networks for Learning Attribute Representations for Word Spotting

Fabian Wolf[1] (✉)[0000−0001−8842−3718], Andreas Fischer[2,3][0000−0003−0069−3436], and Gernot A. Fink[1][0000−0002−7446−7813]

[1] TU Dortmund University, Department of Computer Science, 44227 Dortmund, Germany
{fabian.wolf, gernot.fink}@cs.tu-dortmund.de
[2] Department of Informatics, DIVA Group, University of Fribourg,
[3] Institute of Complex Systems, University of Applied Sciences and Arts Western Switzerland, Fribourg, Switzerland
andreas.fischer@unifr.ch

**Abstract.** Graphs are an intuitive and natural way of representing handwriting. Due to their high representational power, they have shown high performances in different learning-free document analysis tasks. While machine learning is rather unexplored for graph representations, geometric deep learning offers a novel framework that allows for convolutional neural networks similar to the image domain. In this work, we show that the concept of attribute prediction can be adapted to the graph domain. We propose a graph neural network to map handwritten word graphs to a symbolic attribute space. This mapping allows to perform *query-by-example* word spotting as it was also tackled by other learning-free approaches in the graph domain. Furthermore, our model is capable of *query-by-string*, which is out of scope for other graph-based methods in the literature. We investigate two variants of graph convolutional layers and show that learning improves performances considerably on two popular graph-based word spotting benchmarks.

**Keywords:** Graph Neural Networks · Geometric Deep Learning · Word Spotting

## 1 Introduction

The field of pattern recognition distinguishes the two principles of statistical and structural approaches [4]. For any application both approaches need to solve the problem of how to measure the similarity of different objects. Statistical approaches usually rely on numerically representing an object in the form of a high-dimensional feature vector. Measuring similarity is then feasible by common vector distances. Statistical approaches offer a mature framework of algorithms for clustering, retrieval or classification and have strongly benefited from the uprise of deep neural networks. However, the representational power of a vector is

limited, which motivates the structural approach. In this case, data is represented based on symbolic structures such as graphs. While graphs offer a more powerful data representation, they often lack the mathematical simplicity of Euclidean data. Already basic operations such as computing a distance between two graphs constitute a complex problem with high computational demand.

If data can be represented in a Euclidean manner, statistical approaches often dominate the field as in the case of computer vision [9, 22]. Structural approaches are more common in areas, where relational data is essential and a non-Euclidean data structure is the obvious choice. Popular areas in this regard are the analysis of chemical molecules and social or citation networks [3, 11, 34]. Looking at application areas of statistical and structural pattern recognition, handwriting analysis holds a special position. Document analysis methods are highly focused on the image domain as image acquisition is easy and a huge amount of well researched statistical approaches exist. Nonetheless, a structural representation is inherent to any image of handwriting. The underlying structure of a handwritten word can naturally be captured in the form of a graph. This makes it an open question whether handwriting analysis can benefit from a structural approach based on graph representations.

Word spotting is a task that attracted a lot of attention in the document analysis community and also represents an area where structural and statistical approaches coexist [9]. In general, the problem of word spotting is a well researched field in document image analysis and many mature methods exit. However, word spotting has also been a topic of interest with respect to graph representations [2, 18, 26]. As word spotting essentially requires to measure the similarity of a word to a query, many graph-based methods explored how to efficiently compute a distance between graphs [2, 25]. In terms of performance, a significant gap between image and graph-based approaches exists. This gap can be explained by the fact that most methods in the image domain heavily rely on learning and on training powerful models on labelled data. Additionally, handwriting graphs are usually extracted from images relying on binarization and skeletonization methods [24, 26]. This step might limit performances compared to models of the image domain working with unprocessed images.

Learning in the graph domain recently gained a lot of attraction with the generalization of the convolution operation to graph structures. Geometric deep learning [3] provides a framework similar to deep convolutional neural networks that led to significant performance gains for different benchmarks [11]. In [18], the authors propose a learning-based model for word spotting in the graph domain to estimate a graph distance with graph neural networks.

In this work, we propose a graph convolutional neural network to predict an attribute representation from handwritten word graphs. This approach has been proven to show high performances on word spotting benchmarks in the image domain [28]. Compared to other methods considering the graph domain, our learning-based approach exploits character level instead of only word class information. Furthermore, mapping graphs to an attribute space allows to query by string, which is not the case for other methods in the literature.

The remainder of the paper is organized as follows. Sec. 2 presents related works on the topic of word spotting in the image and graph domain. The proposed graph convolutional neural network is discussed in Sec. 3. In our experimental evaluation in Sec. 4, we investigate the influence of key components of our model on four different benchmarks. Finally, we compare the proposed model to other graph and image-based word spotting methods known from the literature.

## 2 Related Work

Word Spotting describes the task of retrieving regions from a document collection that are similar to a query [9]. In contrast to handwriting recognition, the result of a word spotting system is not an explicit transcription result, but a ranked list of possible word occurrences. This retrieval approach allows for interpretation by the user, making word spotting an attractive alternative especially for information retrieval from historic document collections. For an extensive overview on word spotting methodology and taxonomy, see [9].

