On-the-fly Deformations for Keyword Spotting

George Retsinas[1], Giorgos Sfikas[2,3,4], Basilis Gatos[3], and Christophoros Nikou[2]

[1] School of Electrical and Computer Engineering
National Technical University of Athens, Greece
[2] Computational Intelligence Laboratory
Institute of Informatics and Telecommunications, National Center for Scientific Research "Demokritos", Greece
[3] Department of Computer Science and Engineering
University of Ioannina, Greece
[4] Department of Surveying and Geoinformatics Engineering
University of West Attica, Greece
gsfikas@uniwa.gr, gretsinas@central.ntua.gr, cnikou@cse.uoi.gr, bgat@iit.demokritos.gr

Abstract. Modern Keyword Spotting systems rely on deep learning approaches to build effective neural networks which provide state-of-the-art results. Despite their evident success, these deep models have proven to be sensitive with respect to the input images; a small deformation, almost indistinguishable to the human eye, may considerably alter the resulting retrieval list. To address this issue, we propose a novel "on-the-fly" approach which deforms an input image to better match the query image, aiming to stabilize the aforementioned sensitivity. Results on the IAM dataset verify the effectiveness of the proposed method, which outperforms existing Query-by-Example approaches. Code is available at https://github.com/georgeretsi/defKWS/.

Keywords: Keyword Spotting, Query by Example, PHOC descriptor, Image Deformations, Image Warping, Deep Learning

1 Introduction

Distance measures are critical in any retrieval paradigm. Keyword Spotting (KWS), defined as the problem of retrieving relevant word instances over a digitized document, is no exception to this rule. A typical KWS pipeline consists of i) a way to produce compact word image descriptions ii) a distance measure that is used to compare query words against the target content, iii) returning to the user the words that are the closest to the query, in terms of the distance measure considered. We can consider this processing pipeline in terms of the characteristics of the space where the produced image descriptions reside. The choice of this space is interrelated with the choice of certain topological properties; the selected distance measure induces a specific topology for the considered space, and it is in this regard that it is important in the current application context. Consequently, it is perhaps more useful to consider the problem of defining a descriptor *jointly* with the problem of defining a useful distance measure.

Depending on the input format of the query, we can distinguish two scenarios: Query by Example (QbE) and Query by String (QbS). In the QbE scenario the query is given in the form of an image, while in the QbS scenario the query corresponds to a text string. We consider only the QbE KWS scenario.

In this work, we propose focusing on determining a distance measure to accurately perform KWS. We cast the KWS problem as follows: Given a query image $\mathbf{I_q}$ we aim to find a valid spatial transformation $\mathcal{T}$ over an input image $\mathbf{I_w}$, such that the two images are as *visually close* as possible.

Implementation-wise, the distance of the images is to be minimized in the feature domain, by extracting feature vectors for both images using a deep neural network. The transformations are implemented via a deformation set of parameters $\mathbf{d}$ in an iterative manner. Overall, we seek appropriate parameters $\mathbf{d}$, which minimize the distance between the two images using a gradient-based optimization algorithm. This concept is depicted in fig. 1, where the deformed image is generated by the proposed algorithm for a large number of iterations (100). In a sense, we seek a "least resistance" transformation path, where transformations do not disrupt the content of the image. In terms of the manifold hypothesis, this path can be understood as a geodesic connecting the two images along the data manifold; the required distance is defined to be the length of this path.

Fig. 1. Visualization of the paper motivation: transform an input image in such a manner that it becomes sufficiently similar to a given target image. The deformed version of the image was created by the proposed algorithm over an impractically large number of iterations (100).

Even though the core idea of computing a deformation may be attractive for improving performance (it is in this sense that we refer to it as "on-the-fly"), it comes at the cost of computational effort. An iterative process should be performed for each query-word pair. As the experimental section suggests, even when using a very small number of iterations (e.g., 3) we can achieve a considerable boost in performance. This is attributed to the sensitivity of modern

deep learning systems, where small deformations, almost indistinguishable to the human eye, can notably alter the retrieval list of a query.

