

# Learning Features for Writer Retrieval and Identification using Triplet CNNs

Manuel Keglevic, Stefan Fiel and Robert Sablatnig

Computer Vision Lab

Institute of Visual Computing & Human-Centered Technology

TU Wien

Vienna, Austria

Email: {keglevic, fiel, sab}@cvl.tuwien.ac.at

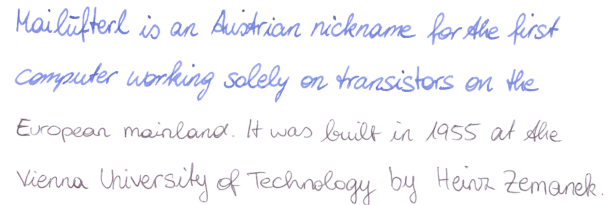
**Abstract**—This paper presents a method for writer retrieval and identification using a feature descriptor learned by a Convolutional Neural Network. Instead of using a network for classification, we propose the use of a triplet network that learns a similarity measure for image patches. Patches of the handwriting are extracted and mapped into an embedding where this similarity measure is defined by the  $L_2$  distance. The triplet network is trained by maximizing the interclass distance, while minimizing the intraclass distance in this embedding. The image patches are encoded using the learned feature descriptor. By applying the Vector of Locally Aggregated Descriptors encoding to these features, we generate a feature vector for each document image. A detailed parameter evaluation is given which shows that this method achieves a mean average precision of 86.1% on the ICDAR 2013 writer identification dataset, but future work has to be done to improve the performance on historic datasets. In addition, the strategy for clustering the feature space is investigated.

## I. INTRODUCTION

Writer retrieval is the task of retrieving document images with similar handwriting from a dataset. Experts then analyze this ranking and thus new documents from the same writer can be found in an archive. Furthermore, in case multiple documents from a single writer are found, connections between different manuscripts can be discovered. In modern context, writer retrieval methods are used in forensics to analyze ransom or threat letters. It can link different letters and thereby improve the chances of finding the author. In contrast to writer retrieval, writer identification is the task of finding the writer of a certain document. The writer has to be known in advance and their handwriting already analyzed for comparison. The procedure can be used to identify the writer of an unknown document in case several possible authors come into question.

The methods for both applications are similar. Both generate a feature vector, which describes the handwriting of a reference document in respect to a particular writer. This feature vector is then compared to the vectors of other documents in a database. By using a distance measure, the similarity of the handwritings is determined and either a ranking is generated or the writer with the smallest distances is assigned to the document.

The handwriting style of people depends on different parameters like which pen is used or external influences such as distractions by something or someone. Thus, the writing of a person exhibits slight changes from document to document;



Mailbüferei is an Austrian nickname for the first  
computer working solely on transistors on the  
European mainland. It was built in 1955 at the  
Vienna University of Technology by Heinz Zemanek.

Fig. 1. Sample image of the CVL dataset. The writer used two different pens, therefore the handwriting looks different.



Dann  
Dann  
Dann  
Dann

Fig. 2. Part of a sample image of the CVL dataset, where the German word “Dann” is written 4 times and looks different each time.

but also within a document itself, small variations occur. Figure 1 shows a sample page from the CVL Database [1] where the writer changed the pen during writing. For humans the handwriting looks different at first glance, but by taking a detailed look at for example the word “the”, it can be seen that the same person wrote all four text lines. Figure 2 shows another sample of the CVL Database with a text containing the German word “Dann” four times. The word is never written exactly the same; small variations in different characters occur. Methods for writer identification and retrieval have to deal with variations like these when applied to real world samples. Another challenge, which has not been covered by any scientific database so far, is that the handwriting changes with the age of the writer. Especially when these methods are applied to historic data, these variations must be investigated.

Features for successful writer identification or retrieval can be computed by analyzing the characters themselves, like

proposed by Marti et al. [2] by describing the slant and the heights of the different writing zones. Bulacu et al. [3] propose to use different features like contour direction, contour-hinge, and direction co-occurrence. The contours of the characters are also used for writer identification and retrieval by Jain and Doerman [4], who use Contour Gradient Descriptors. Other methods calculate local features on the document image describing the neighborhood of specific points. Fiel and Sablatnig for example use SIFT features in [5] and [6] which describe the neighborhood of keypoints. Nicolaou et al. [7] use Local binary patterns, which are calculated for each pixel.