### 2.1 Document Image Analysis

Traditionally, word spotting methods have been highly focused on the image domain. Different methods either work on entire document images [14, 20, 32] or segmented regions such as word images [13, 16, 28]. Several works on word spotting exploit the sequential structure of handwriting. Models such as recurrent neural networks [14] and Hidden Markov Models (HMM) [5, 20] were applied successfully and are still popular.

Traditional feature based approaches also attracted attention and were usually combined with models such as spatial pyramids [21] or HMMs to encode spatial information [16, 20]. As these methods measure visual similarity based on a designed representation, they usually are not capable to generalize well across high variations in writing styles. To overcome this drawback, Almazan et al. proposed to predict certain image properties so called attributes from word images in [1]. In this influential work, the authors proposed the *Pyramidal Histogram of Characters* (PHOC) that encodes the occurrence and spatial position of characters in a pyramidal fashion. By mapping word images to an attribute vector space, *query-by-example* word spotting boils down to the computation of simple vector distances. Since it is straightforward to derive a PHOC vector from a string, *query-by-string* is easily possible.

In [27], the attribute-based approach of [1] was adopted using a convolutional neural network that replaced the formerly used combination of Fisher Vectors and SVMs. Training a neural network to predict an attribute representations from word images, resulted in high performances on almost all popular benchmarks [15, 28]. Recently, image-based word spotting got increasingly more focused on methods that either do not rely on annotated training data [16, 29, 33] or are capable to jointly solve the segmentation problem [14, 32].

## 2.2 Graph Representations

While word spotting in the image domain is a highly researched topic that resulted in well performing methods, significantly less works considered the task from a structural perspective. Here, we focus on methods that first extract a graph representation, in order to tackle the word spotting problem in the graph domain. In [30], the authors propose a graph representation that extracts vertices and edges from skeletonised word images to represent the structural properties of a handwritten word. This representation is enriched by using the Shape Context Descriptor as an additional node feature vector. In order to measure similarity, an approach based on dynamic time warping (DTW) and an approximated Graph Edit Distance (GED) is proposed. Most graph-based methods follow a similar approach. First, a graph representation is extracted from a word image, for example by computing keypoints [6] or projection profiles [24, 25]. Then, the similarity of a query graph to all word graphs is estimated by a graph distance. As common graph distances such as the GED are highly computational demanding, most graph-based word spotting methods rely on an approximation. Popular examples in this regard are bipartite matching (BP) [19], also known as assignment edit distance (AED) [24], or Hausdorff edit distance (HED) [7]. As deep learning and neural networks have drastically increased word spotting performances in the image domain, this approach was just recently investigated for graph representations. The Geometric Deep Learning [3] framework allows to build neural networks similar to CNNs that operate on graphs. In [17], the authors propose a graph neural network that learns an enriched graph representation with a siamese approach. Based on the enriched representation, a graph distance similar to the HED is defined, resulting in a fast and efficient similarity measure. This method is extended in [18] to a triplet approach achieving competitive results on multiple graph-based word spotting benchmarks.

## 2.3 Geometric Deep Learning

Neural networks for graph representations were first proposed in [23]. Motivated by the success in the image domain, generalizing the convolutional operation has been of significant interest [34]. As a general taxonomy, *spectral* and *non-spectral* methods are distinguished. In contrast to *spectral* approaches, which are motivated by the formulation of a graph signal, *non-spectral* approaches are defined for the entire graph representation and usually work on spatially close neighbourhoods [34]. Most graph neural networks share a common structure that can be summarized in the general framework of a *Message Passing Neural Network* [8]. Each layer is defined by a message and an update function. The message function aggregates information from neighbouring nodes, while the update function computes a node embedding based on the aggregated representations. Finally, a readout function is defined, which computes a feature vector for the entire graph. If all three functions are differentiable, the resulting model may be trained in a supervised manner. For an extensive review of graph neural networks, see [3,34].

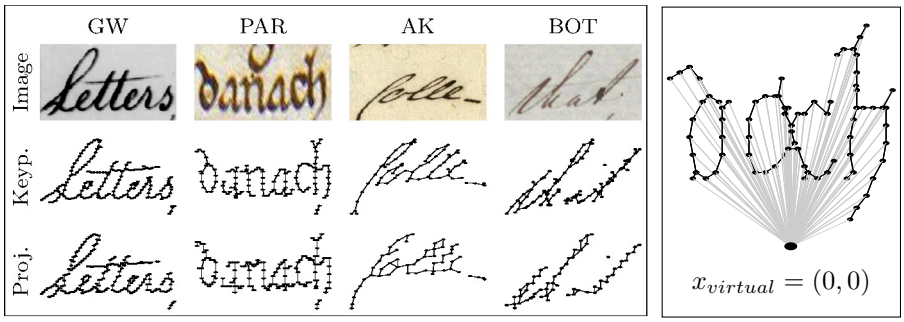



**Fig. 1:** (Left) Examples of keypoint (Keyp.) and projection (Proj.) graphs for George Washington (GW), Parzival (PAR), Alvermann Konzillsprotokolle (AK) and Botany (Bot) [26]. (Right) Representation enriched with virtual node.