Furthermore, as a minor contribution, we propose a simple yet effective deep model based on residual blocks which is used to estimate Pyramidal-Histogram-of-Characters (PHOC) [1] representations. Using this fairly compact network of ~ 8 million parameters, we achieve performance in the ballpark of the current state-of-the-art methods. Using this model, along with the proposed on-the-fly deformation scheme we outperform existing methods for the case of QbE keyword spotting scenario for the challenging IAM dataset.

The remainder of this paper is structured as follows. In Section 2 we briefly review related work. In Section 3 we present the architecture of a reference KWS system that will serve as baseline as well as a backbone of the proposed "on-the-fly deformation" approach, which is presented in Section 4. Experiments are presented in Section 5. We close the paper with a short discussion of our conclusion and plans for future work in Section 6.

2 Related Work

Recent developments in computer vision and KWS in particular have redirected the interest of the community from learning-free feature extraction approaches [4, 15] to learning-based / deep learning approaches [12, 16, 25]. A stepping stone towards this shift was the the seminal work of Almazán et al. [1], which first introduced the Pyramidal-Histogram-of-Characters (PHOC) representation. The main idea was to embed both word images and text strings into this common subspace of PHOC embedding, allowing both QbE and QbS scenarios. Since this method was introduced before the surge of deep learning, it was trained with Support Vector Machines (SVMs) but later inspired a large number of deep learning methods to tackle the KWS problem using the attribute-based rationale of PHOC embeddings. A characteristic example is the introduction of convolutional networks as PHOC estimators [9, 12, 21, 25, 27]. Apart from this direction of attribute-based approaches, Wilkinson et al. [28] used a triplet CNN, accepting pairs of positive word matches together with a negative word match, Retsinas et al. [20] extracted word image features from a Seq2Seq recognition system, while Krishnan et al. [13] adopted a hybrid approach which includes PHOC and one-hot word representations, along with a semantic-driven normalized word embedding, aiming to extract an effective joint image-text feature space.

Ideas related to shape deformation and deformation-induced distances have been proposed and used in different contexts. Rigid and non-rigid transformations (or "warps") are important in image and volume registration [14]. Other examples of context include proposing a distance between shapes and using it to learn a manifold [3, 24], using non-rigid matching between shape and template to create a shape description [2], or using a deformation-based distance to cluster a dataset [23]. Geometric deformations have also garnered much interest in the context of deformable convolutions and spatial transformers [19].

Notably, the proposed work "flirts" with the idea of constructing an implicit manifold path, consisted of valid deformation steps between word representations. Despite the fact that manifold-based approaches have been successfully deployed for QbE KWS along with traditional feature extraction methods [22,26], we are not aware of recent deep learning approaches towards this direction.

3 Reference KWS System

First, we will describe the reference KWS system that we use, which will serve both as baseline and as the backbone of the proposed on-the-fly deformation approach. In what follows, we describe the preprocessing steps, the proposed architecture and the training process.

3.1 Preprocessing

The pre-processing steps, applied to every word image, are: 1. All images are resized to a resolution of 64×256 pixels. Initial images are padded (using the image median value, usually zero) in order to attain the aforementioned fixed size. If the initial image is greater than the predefined size, the image is rescaled. The padding option aims to preserve the existing aspect ratio of the text images. 2. A simple global affine augmentation is applied at every image, considering only rotation and skew of a small magnitude in order to generate valid images.

3.2 Proposed Architecture

The overall architecture of the proposed PHOC estimator is described in fig. 2. We distinguish 4 components:

1. **Convolutional Backbone:** The convolutional backbone is made up of standard convolutional layers and ResNet blocks [6], interspersed with max-pooling and dropout layers. In particular, we have a 7×7 convolution with 32 output channels, a series of 2 3×3 ResNet blocks with 64 output channels, 4 3×3 ResNet blocks with 128 output channels and 4 3×3 ResNet blocks with 256 output channels. The standard convolution and the ResNet blocks are all followed by ReLU activations, Batch Normalization and dropout. Between the aforementioned series of ResNet blocks, a 2×2 max pooling of stride 2 is applied in order to spatially downscale the produced feature map. Overall, the convolutional backbone accepts a word image of size 64×256 and outputs a tensor of size $8 \times 32 \times 256$ (the last dimension corresponds to the feature space). The CNN backbone is followed by a *column-wise max pooling* operation, which transforms the output of the CNN (a feature map tensor of size $8 \times 32 \times 256$), into a sequence of feature vectors along the x-axis (size 32×256).

