Deep Learning for Classification and as Tapped-Feature Generator in Medieval Word-Image Recognition

Chanda, Sukalpa; Okafor, Emmanuel; Hamel, Sebastien; Stutzmann, Dominique; Schomaker, Lambertus

Published in:
13th IAPR International Workshop on Document Analysis Systems (DAS)

DOI:
10.1109/DAS.2018.82

Deep Learning for Classification and as Tapped-Feature Generator in Medieval Word-Image Recognition

Sukalpa Chanda*, Emmanuel Okafor*, Sebastien Hamel†, Dominique Stutzmann† and Lambert Schomaker*
*Institute of Artificial Intelligence & Cognitive Engineering, University of Groningen, The Netherlands
Email:- s.chanda@rug.nl, e.okafor@rug.nl, l.r.b.schomaker@rug.nl
†Institut de recherche et d'histoire des textes, Paris, France
Email:- sebastien.hamel@irht.cnrs.fr, dominique.stutzmann@irht.cnrs.fr

***Abstract*—Historical manuscripts are the main source of information about past. In recent years, digitization of large quantities of historical handwritten documents is in vogue. This trend gives access to a plethora of information about our medieval past. Such digital archives can be more useful if automatic indexing and retrieval of document images can be provided to the end users of a digital library. An automatic transcription of the full digital archive using traditional Optical Character Recognition (OCR) is still not possible with sufficient accuracy. If full transcription is not available, the end users are interested in indexing and retrieving of particular document pages of their interest. Hence recognition of certain keywords from within the corpus will be sufficient to meet the end users needs. Recently, deep-learning based methods have shown competence in image classification problems. However, one bottleneck with deep-learning based techniques is that it requires a huge amount of training samples per class. Since the number of samples per word class is scarce for collections that are freshly scanned, this is a serious hindrance for direct usage of the deep-learning technique for the purpose of word image recognition in historical document images. This paper aims to investigate the problem of recognizing words from historical document images using a deep-learning based framework for feature extraction and classification while countering the problem of the low amount of image samples using off-line data augmentation techniques. Encouraging results (highest accuracy of 90.03%) were obtained while dealing with 365 different word classes.**

I. Introduction

To ensure preservation of historical information, digitization of historical documents is in practice all throughout the world. With the advent of modern and handy image acquisition techniques this has become a widely spread trend, and huge volumes of historical documents encompassing subjects like literature, science, medicine, personal diaries, historical letters of eminent person's etc., are digitized in different corners of the world. Searching for a relevant document/information from a digital archive comprised of such digitized images can be expedited using an intelligent software. However, processing a historical document for OCR or similar other applications is more challenging compared to a contemporary handwritten document. The issues that need to be tackled are mainly as follows: (a) different challenging layout and structure of the historical document image; (b) unwanted noise in the historical document image; (c) artefacts/torn historical document formation due to ageing; (d) handwritten annotations in the text; (e) blurred, broken, faded text regions. (f) the scarcity of training data: Contemporary handwritten characters in roman (western scripts) are quite different in shape from their medieval counterparts. This intensifies the problem of getting enough training data per word-class. For some scripts, the assumption that the characters are individually identifiable is warranted. However, in many script styles, especially cursive connected styles, the individual characters are fused, to the point of being illegible, see Figure 1 for an example. Under such adverse circumstances, a character recognition based transcription result is not reliable. However, searching for a document page with a particular content can be achieved by either word spotting or by word recognition. In word spotting, a query in the form of an image or as a string of text characters is given as an input to the word spotting algorithm which then searches for image regions containing similar words. Word recognition could retrieve a particular document image from a corpus after identifying the word boundaries where the recognizer has recognized the desired word class. Though only words classes from a priorly decided lexicon/dictionary could be used for the purpose of searching all document images. Searching for some selected set of document images

Fig. 1. An example of fused cursive characters in two different words.(Left)"Firmiter".(Right)"Henricus".

within a digital archive can be achieved by four different ways as depicted in Figure 2. As evident from the diagram, retrieval of a word list with hit score/ranks given a query keyword is dependent on either the output of the word spotting algorithm or output from pure handwritten text recognition technique. Word recognition on the other hand relies on a detection algorithm in preprocessing step to determine word image region/boundary of a candidate word and consequently classify the word region to a specific word class.