# 3 Method

In the following section, we discuss the proposed graph convolutional network for graph-based word spotting. Following the approach of [28], we aim at predicting an attribute representation from a handwritten word graph. Predicting attributes from a graph is similar to the problem of graph property prediction, which is a popular task tackled with graph convolutional networks [11].

## 3.1 Graph Representations

In this work, a handwritten word is represented as a set of nodes $V$ and undirected edges that are expressed by a binary adjacency matrix $\boldsymbol{A}$. Each node has a feature vector $x_v$, which represents its spatial position. Multiple extraction methods exist that extract such graph representations from word images [26]. Here, we focus on graphs extracted by means of identifying *keypoints* or by a segmentation resulting from *projection* profiles. See Fig. 1, for examples of keypoint and projection graphs for four different datasets from [26].

Several works in the literature introduce virtual nodes [8, 11] to allow each node to receive context information from the entire graph. We investigate the use of this enriched graph representation, by introducing an additional node to each word graph. The virtual node is introduced with a zero vector as a feature vector and is connected to every node of the graph. Fig. 1 visualizes this enriched graph representation.

## 3.2 Convolutional Layers

In order to map the node features to a hidden state $h_v$, we use convolutional layers that can be formulated in the message passing neural network framework [8] as follows. A message passing network performs a message passing phase for $T$ time steps that is defined by its message passing function $M_t$ aggregating infor-

mation from the node neighbourhood $\mathcal{N}(v)$. The update function $U_t$ computes a hidden state of the node based on the received message $m_v^{t+1}$:

$$m_v^{t+1} = \sum_{w \in \mathcal{N}(v)} M_t(h_v^t, h_w^t) \qquad h_v^{t+1} = U_t(h_v^t, m_v^{t+1}) \tag{1}$$

**Graph Convolutional Networks (GCN)**

In [12], Kipf and Welling propose a convolutional layer for graphs that is based on an approximation of spectral graph convolutions. Despite its spectral nature, the GCN layer can be interpreted as a spatial method. Due to the simplifications introduced in [12], the resulting convolutional layer aggregates information of a node neighbourhood that is transformed using a layer-specific weight matrix $\boldsymbol{W}$. The resulting message and update functions can be expressed as

$$m_v^{t+1} = \sum_{w \in \mathcal{N}(v)} \frac{1}{\sqrt{deg(v) \cdot deg(w)}} \cdot A_{vw} \cdot h_w^t \tag{2}$$

$$h_v^{t+1} = \boldsymbol{W}^t m_v^{t+1}, \tag{3}$$

with $deg(\cdot)$ denoting the degree of a note. As in [12], we limit the message passing time steps to $T = 1$ and only consider a binary adjacency matrix. Therefore, a multi layer graph convolutional network can be mathematically formulated by the computation of a hidden state $h_v^k$ at layer $k$:

$$h_v^{(k+1)} = \sum_{w \in \mathcal{N}(v) \cup \{v\}} \frac{1}{\sqrt{deg(v) \cdot deg(w)}} \cdot (\boldsymbol{W}^{(k)} \cdot h_v^{(k)}) \tag{4}$$

**Sample and Aggregate (SAGE)**

The spatial approach to generalize the convolution operation to graphs proposed in [10] relies on a sampling and aggregation strategy. First, the neighbourhood of a node is sampled followed by the generation of a neighbourhood embedding by means of an aggregation function. Hereby, the model learns a function on how to aggregate neighbourhood information. In this work, we consider the direct neighbourhood at each layer, resulting in $T = 1$ with respect to the message passing framework. As a simple aggregation function, we use the mean over the neighbourhood node embeddings. This results in a formulation of a convolutional layer that is similar to the GCN approach with the following message and update functions:

$$m_v^{k+1} = \text{mean}_{w \in \mathcal{N}(v)}(h_w^{(k)}) \tag{5}$$

$$h_v^{(k+1)} = \boldsymbol{W_1}^{(k)} h_v^{(k)} + \boldsymbol{W_2}^{(k)} \cdot m_v^{k+1} \tag{6}$$

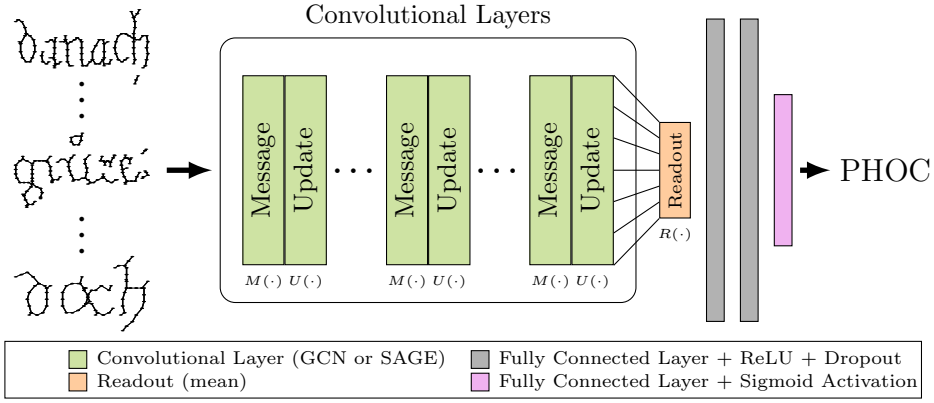



**Fig. 2:** Overview of the proposed graph neural network. Each graph is propagated through $K$ convolutional layers, followed by a readout function to generate a graph embedding. A multilayer perceptron predicts respective PHOC vectors.