2. **1D Convolutional Part:** The sequence of features, as generated by the CNN backbone and the column-wise max-pooling operation, is then processed by a set of 1D convolutional layers in order to add contextual information to the extracted features. This contextual information is in line with the underlying representation of PHOC, where an attribute corresponds to both the class (character) and its relative position in the images (e.g. the rightmost 'a'). The 1D CNN part consists of three consecutive 1D convolutional layers with kernel size 5, stride 2 and 256 channels. Between the layers, ReLU activations, Batch Normalization and Dropout modules are used. Since the convolutional layers are strided, the output of this part would be a reduced feature sequence of size 4×256. This sequence is then concatenated into a single feature vector of size 1024.
3. **Linear Head:** The final part of the proposed architecture takes as input the concatenated feature vector and produces a PHOC estimation. Pyramidal representations up to 4 levels are used and thus the resulting PHOC embedding has a size of 390. The linear head consists only of two linear layers with a ReLU activation between them and a sigmoid activation on the output (as in [25]).

3.3 Training Process

As in the original PHOCNet paper [25], we train our system using the binary cross-entropy (BCE) loss. Training samples are augmented using a typical affine transformation approach. The training of the proposed system is performed via an Adam [8] optimizer using an initial learning rate of 0.001 which gradually decreases using a multistep scheduler. The overall training epochs are 240 and the scheduler decreases the learning rate by a factor of 0.1 at 120 and 180 epochs. Training samples were fed in batches of 64 images.

3.4 Retrieval Application

For the considered QbE setting, retrieval can be trivially performed by comparing image descriptors. A widely-used distance metric for KWS applications is the cosine distance [16, 18, 27] and thus we also perform comparisons by this metric. Even though initial works on neural net-based PHOC estimators (e.g. [25], [27]) performed comparisons on the PHOC estimation space, as these estimations can be also used straightforwardly for the QbS scenario, it has been proven in practice that features drawn from intermediate layers are more effective [17] in QbE applications. Therefore, we select the concatenated output of the 1D CNN part as the default image descriptor.

4 On-the-fly Deformation

In this section, we will describe in detail the proposed on-the-fly deformation approach, where a word image is spatially transformed in order to be as close as possible to the query image with respect to their corresponding feature vectors.