Deep learning methods, which have arisen from digit recognition [8], have been proposed for various computer vision problems in the last years, like image classification [9] and recognition [10]. These methods have found their way back to the field of document image analysis, e.g. handwritten text recognition [11]. Recently, methods using deep learning have also been proposed for writer identification and retrieval by Chu and Srihari [12], Fiel and Sablatnig [13], Christlein et al. [14], [15], and Xing and Qiao [16].

These methods train Convolutional Neural Networks (CNN) on a classification task and use the activations of one of the last fully connected layers of the network as feature descriptor for each image patch and combine them afterwards to generate a feature vector for the complete document image. A natural choice for the targets for the classification are the writers of the training, as used by Fiel and Sablatnig [13]. More recently, Christlein et al. [15] showed that the use of unsupervised clustering to compute surrogate classes can improve the results.

In contrast to learning a classification task, this paper proposes to learn a similarity measurement between image patches using a triplet loss function. This is done using the triplet architecture proposed by Balntas et al. [17]. Triplets of image patches are presented to the network; always two positive (matching, i.e. same writer) and one negative one (non-matching, i.e. different writer). The network then tries to learn a mapping which minimizes the distance between the two positive ones and maximizes the two distances between the positive and negative samples. The distances are illustrated in Figure 3, with  $\Delta^+$  being the distance between the positive samples and  $\Delta_1^-$  and  $\Delta_2^-$  the distance between one positive and the negative sample, respectively. The image patches are then mapped into this embedding and their representations are used, like in Christlein et al. [15], to generate a Vector of Locally Aggregated Descriptors (VLAD) which can be used for retrieval or identification.

The contribution of this paper is that a similarity measure is learned directly from the handwriting, which represents the writing style. This mapping can then be used like traditional features for image patches. In the method proposed, a VLAD is generated for each image.

This paper is organized as follows: Section II describes the methodology that is proposed. Starting with the patch extraction, followed by the deep learning part, the generation of the VLAD, and ending with a whitening of the data as post-

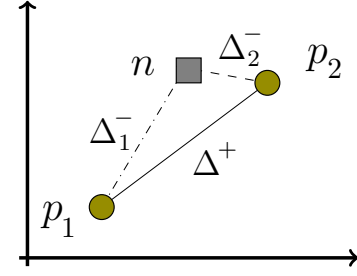


Fig. 3. Distances between the feature representations of a triplet in the embedding.

processing. In Section III a detailed evaluation of the method proposed is presented and a conclusion is given in Section IV.

## II. METHODOLOGY

This section describes the methodology proposed in detail. First, for pre-processing, the images are binarized and image patches are extracted. These patches are then presented to the network, which learns a mapping based on these patches, minimizing intraclass distances and maximizing interclass distances. The mapped representations obtained by the network are then used to generate a VLAD encoding of the writing style of a complete page. In a last step the data is whitened.

### A. Extraction of Patches

The method takes a binarized image as input. Binarization is principally not necessary for the rest of the pipeline, but since some databases are only provide binarized images, this step was introduced. Another reason is that, when dealing with historical data, the background does not have an influence on the learning of the features. For the patch extraction, the location of SIFT keypoints, which originate from the Harris corner detector, are used as centers of the patches. The advantage of SIFT keypoint locations is that previous methods, such as [5] and [6], have shown that there is enough information around these locations for a successful identification or retrieval and further, these keypoints lie on or near the strokes. They also show that even though the number of keypoints varies heavily, this has no negative influence on the performance. The size of the patches is  $32 \times 32$  pixels. Figure 4 shows sample images patches which have been extracted at the SIFT keypoint locations.

In [5] and [6] the SIFT features are filtered according to their size. The idea is to ignore the features with small and large sizes, since they are mostly located at the end of the line of a character or between text lines. [15] use the SIFT features to filter the patches after the creation of the surrogate classes, i.e. the clustering. For this, they use the distance ratio of the two distances between the closest and second closest cluster center. This filters out patches which lie between clusters and are thus not representative for any particular class.