### 3.3 Architecture

Fig. 2 presents an overview of the overall architecture. The extracted word graphs are propagated through the network to predict a PHOC representation similar to [28]. The backbone of the architecture is a number of $K$ convolutional layers, as described in section Sec. 3.2. Following the message passing neural network framework, a readout function $R$ generates an embedding $\hat{y}$ for the entire graph. We use the mean over all node embeddings as a readout function:

$$\hat{y} = R(\{h_v^K | v \in V\}) = \frac{1}{|V|} \sum_{v \in V} h_v^K \tag{7}$$

In analogy to the attribute CNN approach in the image domain [28], we use a multilayer perceptron with sigmoid activations to predict a PHOC representation from the learned graph embedding. The model is then optimized fully supervised in an end-to-end manner.

## 4 Experiments

In our experiments, we investigate the proposed model on datasets for graph-based word spotting (Sec. 4.1). We focus on the influence of increasing number of layers for GCN and SAGE convolutions and the introduction of a virtual node. Finally, we compare our model to other graph-based methods, as well as methods for *segmentation-based* word spotting in the image domain (Sec. 4.3).

In all experiments, we use a PHOC vector with splits $[2, 3, 4, 5]$ and the cosine similarity to measure similarity between the representations in the attribute space. The size of the node embedding is set to 256 and both fully connected

**Table 1:** Number of samples for different dataset splits.

| Split | GW | PAR | AK | BOT |
|---|---|---|---|---|
| Train | 2447 | 11468 | 1849 | 1684 |
| Validation | 1224 | 4621 | 3734 | 3380 |
| Test | 1224 | 6869 | - | - |
| Keywords | 105 | 1217 | 200 | 150 |

layers consist of 1024 neurons. All models are trained with ADAM optimization, binary cross entropy loss, a learning rate of 0.001 and a batch size of 64. We train our model for 500 epochs on the designated training splits of the datasets. Performance is measured with mean average precision (mAP) [9].

Due to the large number of different works that have been published on the topic of word spotting, several evaluation protocols exist, despite most works focus on the same datasets. We follow the nomenclature proposed in [18], distinguishing the two ways of an *individual* and *combined* query representation. In this regard, individual means that a query is represented by a single graph. This representation is used in most image-based protocols where queries are usually represented by a single exemplar image. As most graph-based methods measure the structural similarity between graphs, retrieval performance is quite sensitive with respect to writing style variations. In the *combined* query protocol, this problem is mitigated by combining multiple graphs of the same keyword to represent a single query. The resulting similarity measure is then based on the most similar keyword graph. For our attribute model, this corresponds to the minimum over all distances between the estimated PHOC vector of a word graph and all query graphs corresponding to a single keyword. In case of *query-by-string*, each keyword is used as query once.

## 4.1 Datasets

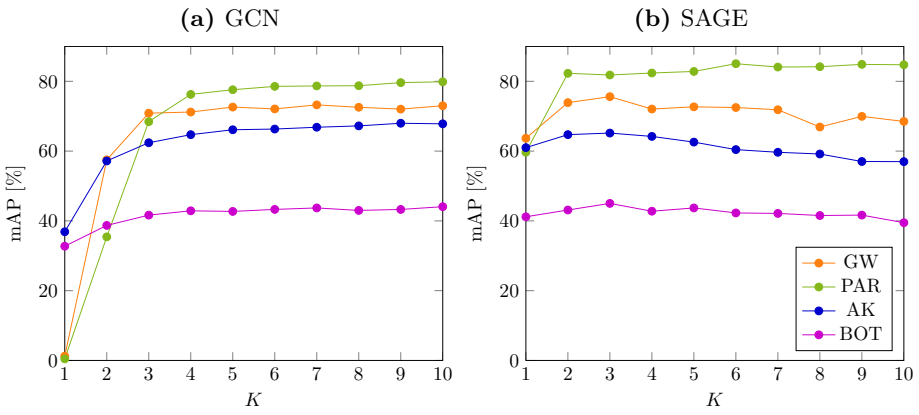
In our experimental evaluation, we rely on the Histograph database [24,26]. The database provides graph representations for four popular manuscripts known from *segmentation-based* word spotting in the image domain. The authors provide multiple datasets that result from different graph extraction strategies, with keypoint and projection graphs being the most popular. See Tab. 1 for an overview of the datasets and corresponding numbers of samples.