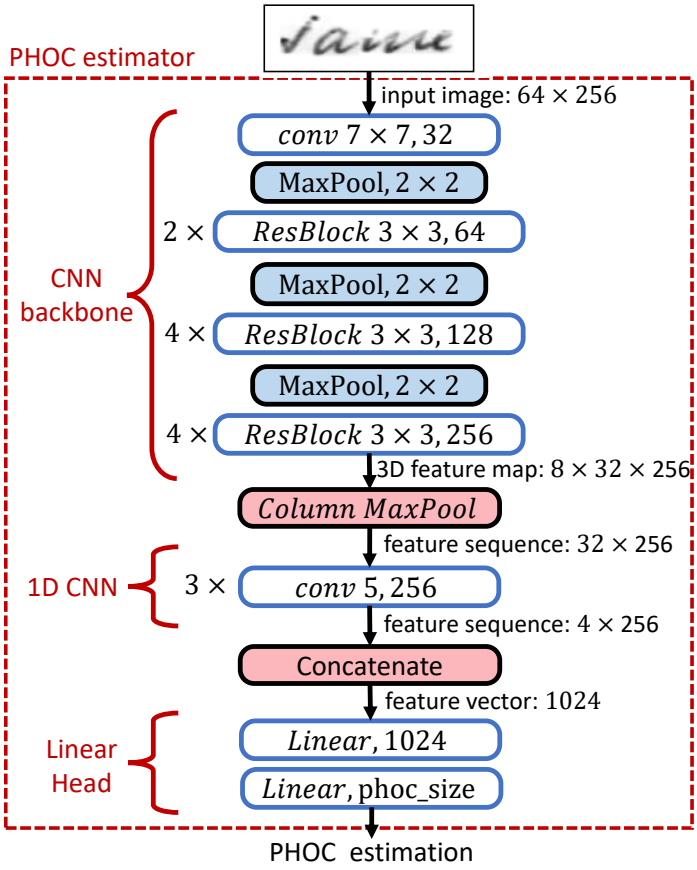

Fig. 2. Overview of the reference architecture consisted of three distinct architecture blocks: CNN backbone, 1D CNN part and Linear Head.

4.1 Considered Deformations

First, we describe the considered deformations which can be applied to the image during the proposed process. Specifically, we consider three categories of increasing deformation "freedom":

1. **Global Affine:** A typical affine transformation is applied on the image, defined by a 2×3 transformation matrix.
2. **Local Affine:** Seeking more refined transformation, we consider a local affine approach, where the image is split to overlapping segments along the x-axis and a typical affine transformation is applied in each segment. Consistency-wise, bilinear interpolation is performed in order to compute the per pixel translation of neighboring segments.

3. **Local Deformation:** Image patches are deformed according to independent x-y pixel translation vectors. Patches of 8×8 are considered and the deformation vectors correspond to the center of these patches.

Examples of aforementioned deformation categories are depicted in fig. 3.

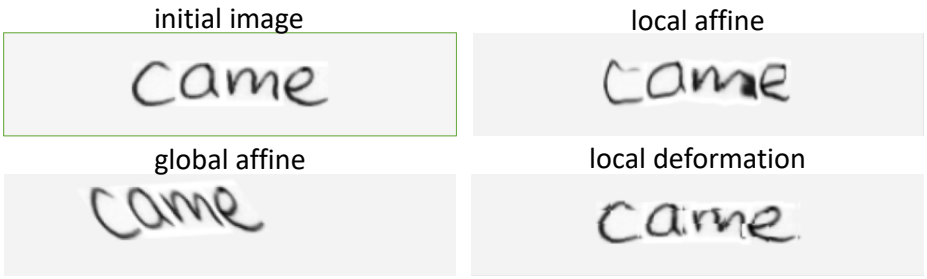

Fig. 3. Visualization of the possible deformation categories. Deformation parameters were selected in order to clearly show the effect of the deformations without significantly distorting the image.

The application of these deformations over the initial image is performed through a grid sampler which allows back-propagation, as defined in Spatial Transformer Networks [7], since the deformation parameters are to be iteratively defined.

Overall, the deformation parameters $\mathbf{d}$ consist of three distinct subsets, $\mathbf{d} = \{\mathbf{d}_{ga}, \mathbf{d}_{la}, \mathbf{d}_{ld}\}$ (ga: global affine, la: local affine and ld: local deformation) which form an interpolation grid. The interpolation grid is applied over the image via a grid sampling technique, generating a transformed image: $\mathbf{I}' = \mathcal{T}(\mathbf{I}; \mathbf{d})$.

4.2 Query-based Deformation

As we have already stated, we aim to find an appropriate transformation of a word image with respect to a target query image. Given a feature extractor f, a bilinear grid sampler $\mathcal{T}$ controlled by deformation parameters d and a pair of input/query images $\mathbf{I_w}$ / $\mathbf{I_q}$, we could opt to maximize the quantity

$$S_C(f(\mathcal{T}(\mathbf{I}_w; \mathbf{d})), f(\mathbf{I}_q)) \tag{1}$$

where S_C denotes the cosine similarity metric. Eq. 1 hints towards the main objective function component, which we proceed to regularize by adding intuitive constraints. First, deformation parameters should be minimized in order to avoid inconsistencies and abrupt transformations ($\|\mathbf{d}\|_2$). As we have already mentioned, we want to create a "least resistance" path of consecutive transformations and thus we do not want to deviate from the solution of the previous step by allowing large changes in deformation parameters. Furthermore, a per-pixel

comparison between the transformed image and the query ($\|\mathcal{T}(\mathbf{I}_w;\mathbf{d}) - \mathbf{I}_q\|_F$) may be helpful to adapt finer details.