We adopt this idea to filter out patches for the training step. However, in order to get character like clusters, we use a lower number of classes. The goal is to filter out patches

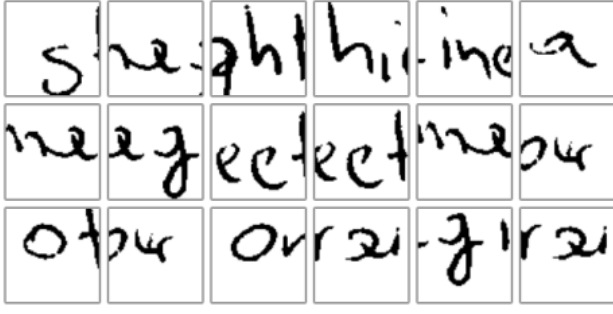


Fig. 4. Sample patches extracted at the SIFT keypoint locations.

with patterns that do not occur often and therefore do not form a cluster. This filtering is restricted to the training step, since during evaluation we might filter out patches containing writer specific features. The goal is, that our system learns to distinguish between different writers within these clusters.

### B. Learning the similarities

In contrast to other approaches using the output of the last layer of a fully connected CNN trained for classification, e.g. with a SoftMax layer and a Mean Square Error loss function, we propose the use of the triplet architecture described by [17]. Similarly to siamese networks, an embedding is learned using multiple CNN branches with shared weights in which the  $L_2$  distance can be used to measure similarities. Contrary to siamese networks however, the loss function is evaluated using negative and positive distances simultaneously in the triplet architecture with each triplet  $T = \{x_{p_1}, x_{p_2}, x_n\}$  consisting of two matching samples and one non-matching sample, i.e.  $x_{p_1}$ ,  $x_{p_2}$  and  $x_n$ , respectively. In our case the matching samples are determined using the writer label. As shown in Figure 5, during a single training step, the three samples of a triplet are forwarded through the three identical branches with shared weights, i.e. mapped into the embedding  $f(x_i)$ . In this embedding the loss function is defined so that the  $L_2$  distance between the positive samples, i.e. the positive distance  $\Delta^+$ , is minimized while the  $L_2$  distances between the positive samples and the negative sample, i.e. the negative distances  $\Delta_1^-$  and  $\Delta_2^-$ , are maximized. The dimension of the embedding is controlled by the size of the last layer of the CNN branches. Since the weights are shared between all three branches, only one branch is needed during inference while the other two can be discarded after the training is finished.

For each triplet  $T = \{x_{p_1}, x_{p_2}, x_n\}$  the distances in the embedding  $f(x_i)$  are then defined as

$$\begin{aligned}\Delta^+ &= \|f(x_{p_1}) - f(x_{p_2})\|_2 \\ \Delta_1^- &= \|f(x_{p_1}) - f(x_n)\|_2 \\ \Delta_2^- &= \|f(x_{p_2}) - f(x_n)\|_2.\end{aligned}$$

The triplet loss function can now take either only one [18] or both [17] negative distances into account. However, by forcing the positive distance to be smaller than both negative distances, as proposed by Balntas et al. [17], an implicit soft negative

mining is performed, leading to a faster convergence of the network. The triplet loss function is then defined as in [17]

$$\ell(T) = \left( \frac{e^{\Delta^+}}{e^{\Delta^+} + e^{\Delta^*}} \right)^2 + \left( 1 - \frac{e^{\Delta^*}}{e^{\Delta^+} + e^{\Delta^*}} \right)^2$$

with  $\Delta^* = \min(\Delta_1^-, \Delta_2^-)$ , which can for instance be implemented using a Softmax layer and the Mean Square Criterion.