The *George Washington* (GW) dataset has traditionally been a key benchmark dataset in the word spotting community. The historic dataset shows almost no degradation and variations in handwriting style are limited.

The *Parzival* (PAR) dataset is a historic dataset written in German. 45 pages are available and show degradations. Despite the fact that is was written by three authors, variations in writing style are comparable low.

The *Alvermann Konzilsprotokolle* (AK) dataset emerged from the keyword spotting competition at the International Conference of Frontiers in Handwriting

**Fig. 3:** *Query-by-example* performances for different numbers of convolutional layers $K$. Results reported as mAP [%].




Recognition 2016 [15]. The corresponding competition had a focus on evaluating the influence of different amounts of training data. The histograph database provides graph representation for period $I$ of the competition which contains only the smallest number of training samples.

*Botany* (BOT) stems from the same competition as the AK dataset [15]. The manuscript shows significant marks of degradation such as fading. Furthermore, the rather artistic style of the botanical records lead to significant variations in writing style and also in scale.

## 4.2 Results

In order to investigate the depth of the model and the introduction of a virtual node, we evaluate performances in terms of mean average precision in the *query-by-example* scenario on the keypoint graph representations. We follow the combined query protocol in all experiments. As training data is highly limited in all cases except PAR, we include also validation data for GW during training. This is similar to the evaluation protocol in the image domain, where usually no designated validation split is used for GW [1]. Note that all evaluated models are also capable to perform *query-by-string* as the model estimates PHOC representations. Quantitative results for *query-by-string* are provided in Sec. 4.3.

### Depth

In the application of convolutional neural networks, we observe a trend towards increasingly deep architectures. While it seems that depth often leads to superior performances in the image domain, most graph neural networks are quite shallow [12, 18, 34]. In this set of experiments, we vary the number of layers from one to ten for GCN and SAGE convolutions, as introduced in Sec. 3.

**Table 2:** Evaluation of Virtual Node (VN). Results reported as mAP [%].

| Convolution | Layer | VN | GW QbE | PAR QbE | AK QbE | BOT QbE |
|---|---|---|---|---|---|---|
| SAGE | 3 | No | 75.60 | 81.81 | 65.17 | **45.01** |
| SAGE | 3 | Yes | **75.84** | 86.59 | 63.62 | 44.69 |
| SAGE | 6 | No | 72.49 | 85.05 | 58.24 | 42.28 |
| SAGE | 6 | Yes | 73.63 | **88.93** | 60.43 | 41.78 |
| GCN | 10 | No | 73.00 | 79.87 | **67.82** | 44.07 |
| GCN | 10 | Yes | 71.18 | 81.55 | 63.52 | 41.79 |

Fig. 3a shows performances for GCN layer. We are able to observe performance improvements for increasing depth for all datasets. This is especially interesting as often no performance gains are reported in the literature for models with more than three layers [12, 18]. While we observe minor performance gains for increasing numbers of GCN layers, this is not the case for SAGE layers, see Fig. 3b. Only for PAR, using more than three convolutional layers improves performances and we do not observe any further gains beyond six layers. For all other datasets where considerably less training data is available, performances degrade after three layers. These results indicate that the availability of training data determines in how far the model can benefit from the increased complexity. In general, the models based on SAGE convolutions yield higher performances for GW and PAR despite using fewer layers. For AK and BOT the deep GCN models with ten layers, show slightly higher performances.

**Virtual Node**

Introducing a virtual node to a graph, allows message flow between all nodes. Motivated by the previously discussed analysis on increasing depth, we investigate the influence on a GCN model with ten layers and two SAGE models with three and six convolutional layers. Tab. 2 presents *query-by-example* performances on all datasets. For GW, we observe only a limited influence on the SAGE models of different depth, while performances decrease in case of the GCN model. In contrast to GW, we observe some clear performance gains for all models in case of PAR. For AK and Botany the results are not conclusive. It seems that the GCN models do not benefit from the introduction of a virtual node. However, when it is feasible to train an accurate SAGE model, a virtual node fosters performances. As in the case of GW and PAR, the highest performances are reported for SAGE models including virtual nodes.

## 4.3 Comparison to the literature

In this section, we compare our model to other graph and image-based word spotting methods. To allow for a fair comparison, we only train our model on

**Table 3:** *Query-by-example* performances based on keypoint (Keyp.) and projection (Proj.) graphs of the Histograph DB. All method follow the indiviual query protocol. Results reported as mAP [%].

| | Method | GW QbE | PAR QbE | AK QbE | BOT QbE |
|---|---|---|---|---|---|
| Keyp. | Ours | **77.99** | **93.98** | 55.37 | 34.21 |
| | Riba et al. [18] | 76.92 | 73.14 | **62.90** | **41.52** |
| Proj. | Ours | **77.82** | **95.48** | 59.92 | 33.31 |
| | Riba et al. [18] | 70.25 | 75.19 | **65.04** | **42.83** |

the designated training splits, if not further noted. In this work, we focus on *segmentation-based* methods, as the graph extraction methods that underly the Histograph database requires an independent segmentation step.