Summing up the aforementioned loss components, we finalize our proposal for the objective function as:

$$\begin{aligned}\mathcal{L}(\mathbf{d}) =& 1 - S_C(f(\mathcal{T}(\mathbf{I}_w;\mathbf{d})), f(\mathbf{I}_q)) + \\ & a\|\mathcal{T}(\mathbf{I}_w;\mathbf{d}) - \mathbf{I}_q\|_2 + b\|\mathbf{d}\|_2 \end{aligned} \tag{2}$$

The optimization of eq. 2 is performed via Adam [8]. A sketch of the proposed algorithm is presented in Algorithm 1. Loss hyperparameters are empirically set to $a = 10$ and $b = 1$ by visually observing the resulted transformations. The critical hyper-parameters to be set are the learning rate of Adam and the number of iterations. These hyperparameters are correlated and essentially define the trade-off of improvement vs cost of computation.

Algorithm 1 On-the-fly Deformation

Input: Adam hyperparameters, number of iterations K,
initial deformation $\mathbf{d}_0$, loss hyperparameters a, b
Output: optimized deformation parameters $\mathbf{d}_K$
1: Initialize $\mathbf{d}$ as $\mathbf{d}_0$
2: **for** $i = 0$ to $K - 1$ **do**
3: Forward Pass: Compute $\mathcal{L}(\mathbf{d}_i)$ according to Eq. 2
4: Backward Pass: Compute $\nabla\mathcal{L}(\mathbf{d}_i)$
5: Adam Update: $\mathbf{d}_{i+1}$
6: **end for**

4.3 Implementation Aspects

The extra computational effort introduced by the proposed iterative approach is not trivial. Transforming each and every word image with respect to a specific query is inefficient for large-scale applications. Nonetheless, direct comparison of the initial feature vectors yields notable retrieval performance and thus we expect all the relevant images to be brought up in the first places of the retrieval list. Therefore, instead of applying the proposed method to every word image, we can use only a subset of the N_w most relevant images, as done in [15]. This way, the computationally intense proposed algorithm is performed for only a dozens of words.

5 Experimental Evaluation

Evaluation of the proposed system is performed on the IAM dataset, consisted of handwritten text from 657 different writers and partitioned into writer - independent train / validation / test sets. As IAM is a large and multi-writer dataset, it is very challenging and typically used as the standard benchmark of

comparison for Keyword Spotting methods. The setting under investigation is Query-by-Example (QbE) spotting. As evaluation metrics, we use the standard metrics that are used in the related literature: mean average precision (MAP). Following the majority of KWS works, images in test set with more than one occurrence that do not belong to the official stopword list comprise the query list [1, 25]. Results are reported as average values over 3 separate runs.

5.1 Ablation Study

First, we report basic spotting performance when the proposed KWS system is used. Compared to applying cosine distance on the PHOC estimations which corresponds to 88.78% MAP, using the proposed intermediate feature vectors (i.e., the concatenated output of the 1D CNN part) results to the significantly increased performance of 91.88% MAP.

An interesting aspect of the problem at hand is the sensitivity of the deep learning model to small deformation on the input image. This observation is not restricted to the specific architecture, but pertains to deep neural nets in general. Specifically, it is closely related to the recent field of adversarial examples [5], where a network can be "fooled" and misclassify an input image by adding barely noticeable noise to the image.