978-1-5386-3346-5/18 $31.00 © 2018 IEEE
DOI 10.1109/DAS.2018.82

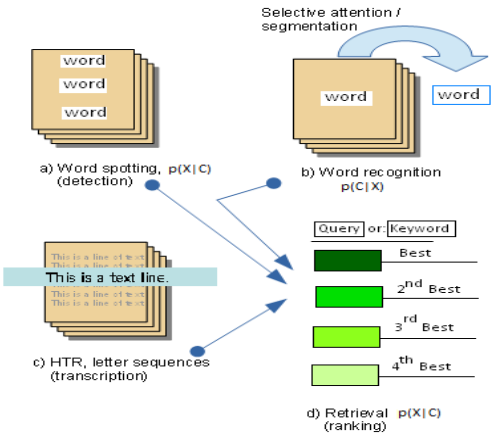

Fig. 2. Four different ways to document image indexing.

Although Convolutional Neural Networks (CNN) have been used for various tasks like image quality enhancement, image retrieval and even writer identification in the recent past, yet there are some serious obstacles in directly using any off-the-shelf CNN algorithm for the historical word image classification problem. They are as follows:(a) scarcity of labelled word images; (b) Using the in-built data augmentation within CNN algorithm might distort the word image in a manner which is not optimal for retaining text information. We used an offline data augmentation technique as stated in [1] to counter such adversities. The objective of the paper is to investigate the efficacy of a Convolutional Neural Network, both as a feature extractor and a classifier in recognizing word images from the medieval handwritten historical document images when a large number of samples per word class is not available. However, as with many CNN studies, we disregard, for the occasion, the essential function of selective attention and word candidate segmentation. This study will focus on recognition of words using CNN, given a Region of Interest (ROI) containing a well segmented word image. This means that the classification task does not take into account some essential aspects of handwritten text recognition.

II. Related Work

"Word Spotting" and "Word Recognition" have been extensively used for the purpose of document image retrieval. Depending on the scenario there exist different approaches investigated by various researchers.

An approach stated in [2] uses DTW and semicontinuous HMMs (SC-HMMs) for word spotting. Frinken et al.[3] proposed a method where the BLSTM neural network was coupled with CTC Token Passing algorithm for a segmentation free keyword spotting. In their scheme, they extracted 9 different geometric features by sliding a window of width 1 pixel from left to right over the normalized text line images. The neural network maps each position of an input sequence to a vector to indicate the probability of a particular character being written at that position. Later, the CTC Token Passing algorithm considers this sequence of letter probabilities, as well as a dictionary and a language model, as its input and computes a likely sequence of words[3]. In [4] the authors have proposed to embed both word images and text strings in a common vectorial subspace, by means of label embedding and attributes learning, and a common subspace regression. It was observed that in this subspace, images and strings that represent the same word are close together, which allowed to frame the recognition and retrieval tasks as a nearest neighbor problem. Recently Rusinol et al.[5] used the query-by-example paradigm for word spotting. In their method, initially, local patches were described by a bag-of-visual-words model generated from SIFT feature descriptors. In the later stage, the patch descriptors were projected to a topic space with the latent semantic analysis technique and compressing the descriptors with the product quantization method. In a HMM-based word recognition system [6], words that were to be recognize were represented as hidden states of the HMM and word bigram frequencies were used as the state transition probabilities. Word length and word profile were used as features to recognizing words from historical document images and gave an accuracy of $\approx 65\%$. In another recent endeavour on word spotting following query-by-string paradigm [7], word images are jointly represented by textual and visual form. The textual representation is formulated in terms of character n-grammes while the visual representation is based on the bag-of-visual-words scheme. These two representations are merged together and projected to a sub-vector space. Hence given a textual query, the system could retrieve word instances that were represented by the visual modality[7]. In [8], Ghosh et al. extended the research from [4] to a segmentation free approach.