### Graph Domain

A first comparison can be drawn between the proposed model and [18]. Tab. 3 presents *query-by-example* performances for keypoint and projection graphs following the individual query protocol. Our proposed attribute-based approach compares well on GW and PAR and improves performances in all cases. Performance gains are especially striking in case of PAR where a large training set is available. In [18], the authors propose a graph convolutional network trained with triplets in order to learn a graph distance. This approach only considers word class information during training, as opposed to attribute learning that relies on transcriptions. Exploiting the richer annotation offers an explanation for the observed performance gains in cases where a sufficiently large training set is available. For the smaller datasets of AK and BOT, the metric learning approach presented in [18] seems to be beneficial.

Tab. 4 compares our model to other graph-based methods from the literature under the combined query protocol. Except for [18], all other method are *learning-free*. Additionally, we report numbers for an extended training set, where we included the validation data during training for GW and PAR. The proposed attribute approach outperforms all *learning-free* methods, given enough training data as in case of PAR or the extended GW dataset. This emphasizes that the proposed method scales fairly well with the availability of labeled data. Furthermore, we see that end-to-end learning is feasible in the graph domain and considerable performance gains can be achieved. When data is highly limited as in the case of AK or BOT similarity-based approaches seem to be advantageous.

Another interesting observation can be made, comparing the results of the individual (Tab. 3) and combined (Tab. 4) protocol. While [18] reports higher performances under the combined protocol, this is not the case for our model on GW and PAR. As the PHOC vector is independent from the structural characteristics of the query graphs, our model does not benefit from combining multiple

**Table 4:** *Query-by-example* and *string* performances based on keypoint and projection graphs of the Histograph DB. All methods from the literature follow the combined query protocol. Results reported as mAP [%].

| | Method | GW | | PAR | | AK | | BOT | |
|---|---|---|---|---|---|---|---|---|---|
| | | QbE | QbS | QbE | QbS | QbE | QbS | QbE | QbS |
| Keypoint | Ours | 66.73 | 66.57 | 89.03 | 88.80 | 67.82 | 38.68 | 45.01 | 8.48 |
| | Ours† | 66.71 | 66.57 | 88.93 | 88.80 | 59.14 | **38.68** | 36.61 | **8.48** |
| | Ours* | 75.84 | **75.74** | **90.61** | **90.66** | - | - | - | - |
| | Riba et al. [18] | **78.48** | - | 79.29 | - | 78.64 | - | **51.90** | - |
| | AED [2] | 68.42 | - | 55.03 | - | 77.24 | - | 50.94 | - |
| | HED [2] | 69.28 | - | 69.23 | - | **79.72** | - | 51.74 | - |
| Projection | Ours | 68.28 | 68.20 | 90.55 | 90.51 | 71.98 | 39.98 | 44.43 | 7.77 |
| | Ours† | 68.17 | 68.20 | 90.59 | 90.51 | 63.25 | **39.98** | 37.15 | **7.77** |
| | Ours* | **73.61** | **73.21** | **92.77** | **92.80** | - | - | - | - |
| | Riba et al. [18] | 73.03 | - | 79.95 | - | 79.55 | - | **52.83** | - |
| | AED [2] | 60.83 | - | 63.35 | - | 76.02 | - | 50.49 | - |
| | HED [2] | 66.71 | - | 72.82 | - | **81.06** | - | 51.69 | - |
| | Ensemble [26] | 70.56 | - | 79.38 | - | 84.77 | - | 68.88 | - |

(∗) extended training data (†) no query combination

query graphs, if an accurate PHOC estimation is possible. In order to show this characteristics, we report results under a changed combined protocol in row two of Tab. 4. Instead of taking the minimal distance to all query instances, we do not combine queries, but average over the average precisions for each keyword. This accounts for the different query counts per keyword under the individual protocol. It can be concluded that the performance loss of our model is a result of the change of query distributions. On the other hand, our model does not benefit from combing query instances in case of GW and PAR. This result underlines the potential power of the proposed model, as performances are expected to strongly decrease for the *learning-free* methods without query combination. As gathering multiple instances of a keyword is a high demand, we advocate to report results under an individual protocol.

While *query-by-string* is out of scope for all other graph-based methods, it is easily possible with the proposed attribute approach. Our model is capable of mapping graphs to an attribute space. This representation is more powerful than a simple numeric vector embedding, as it encodes symbolic information. In case of AK and BOT *query-by-string* performance is comparable poor, illustrating that the model is not capable to learn the desired character models.