This observation would be our motivation for dramatically reducing the number of required steps for the proposed iterative algorithm. If our objective would be to aim at well-performing image deformations of large magnitude, then we should cautiously perform many steps of the proposed algorithm with a small learning rate. Nonetheless, in practice we can greatly simplify the procedure by performing 2-3 steps with a larger learning rate, under the implicit assumption that a solution with considerably improved performance exists in the immediate neighborhood of the image.

To support this idea, we report the per-query difference in spotting performance when applying random transformations from the described categories. The magnitude of the deformation parameters are close to zero, essentially having no visible impact on the image. Specifically, the mean absolute difference in AP for all considered queries is $\sim 1.5\%$, even though the overall MAP lies in the ballpark of the initial performance ($91.88 \pm 0.18\%$). This means that applying visually trivial deformations could generate notable fluctuations in per-query performance. Therefore, we aim to harness this fluctuations and find the proper deformations that can have a positive impact on the overall performance through the proposed algorithm.

Relying on the idea of minor yet impactful deformations, we set the learning rate to 0.01, the default iterations to $K = 3$ and the considered subset to the $N_w = 50$ more relevant images.

In the following ablation experiments, we examine different aspects of the proposed iterative algorithm over the validation set of the IAM (first validation set of the official writer-independent partitions), In Table 1 we report the impact of the different categories of deformations. As we can see, the deformations of higher degree of freedom, i.e. local affine and local deformation, yield

better results. Nonetheless, the best results are reported when all three types of deformation are used together. Therefore, in the upcoming experiments, the combination of the three deformation types is the default approach.

Table 1. Exploring the impact of the different categories of deformations and their combinations.

deformation type	MAP (%)
reference	95.59
gaffine	95.91
laffine	96.22
ldeform	96.19
gaffine + laffine	96.12
gaffine + ldeform	96.14
laffine + ldeform	96.32
gaffine + laffine + ldeform	96.40

Table 2 contains the results for different number N_w of retrieved words. Along the performance metric, we report the required extra time per query. We observe that in all considered cases, there is a noticeable performance increase. Performance is gradually increased up to 50 retrieved words. For 75 retrieved words, no further increase is observed and thus $N_w = 50$ is set as the default value for the rest of the experiments. As expected, the time requirements are almost linearly increased by the number of the retrieved words. In the proposed setting the per-query retrieval time is under 1 second, but this linear correlation hints that applying the proposed algorithm to the full retrieval list of thousand of words would lead to impractical time requirements.

Table 2. Exploring the impact of the N_w (size of considered subset for applying the deformation algorithm) to both performance and time requirements. The IAM validation set is used for evaluation.

N_w	MAP (%)	time (sec/query)
reference	95.59	-
10	96.21	0.14
25	96.33	0.30
50	96.40	0.57
75	96.39	0.83

Finally, we explore the impact of K, i.e. the number of iterations required by the proposed approach. In Table 3, we summarize both the MAP and the required time per query for different values of K. Under the proposed framework, even a single iteration can provide a boost in performance. Nonetheless, (too) many iterations seem to have a negative effect on performance. This can

be attributed to the rather high learning rate (0.01), selected specifically for performing a small number of iterations. Again, the time requirements have a linear dependence to the number of iterations, as expected by the computational complexity of the algorithm. Even though we achieve significant boost by simply following a few steps towards the gradient direction, the case of visually intuitive transformations (cf. fig. 1) still requires a large number of constrained steps. The reported time requirements raise an obstacle towards this direction, since even for only 50 images, several seconds of run-time are required.

Table 3. Exploring the impact of the number of iterations K to both performance and time requirements. The IAM validation set is used for evaluation.

K	MAP (%)	time (sec/query)
reference	95.59	-
1	95.97	0.24
2	96.34	0.40
3	96.40	0.57
4	96.26	0.74
5	96.23	0.91
10	96.11	1.75
15	96.07	2.61
20	95.98	3.45

5.2 Comparison to State-of-the-Art Systems

A comparison of our method versus state-of-the-art methods for KWS is presented in Table 4. Notably, the proposed reference system achieves performance in the ballpark of the state-of-the-art approaches, supporting the effectiveness of the extracted deep features. More importantly, the proposed iterative on-the-fly approach achieves a significant boost of over 1% MAP compared to the reference system, outperforming existing KWS methods for the QbE scenario.