In the recent past, some endeavours on word spotting are evident using deep learning-based recognition framework. The very first in this context is due to Sharma et al.[9] where a pre-trained CNN is deployed to learn classes of word images. The output is then used to perform word spotting.The study asserts that features extracted from an adapted-CNN can outperform hand-designed features on both spotting and recognition tasks for printed (English and Telugu) and handwritten document collections. However, their study does not deal with historical document images. Very recently Sudholt et al.[10] proposed a novel CNN architecture for the purpose of word spotting, they experimented with both contemporary as well as historical document images and obtained encouraging results. Their proposed system can be customized as a QBE or QBS based system. In [10] the network is trained with the help of Pyramidal Histogram of Characters (PHOC) representation. In the final layer of the network, instead of traditional softmax function the authors proposed to use a sigmoid activation function that is applied to every element of the output vector. Another deep learning based approach for Arabic word recognition is due to [11]. A deep recurrent neural network (RNN) and a statistical character language model-based approach is due to [12], where the same vast medieval manuscript collection as ours has been considered for indexing. In this

study, we will focus on recognition $argmax\, P(C|X)$, where C stands for the class hypothesis and X stands for the feature vector. However it is clear that a combined architecture with word spotting (where first we look for $argmax\, P(X|C)$ and then consequently confirm our finding through recognition is an interesting direction for future research.

III. Dataset Details & Motivation

Our "original" word samples are being procured from scanned images of initial few volumes of a large French administrative document collection (termed as "Chancery Corpus") produced by the French royal administrations during the period 1302 to 1310 (involving volumes 35-42 of the corpus). Those word samples were labelled in a crowd-sourced environment where using a web interface a user could label a bounding box/word region, we followed an approach similar to [13]. All crowd-sourced labelled word images were stored in grey scale single channel pgm format. In order to comply with the AlexNet input architecture while creating the dataset in lmdb file format, we used off-the-shelf converter from [14] to convert those single channel images to 3 channel equivalent format with a size of 256×256 pixels. However, no extra information was added as all 3 channels consist of the same grey scale values. There are 11,568 human labelled images in the corpus that has been used in the experiment, the rest of our images are synthetic/augmented in nature.

This corpus could be used for network analysis and to study the influence of French Royal Chancery. However, the mammoth size of this corpus is a barrier to the scholars who would like to study it exhaustively. To aid an user-friendly access to the contents of this corpus, recognition of text word in the corpus could be one possible way out. This research attempts to address the issue of word recognition in a historical document image corpus by customizing a traditional CNN architecture aptly to adapt the network for the purpose of word recognition.

IV. Methodology

The objective of this study is to investigate the prowess of deep-learned features for recognition of medieval word images when not enough amount of data is available per word class. So we were more keen to get hold of the activation values with respect to an input image, a concept close to [15]. Though deep learning techniques have been very successful in diverse image classification problems, the huge number of network parameters that we need to deal in a deep-learning framework should not be ignored. One way out to decrease the number of parameters could be tapping features from sufficiently deep layers of the network, so that the feature retains enough discriminative characteristics between classes. Those features can be later fed to any classifier to perform the final classification. This idea is the basis of procuring the output of the 2nd fully connected Layer of AlexNet as a feature vector to classify word images in our experiment. We termed those features as "**Tapped-FC7@4096**".

In this experiment only those classes of word image that consists of at least 5 "original" labelled samples in the corpus have been considered. So that in a five-fold cross-validation setup we have at least one "original" sample per class in each of those five folds. It is worth mentioning here that we did not perform data augmentation on the "original" samples before they were put into training and testing set for each fold. We wanted to ensure that in each of those five folds, the training and the testing set does not consist of augmented image samples derived from the same "original" image. Hence first original images for all classes were sorted into respective training and testing directories for each fold. Later, our data augmentation tools were executed to generate synthetic variants of those word images. We used an AlexNet architecture [16] from caffe [14] for our experiments involving Convolutional Neural Network. In each fold, the total number of images was divided in training (80%) and test dataset (20%); the former was subdivided in training "proper"(80%, ie 64% of total) and validation (20%, ie 16% of total) datasets.

A. Offline Data Augmentation

We took help of off-line data augmentation technique to generate more samples of images given a set of "original" image per word class. We used morphing and shearing operations on those "original" images to get augmented samples. We used the same method as in [1].