### Image Domain

Tab. 5 compares our method to image-based word spotting systems from the literature and illustrates the existing performance gap between structural and statistical approaches. An interesting observation can be made with respect to

**Table 5:** Image and graph-based *query-by-example* performances. All method follow an individual query protocol. Results reported as mAP [%].

| | Method | GW | PAR | AK | | | BOT | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | I | II | III | I | II | III |
| Graph | Ours | 77.99 | **95.48** | 59.92 | - | - | 34.21 | - | - |
| | Riba et al. [18] | 76.92 | 79.95 | 64.42 | - | - | 41.52 | - | - |
| Image | TPP-PHOCNet [28] | **97.98** | - | **86.01** | **97.05** | **98.11** | 47.75 | **83.51** | **96.05** |
| | CNN & HMM [31] | 85.06 | 94.57 | - | - | - | - | - | - |
| | CVCDAG [15] | - | - | 77.91 | - | - | **75.77** | - | - |
| | TAU [15] | - | - | 71.11 | - | - | 50.64 | - | - |
| | QTOB [15] | - | - | 82.15 | - | - | 54.95 | - | - |

the TPP-PHOCNet, which follows an attribute learning approach. While the PHOCNet clearly outperforms the other models that reported results in the competition based on the highest number of training samples, the performance gain is smaller in case of the smallest training set. Especially in case of BOT, the performance of the attribute-based approach degrades strongly, indicating the high complexity of the attribute prediction task and its sensitivity to training data. A similar observation can be made with respect to to the proposed graph convolutional neural network, which performs comparable poor in these cases.

Overall, the attribute-based approach improves performances given enough training data in the graph domain and contributes to closing the performance gap between the graph and image domain. In case of PAR, we achieve comparable performances to the image domain motivating the further exploration of *learning-based* approaches for structural pattern recognition.

# 5 Conclusions

In this work, we propose a graph convolutional neural network for predicting attribute representations from handwritten word graphs. By mapping a graph to an attribute vector space, the word spotting problem can be solved with the help of a simple vector distance. We are able to show that a fully supervised learning approach is feasible in the graph domain and achieves considerable performance gains when sufficient training data is available. As performance depends on the availability of labeled samples, the exploration of techniques such as synthetic data, semi-supervised or transfer learning is a future line of research. In these limited data cases, methods potentially increasingly benefit from the high representational power of graphs. Our work constitutes a step towards bridging the performance gap between structural and statistical pattern recognition approaches for word spotting. A *learning-based* unification of both paradigms offers the potential to combine the representational power of graphs with the benefits from statistical approaches.

**Acknowledgement.** This work has been supported by the Swiss Hasler Foundation (project 20008).

# References

1. Almazán, J., Gordo, A., Fornés, A., Valveny, E.: Word spotting and recognition with embedded attributes. TPAMI **36**(12), 2552–2566 (2014). https://doi.org/10.1109/TPAMI.2014.2339814
2. Ameri, M.R., Stauffer, M., Riesen, K., Bui, T.D., Fischer, A.: Graph-based keyword spotting in historical manuscripts using Hausdorff edit distance. Pattern Recognition Letters **121**, 61–67 (2019). https://doi.org/10.1016/j.patrec.2018.05.003
3. Bronstein, M.M., Bruna, J., LeCun, Y., Szlam, A., Vandergheynst, P.: Geometric deep learning: Going beyond Euclidean data. IEEE Signal Processing Magazine **34**(4), 18–42 (2017). https://doi.org/10.1109/MSP.2017.2693418
4. Bunke, H., Riesen, K.: Towards the unification of structural and statistical pattern recognition. Pattern Recognition Letters **33**(7), 811–825 (2012). https://doi.org/10.1016/j.patrec.2011.04.017
5. Fischer, A., Keller, A., Frinken, V., Bunke, H.: Lexicon-free handwritten word spotting using character HMMs. Pattern Recognition Letters **33**(7), 934–942 (2012). https://doi.org/10.1016/j.patrec.2011.09.009
6. Fischer, A., Riesen, K., Bunke, H.: Graph similarity features for HMM-based handwriting recognition in historical documents. In: ICFHR. pp. 253–258. Kolkata, India (2010). https://doi.org/10.1109/ICFHR.2010.47
7. Fischer, A., Suen, C.Y., Frinken, V., Riesen, K., Bunke, H.: Approximation of graph edit distance based on Hausdorff matching. Pattern Recognition **48**(2), 331–343 (2015). https://doi.org/10.1016/j.patcog.2014.07.015
8. Gilmer, J., Schoenholz, S.S., Vinyals, O., Dahl, G.E.: Neural message passing for quantum chemistry. In: ICML. vol. 70, pp. 1263–1272. Sydney, Australia (2017)
9. Giotis, A.P., Sfikas, G., Gatos, B., Nikou, C.: A survey of document image word spotting techniques. Pattern Recognition **68**, 310–332 (2017). https://doi.org/10.1016/j.patcog.2017.02.023
10. Hamilton, W.L., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: NIPS. pp. 1024–1034. Long Beach, CA, USA (2017)
11. Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., Leskovec, J.: Open graph benchmark: Datasets for machine learning on graphs. In: NIPS (2020)
12. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: ICLR. Toulon, France (2017)
13. Krishnan, P., Jawahar, C.V.: HWNet v2: An efficient word image representation for handwritten documents. IJDAR **22**(4), 387–405 (2019). https://doi.org/10.1007/s10032-019-00336-x
14. Lang, E., Puigcerver, J., Toselli, A.H., Vidal, E.: Probabilistic indexing and search for information extraction on handwritten german parish records. In: ICFHR. pp. 44–49. Niagara Falls, NY, USA (2018). https://doi.org/10.1109/ICFHR-2018.2018.00017
15. Pratikakis, I., Zagoris, K., Gatos, B., Puigcerver, J., Toselli, A.H., Vidal, E.: ICFHR2016 handwritten keyword spotting competition (H-KWS 2016). In: ICFHR. pp. 613–618. Shenzhen, China (2016). https://doi.org/10.1109/ICFHR.2016.0117