6 Conclusions and Future Work

In this work, we proposed an iterative approach which provides on-the-fly deformations capable to minimize the distance between image descriptions. Considered deformations spatially transform the image according to a minimization loss, following a gradient-based optimization approach. The proposed algorithm is utilized in QbE KWS setting, where our aim is to transform an input word image to be as close as possible to a query image, with respect to an extracted feature space. The feature space in our case is created by a deep neural network, acting as PHOC estimator, while the features are drawn from an intermediate layer. An interesting aspect of this work is the observation that no visually intense deformations are required for achieving considerable boost in performance.

Table 4. Comparison of state-of-the-art KWS approaches for the IAM dataset and the QbE setting.

Method	MAP (%)
PHOCNet [25]	72.51
HWNet [11]	80.61
Triplet-CNN [28]	81.58
PHOCNet-TPP [27]	82.74
DeepEmbed [9]	84.25
Deep Descriptors [18]	84.68
Zoning Ensemble PHOCNet [21]	87.48
End2End Embed [10]	89.07
DeepEmbed [10]	90.38
HWNetV2 [12]	92.41
NormSpot [13]	92.54
Seq2Emb [20]	92.04
Proposed Systems	
reference system	91.88
on-the-fly deformations	93.07

This phenomenon can be attributed to the nature of neural networks, which are susceptible to adversarial attacks, a field that has been studied extensively in the recent years. This observation leads to a cost-effective algorithm, since a few steps of the algorithm are sufficient for attaining increased performance. In fact, the proposed approach achieves the best performance over existing state-of-the-art approaches, at the cost of the required on-the-fly estimation of an appropriate deformation over a reduced set of images.

Still, several interesting research questions need to be addressed as future work: is there an efficient way to generate larger yet "valid" deformations? can we connect this idea to manifold exploration and shortest path approaches through a transformation space? can we apply the deformation protocol only to the query image, as an efficient alternative, in order to be as close as possible to its neighbors?

Acknowledgments

This research has been partially co - financed by the EU and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the calls : "RESEARCH - CREATE - INNOVATE", project *Culdile* (code T1EΔK - 03785) and "OPEN INNOVATION IN CULTURE", project *Bessarion* (T6YBΠ - 00214).

References

1. Almazán, J., Gordo, A., Fornés, A., Valveny, E.: Word spotting and recognition with embedded attributes. IEEE Transactions on Pattern Analysis and Machine Intelligence **36**(12), 2552–2566 (Dec 2014)