- **Elastic Morphing**- For every pixel (i,j) of the "original" image, a random displacement vector $(\Delta x, \Delta y)$ is generated. The displacement field of the complete image is smoothed using a Gaussian convolution kernel with standard deviation σ. The field is finally rescaled to an average amplitude A. The new morphed image $(\hat{i}, \hat{j})$ is generated using the displacement field and bilinear interpolation $(\hat{i} = i + \Delta x, \hat{j} = j + \Delta y)$. Thus the morphing process acts on the basis of the smoothing radius σ and the average pixel displacement "A" [1].
- **Shearing**- In a two-dimensional plane **shearing** is a function that maps a generic point with coordinates (x, y) to the point $(x + my, y)$; where m is a fixed parameter, called the shear factor.

The effect of our data augmentation technique is evident in Figure 3. The top two images are "original" sample of the words "aliqua" and "plene", below are their augmented version after morphing and shearing has been performed. It can be noted that due to the shearing operation the left word has been tilted a bit on the right whereas the right word got tilted towards left.

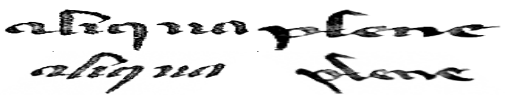

Fig. 3. (Top) Original word (Bottom) After morph and shear operation

B. Brief Discussion on CNN and its Parameter Details

The CNN architecture can be dissected into two distinct elements:(a)the first few layers of the network - which is responsible for performing the convolution with different filters and acts as a feature generator (b) the features generated in the first few layers are propelled to the deeper layers-which are known as fully connected layers. Here, multiple fully connected layers are stacked together to form the Multi Layer Perceptron architecture. Finally, applying the softmax function to the output of the last layer of the network, a vector is generated that gives the class probabilities for respective classes. There are few tiny technical issues that we need to address while deploying our CNN architecture. One such primary issue is setting the batch size of the images while training the network. In the recent past, it has been observed that using a larger batch size leads to a poorly generalized learning model[17]. The article [17] reveals the reason behind such undesirable behaviour. It has been observed that in models generated with huge batch size, large positive eigenvalues could be observed in the Hessian ($\nabla^2 f(x)$) of the objective function. Such large batch sized based methods converge to sharp minimizers of the training and testing functions and thus compromises in terms of generalization ability. On the other hand small batch sized based methods leads to small eigenvalues in the Hessian ($\nabla^2 f(x)$) of the objective function and hence converge to function landscape regions with flat minimizers. Keeping this observation into account we set our training batch size to 28. The batch size and number of test iteration parameter for the validation and test dataset were also meticulously chosen so that the number of images in the validation and test set for each fold is $\approx$ equal to the product of the batch size $\times$ number of validation/test iterations. To comply with this rule few images(2-7) were deleted from the test and validation set of all folds. However, the deleted images were chosen to be of the different class in all cases. The first convolutional layer consists of 96 filters/feature maps of size 11×11 where each filter elements were randomly generated using a Gaussian distribution. Since we did not perform any cropping, given the initial size of 256×256 of the input image, in comparison to regular AlexNet, there is a difference in the size of the produced activation map in all convolution layers(1-5). For example, the activation map produced in convolution layer 1 is of size 62×62, whereas in Regular AlexNet it becomes 55×55. Details can be seen in Figure 5. For all convolution layers and fully connected layers "Relu" was deployed as an activation function.

Due to the subtle change that we made in AlexNet to adapt to our word recognition problem, we decided to train our network from "**scratch**". Optimal network weights were obtained using a Stochastic Gradient Descent (SGD) optimizer with an initial learning rate of 0.001 and an inverse decay function was deployed to lower the learning rate as the number of iterations increases. The max. number of iterations while training, was set to 92000. It can be easily noted from Figure 4 that the loss function values with respect to iterations for all folds converge with similar value at higher iterations. Another point worth to mention here is that the original AlexNet CNN architecture uses a crop size of 224×224 with horizontal reflection (flipping). In this study, the original crop size was set to 256×256 and did not consider the flipping option for our modified CNN setup. The motivation for our setup is that flipping performs the online data-augmentation task and it is not relevant in our case of text recognition. Since the mirror version of the word may be a totally different word in the lexicon, this may lead to suboptimal recognition rates. Indeed, in mirrored version, words like "pond" and "poud" would be a single word. At this point, it is important to note that although to some extent, CNN's are aimed at handling positional and scale variation by using convolutional kernels, data augmentation and layer topology, the feature maps will be contaminated with some position and size information.