16. Retsinas, G., Louloudis, G., Stamatopoulos, N., Gatos, B.: Efficient learning-free keyword spotting. TPAMI **41**(7), 1587–1600 (2019). https://doi.org/10.1109/TPAMI.2018.2845880
17. Riba, P., Fischer, A., Lladós, J., Fornés, A.: Learning graph distances with message passing neural networks. In: ICPR. pp. 2239–2244. Beijing, China (2018). https://doi.org/10.1109/ICPR.2018.8545310
18. Riba, P., Fischer, A., Lladós, J., Fornés, A.: Learning graph edit distance by graph neural networks. CoRR **abs/2008.07641** (2020), arXiv preprint
19. Riesen, K., Bunke, H.: Approximate graph edit distance computation by means of bipartite graph matching. Image and Vision Computing **27**(7), 950–959 (2009). https://doi.org/10.1016/j.imavis.2008.04.004
20. Rothacker, L., Wolf, F., Fink, G.A.: Annotation-free word spotting with bag-of-features HMMs. IJPRAI p. 2153001 (2020)
21. Rusiñol, M., Aldavert, D., Toledo, R., Lladós, J.: Efficient segmentation-free keyword spotting in historical document collections. Pattern Recognition **48**(2), 545 – 555 (2015). https://doi.org/10.1016/j.patcog.2014.08.021
22. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M.S., Berg, A.C., Li, F.: ImageNet large scale visual recognition challenge. Int. J. of Computer Vision **115**(3), 211–252 (2015). https://doi.org/10.1007/s11263-015-0816-y
23. Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G.: The graph neural network model. IEEE Trans. Neural Networks **20**(1), 61–80 (2009). https://doi.org/10.1109/TNN.2008.2005605
24. Stauffer, M., Fischer, A., Riesen, K.: A novel graph database for handwritten word images. In: S+SSPR. pp. 553–563. Mérida, Mexico (2016). https://doi.org/10.1007/978-3-319-49055-7_49
25. Stauffer, M., Fischer, A., Riesen, K.: Graph-based keyword spotting in historical documents using context-aware hausdorff edit distance. In: DAS. pp. 49–54. Vienna, Austria (2018). https://doi.org/10.1109/DAS.2018.31
26. Stauffer, M., Fischer, A., Riesen, K.: Keyword spotting in historical handwritten documents based on graph matching. Pattern Recognition **81**, 240–253 (2018). https://doi.org/10.1016/j.patcog.2018.04.001
27. Sudholt, S., Fink, G.A.: PHOCNet: A deep convolutional neural network for word spotting in handwritten documents. In: ICFHR. pp. 277–282. Shenzhen, China (2016). https://doi.org/10.1109/ICFHR.2016.0060
28. Sudholt, S., Fink, G.A.: Attribute CNNs for word spotting in handwritten documents. IJDAR **21**(3), 199–218 (2018). https://doi.org/10.1007/s10032-018-0295-0
29. Vats, E., Hast, A., Fornés, A.: Training-free and segmentation-free word spotting using feature matching and query expansion. In: ICDAR. pp. 1294–1299. Sydney, NSW, Australia (2019). https://doi.org/10.1109/ICDAR.2019.00209
30. Wang, P., Eglin, V., Garcia, C., Largeron, C., Lladós, J., Fornés, A.: A novel learning-free word spotting approach based on graph representation. In: DAS. pp. 207–211. Tours, France (2014). https://doi.org/10.1109/DAS.2014.46
31. Wicht, B., Fischer, A., Hennebert, J.: Deep learning features for handwritten keyword spotting. In: ICPR. pp. 3434–3439. Cancún, Mexico (2016). https://doi.org/10.1109/ICPR.2016.7900165
32. Wilkinson, T., Lindström, J., Brun, A.: Neural Ctrl-F: Segmentation-free query-by-string word spotting in handwritten manuscript collections. In: ICCV. pp. 4443–4452. Venice, Italy (2017). https://doi.org/10.1109/ICCV.2017.475

33. Wolf, F., Fink, G.A.: Annotation-free learning of deep representations for word spotting using synthetic data and self labeling. In: DAS. pp. 293–308. Wuhan, China (2020). https://doi.org/10.1007/978-3-030-57058-3_21
34. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Yu, P.S.: A comprehensive survey on graph neural networks. IEEE Trans. Neural Networks Learning Systems **32**(1), 4–24 (2021). https://doi.org/10.1109/TNNLS.2020.2978386