2. Cootes, T.F., Twining, C.J., Babalola, K.O., Taylor, C.J.: Diffeomorphic statistical shape models. Image and Vision Computing **26**(3), 326–332 (2008)
3. Gerber, S., Tasdizen, T., Joshi, S., Whitaker, R.: On the manifold structure of the space of brain images. In: International conference on medical image computing and computer-assisted intervention. pp. 305–312. Springer (2009)
4. Giotis, A.P., Sfikas, G., Nikou, C., Gatos, B.: Shape-based word spotting in handwritten document images. In: 13th International conference on document analysis and recognition (ICDAR). pp. 561–565. IEEE (2015)
5. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: Proceedings of the International Conference on Learning Representations (ICLR) (2015)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)
7. Jaderberg, M., Simonyan, K., Zisserman, A., et al.: Spatial transformer networks. Advances in neural information processing systems **28**, 2017–2025 (2015)
8. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Proceedings of the International Conference on Learning Representations (ICLR) (2015)
9. Krishnan, P., Dutta, K., Jawahar, C.V.: Deep feature embedding for accurate recognition and retrieval of handwritten text. In: Proceedings of the 15^{th} International Conference on Frontiers in Handwriting Recognition (ICFHR). pp. 289–294 (2016)
10. Krishnan, P., Dutta, K., Jawahar, C.: Word spotting and recognition using deep embedding. In: 2018 13th IAPR International Workshop on Document Analysis Systems (DAS). pp. 1–6. IEEE (2018)
11. Krishnan, P., Jawahar, C.: Matching handwritten document images. In: European Conference on Computer Vision. pp. 766–782. Springer (2016)
12. Krishnan, P., Jawahar, C.: HWNet v2: An efficient word image representation for handwritten documents. arXiv preprint arXiv:1802.06194 (2018)
13. Krishnan, P., Jawahar, C.: Bringing semantics into word image representation. Pattern Recognition **108** (2020)
14. Noblet, V., Heinrich, C., Heitz, F., Armspach, J.P.: 3-D deformable image registration: a topology preservation scheme based on hierarchical deformation models and interval analysis optimization. IEEE Transactions on image processing **14**(5), 553–566 (2005)
15. Retsinas, G., Louloudis, G., Stamatopoulos, N., Gatos, B.: Efficient learning-free keyword spotting. IEEE Transactions on Pattern Analysis and Machine Intelligence **41**(7), 1587–1600 (2018)
16. Retsinas, G., Louloudis, G., Stamatopoulos, N., Sfikas, G., Gatos, B.: An alternative deep feature approach to line level keyword spotting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12658–12666 (2019)
17. Retsinas, G., Sfikas, G., Gatos, B.: Transferable deep features for keyword spotting. In: International Workshop on Computational Intelligence for Multimedia Understanding (IWCIM), held in conjunction with EUSIPCO (2017)
18. Retsinas, G., Sfikas, G., Louloudis, G., Stamatopoulos, N., Gatos, B.: Compact deep descriptors for keyword spotting. In: 2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR). pp. 315–320. IEEE (2018)
19. Retsinas, G., Sfikas, G., Nikou, C., Maragos, P.: Deformation-invariant networks for handwritten text recognition. In: 2021 IEEE International Conference on Image Processing (ICIP). pp. 949–953. IEEE (2021)

20. Retsinas, G., Sfikas, G., Nikou, C., Maragos, P.: From Seq2Seq recognition to handwritten word embeddings. In: Proceedings of the British Machine Vision Conference (BMVC) (2021)
21. Retsinas, G., Sfikas, G., Stamatopoulos, N., Louloudis, G., Gatos, B.: Exploring critical aspects of cnn-based keyword spotting. a phocnet study. In: 13th IAPR International Workshop on Document Analysis Systems (DAS). pp. 13–18. IEEE (2018)
22. Retsinas, G., Stamatopoulos, N., Louloudis, G., Sfikas, G., Gatos, B.: Nonlinear manifold embedding on keyword spotting using t-sne. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). vol. 1, pp. 487–492. IEEE (2017)
23. Sfikas, G., Heinrich, C., Nikou, C.: Multiple atlas inference and population analysis using spectral clustering. In: 2010 20th International Conference on Pattern Recognition. pp. 2500–2503. IEEE (2010)
24. Sfikas, G., Nikou, C.: Bayesian multiview manifold learning applied to hippocampus shape and clinical score data. In: Medical Computer Vision and Bayesian and Graphical Models for Biomedical Imaging, pp. 160–171. Springer (2016)
25. Sudholt, S., Fink, G.A.: PHOCNet: A deep convolutional neural network for word spotting in handwritten documents. In: Proceedings of the 15^{th} International Conference on Frontiers in Handwriting Recognition (ICFHR). pp. 277–282 (2016)
26. Sudholt, S., Fink, G.A.: A modified isomap approach to manifold learning in word spotting. In: German Conference on Pattern Recognition. pp. 529–539. Springer (2015)
27. Sudholt, S., Fink, G.A.: Evaluating word string embeddings and loss functions for CNN-based word spotting. In: 2017 14th iapr international conference on document analysis and recognition (ICDAR). vol. 1, pp. 493–498. IEEE (2017)
28. Wilkinson, T., Brun, A.: Semantic and verbatim word spotting using deep neural networks. In: 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR). pp. 307–312. IEEE (2016)