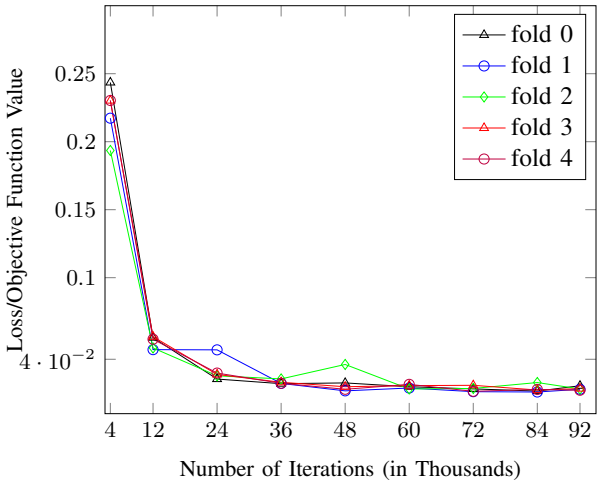

Fig. 4. Loss function value during training as a function of iteration number for each fold.

TABLE I
NUMBER OF IMAGES USED IN ALL FOLDS FOR TRAINING (TR), VALIDATION (VAL) AND TEST (TE) SETS WITH THEIR RESPECTIVE BATCH SIZES (BS).

Fold Number	**Total Number of TR**	**Total Number of VAL**	**Total Number of TE**	**Max. Number of Iter. for TR / VAL / TE**
Fold 0	176805	44300, BS: 50	50490, BS: 30	TR: 92000, VAL: 886, TE: 1683
Fold 1	178600	44790 BS: 30	50400, BS: 25	TR: 92000, VAL: 1493, TE: 2016
Fold 2	182756	45850 BS: 50	50000, BS: 50	TR: 92000, VAL: 917, TE: 1010
Fold 3	181278	45480, BS: 30	49980, BS: 28	TR: 92000, VAL: 1516, TE: 1785
Fold 4	183030	45920, BS: 35	50600, BS: 50	TR: 92000, VAL: 1312, TE: 1012

C. Classification of Tapped Features

The Support Vector Machine and different distance functions were used as classification tool to classify the tapped

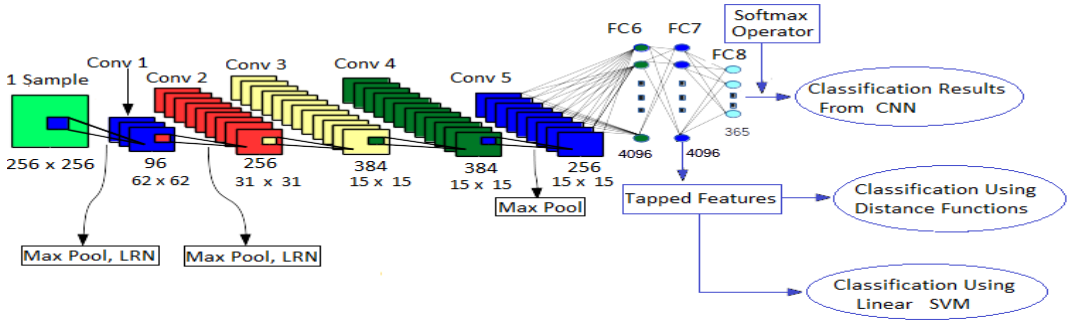

Fig. 5. Schemetic diagram of our customized AlexNet architecture as used in our experiment.

features. We had training images in the range of $176k - 183k$ in each fold (see Table I), Since we obtained reasonable classification accuracy with linear SVM, we avoided using non-linear kernel SVM, considering the need for huge computational resources in order to use the non-linear SVM.

V. Experimental Results

We evaluated our system in five-fold validation framework. We compared three different approaches for classification of word images;(a) simple distance function to compute the distance between the training class centroid vectors and the test vector;(b) classification using a Linear SVM while using the "**Tapped-FC7@4096**";(c) regular CNN classification with Softmax operator in the final layer to generate the final class with max. probability.

A. Classification Results with Tapped Features

We deployed 12 distance functions to classify our test samples. We used following distance functions in our experiment: Minkowski, Cosine, Correlation, Variance, Chi-square, Kullback, Dice, Jaccard, Tanimoto, Yulesq, Manhattan, Euclid. We computed the centroid vector for each word class from the training set and used the distance function to compute the distance from a test feature vector. Best performance was observed with Minkowski. However, computationally efficient Manhattan distance also yields 81.34%. The average accuracies for all five-folds are depicted in Table II along with the corresponding method being used. It can be noted that 11 out of 12 distance functions gave us a standard deviation of below 1 on accuracy for 5 folds .

1) SVM Classification Results: Using Linear SVM as classifier we obtained an average accuracy of 90.03% from all five-folds. See the first row in Table II. "Tapped Features" from the second fully connected layers were fed to the SVM from respective training and testing sets.

B. CNN Classification Results

We obtained an average recognition accuracy of ≈89% (best) in our five-fold experiment. We also observed the accuracy with respect to different training iterations. They are depicted in Table III.

TABLE II
Five-fold Average classification Accuracy with the standard deviation for Linear SVM and different Distance Functions.

Matching Function	Mean Accuracy (%) for all folds	Standard Deviation
Linear SVM	90.03	0.66
Minkowski(order=3)	85.36	0.57
Euclid	85.15	0.59
Variance	85.10	0.69
Cosine	84.89	0.54
Correlation	84.77	0.54
Tanimoto	83.72	0.62
Chi-square	83.08	0.87
Yulesq	82.55	0.49
Kullback	82.02	0.59
Dice	81.65	0.82
Jaccard	81.65	0.82
Manhattan	81.34	1.05

TABLE III
CNN Classification Accuracy in % on Test set with Respect to number of training iterations on training set in each of the Five Folds with FC8 trained features.

Fold	Iter. 24k	Iter. 48k	Iter. 60k	Iter. 72k	Iter. 84k	Iter. 92k
0	85.60	86.80	87.06	88.06	88.00	87.60
1	87.64	89.44	90.56	89.92	90.72	89.92
2	88.02	87.84	89.48	88.68	90.24	88.88
3	86.85	87.00	86.64	86.07	86.92	87.42
4	84.44	87.48	85.44	87.76	88.16	87.56

VI. Discussion & Analysis

A. Utility of the Tapped Features

We have noticed that simple Linear SVM classifier gave an average accuracy of 90.03% when being fed with the tapped features from the second fully connected layer of the network. With this positive results, the question arises whether the system has learned specific features for the classification purpose, or, alternatively more general features for "out of vocabulary" words. The classification accuracy shows stable and good results with several distance functions, indicating that the CNN derived "tapped features" are robust. Moreover,

we now can easily gauge the performance of the system in a transfer-learning setup, where the tapped features are computed from unseen word images in 81 unseen classes for training and testing. The training set features from 365 classes are concatenated with training set features from the 81 unseen classes. The test set consists of unseen samples from those 81 classes only. As in previous experiments, the centroid vectors are computed for each class to compute the distance for each test sample. This yielded 96% recognition rate on an average for 7 different distance functions(In this experiment, the number of tests samples were less than 200 in each fold and not in the order of 50k as in the first experiment). It is also worth mentioning that we need to deal with $4096 \times 365 = 1495040$ fewer network parameters if we simply prune the last layer and use the nearest centroid matching method.

B. Error Analysis

We were inquisitive to analyze the reason behind incurred misclassifications. We noted that the overall classification accuracy dropped below 80% for some word classes having ≤ 2 characters. Also, "misclassification" was observed between two classes having a similar visual appearance, for example,  and  .

VII. Conclusion

This study investigates the performance of deep-learned features for the task of medieval word image recognition. Our experiments show that data augmentation techniques can be effectively used to counter the problem of data scarcity per word image class. Apart from the regular CNN classification using a fully connected last classification layer, we show that also tapped hidden feature vectors yield a higher performance. We used the deep-learned features with different type of classification methods and obtained promising results from our experiments. If the approach proposed in this study is integrated into a large document-mining architecture like "Monk"[18], it depends on available computing resources whether one would use the SVM as the final classifier, otherwise, the nearest centroid approach is already yielding usable results. Future research directions could be evaluating other CNN architectures for word image recognition to deal with "Out Of Vocabulary" transfer task for data-mining of unseen classes.

VIII. Acknowledgements

This research work has been funded by NWO/Creative Industries grant (Project No.335-54-001) under the aegis of "HIstorical MANuscript Indexing for user-controlled Search" an EU project jointly funded by the Dutch, French, Spanish research authorities in the framework of the "Joint Programming Initiative on Cultural Heritage and Global Change".

References

[1] Marius Bulacu, Axel Brink, Tijn van der Zant, and Lambert Schomaker, "Recognition of handwritten numerical fields in a large single-writer historical collection," in *10th International Conference on Document Analysis and Recognition,ICDAR 2009, Barcelona, Spain, 26-29 July 2009*, 2009, pp. 808–812.

[2] José A. Rodríguez-Serrano and Florent Perronnin, "A model-based sequence similarity with application to handwritten word spotting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2108–2120, 2012.

[3] Volkmar Frinken, Andreas Fischer, R. Manmatha, and Horst Bunke, "A novel word spotting method based on recurrent neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 2, pp. 211–224, 2012.

[4] Jon Almazon, Albert Gordo, Alicia Fornes, and Ernest Valveny, "Word spotting and recognition with embedded attributes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 12, pp. 2552–2566, 2014.

[5] Marcall Rusinol, David Aldavert, Ricardo Toledo, and Josep Llados, "Efficient segmentation-free keyword spotting in historical document collections," *Pattern Recognition*, vol. 48, no. 2, pp. 545–555, 2015.

[6] Victor Lavrenko, Toni M. Rath, and R. Manmatha, "Holistic word recognition for handwritten historical documents," in *1st International Workshop on Document Image Analysis for Libraries (DIAL 2004), 23-24 January 2004, Palo Alto, CA, USA*, 2004, pp. 278–287.

[7] David Aldavert, Marcal Rusinol, Ricardo Toledo, and Josep Llados, "Integrating visual and textual cues for query-by-string word spotting," in *2013 12th International Conference on Document Analysis and Recognition, Washington, DC, USA, August 25-28, 2013*, 2013, pp. 511–515.

[8] Suman K. Ghosh, Lluís Gómez i Bigorda, Dimosthenis Karatzas, and Ernest Valveny, "Efficient indexing for query by string text retrieval," in *13th International Conference on Document Analysis and Recognition, ICDAR 2015, Nancy, France, August 23-26, 2015*, 2015, pp. 1236–1240.

[9] Arjun Sharma and K. Pramod Sankar, "Adapting off-the-shelf cnns for word spotting & recognition," in *13th International Conference on Document Analysis and Recognition, ICDAR 2015, Nancy, France, August 23-26, 2015*, 2015, pp. 986–990.

[10] Sebastian Sudholt and Gernot A. Fink, "Phocnet: A deep convolutional neural network for word spotting in handwritten documents," in *15th International Conference on Frontiers in Handwriting Recognition, ICFHR 2016, Shenzhen, China, October 23-26, 2016*, 2016, pp. 277–282.

[11] Alex Graves and Juergen Schmidhuber, "Offline handwriting recognition with multidimensional recurrent neural networks," in *Advances in Neural Information Processing Systems 21*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds., pp. 545–552. 2009.

[12] Théodore Bluche, Sebastien Hamel, Christopher Kermorvant, Joan Puigcerver, Dominique Stutzmann, Alejandro Héctor Toselli, and Enrique Vidal, "Preparatory KWS experiments for large-scale indexing of a vast medieval manuscript collection in the HIMANIS project," in *14th IAPR International Conference on Document Analysis and Recognition, ICDAR 2017, Kyoto, Japan, November 9-15, 2017*, 2017, pp. 311–316.

[13] Tijn van der Zant, Lambert Schomaker, and Koen Haak, "Handwritten-word spotting using biologically inspired features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 11, pp. 1945–1957, 2008.

[14] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM International Conference on Multimedia*, 2014, MM '14, pp. 675–678.

[15] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson, "Cnn features off-the-shelf: An astounding baseline for recognition," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 512–519.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems.*, 2012, pp. 1106–1114.

[17] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang, "On large-batch training for deep learning: Generalization gap and sharp minima," *CoRR*, vol. abs/1609.04836, 2016.

[18] Lambert Schomaker, "Design considerations for a large-scale image-based text search engine in historical manuscript collections," *it - Information Technology*, vol. 58, no. 2, pp. 80–88, 2016.