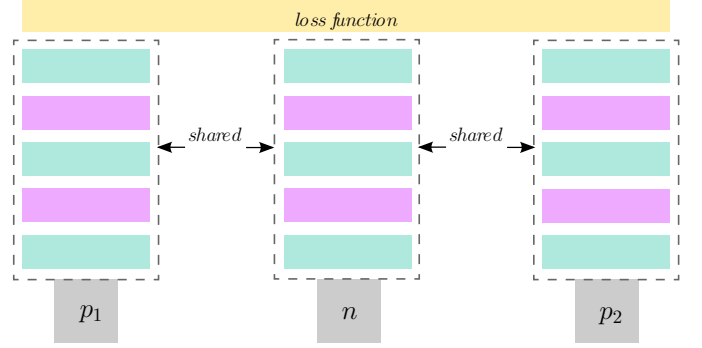


Fig. 5. Triplet architecture

In contrast to the network proposed by Balntas et al., we employ a DenseNet CNN architecture [19] for each of the branches, which has shown to outperform other network architectures on object recognition benchmarks like CIFAR-10 and ImageNet. As in other architectures, a sequence of convolutional layers with rectified linear units (ReLU) as activation functions and batch normalization is used. However, this architecture utilizes densely connected blocks in which all the feature layers of the previous layers are concatenated to the current input. In this way, state-of-the-art performance can be achieved while greatly reducing the number of parameters of the network. In Figure 6, a dense block with five layers is shown.

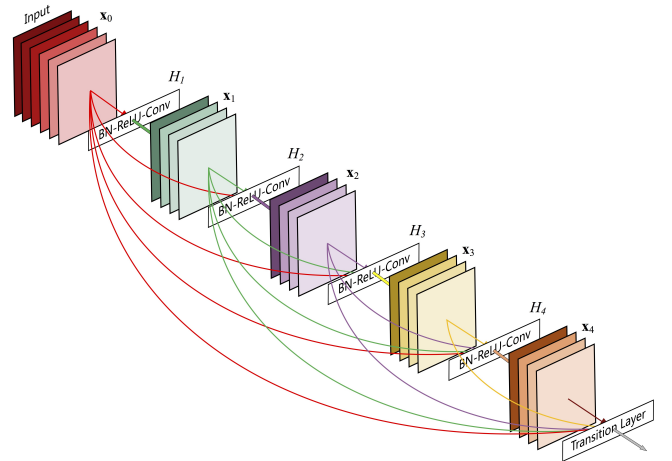


Fig. 6. Dense block with 5 layers. Image taken from [19]

As in the architecture proposed for the CIFAR-10 benchmark, we use a total number of 50 layers with a growth rate

$k = 12$  and 3 blocks. To compress the number of channels  $1 \times 1$  convolutions are used as *bottleneck* layers as proposed by Huang et al. [19].

The output of the last layer determines the embedding dimension. As in [15] we set it to 128 and additionally evaluated embedding dimensions of 32, 64 and 256.

### C. VLAD Encoding

The patches extracted from the document image are mapped into the embedding learned by the CNN. Their representations are then encoded to form a feature vector for each document image. This is done by using the VLAD encoding [20] a simplified non-probabilistic version of the Fisher Vector which has also been successfully applied to writer retrieval and identification by Christlein et al. [14]. It outperforms the bag of words methods and provides comparable results to the Fisher Vector [20]. Similarly to the bag of words method, k-means with  $k$  cluster centers is used to learn a vocabulary  $\{\mu_1, \dots, \mu_k\}$ . However, since the residuals to the cluster centers are accumulated, this has the advantage that the separation of the feature space is not as strict as when just counting the occurrences of the features in the clusters.

The input for the k-means clustering are the mapped images patches  $\mathcal{X} = \{f(x_t), t = 1 \dots T\}$  from the training set, where  $f$  is the mapping function learned by the CNN. Every input feature  $f(x_t)$  with dimension  $D$  is then assigned to its nearest cluster center  $NN(f(x_t))$ . For each cluster, all the residuals between the cluster center and the assigned features are accumulated:

$$v_i = \sum_{f(x_t): NN(f(x_t))=i} f(x_t) - \mu_i$$

The feature vector for a document can then be generated by concatenating all the  $k$  vectors  $v_i$ :

$$F = (v_1^T, \dots, v_k^T)^T$$

Thus, a document image is represented by a  $kD$ -dimensional feature vector where  $k$  is the number of clusters used for the vocabulary and  $D$  is the dimension of the embedding.

### D. Whitening

Whitening of the data is applied to limit the impact of visual word co-occurrences as proposed by [21]. To estimate the Covariance matrix as  $C = F \times F^T$ , the VLAD features of the training database  $F = [F_1 | \dots | F_n]$  are used. Each vector  $F_i$  represents the feature vector for an image in the training set after *power-law* normalization and centering around the mean. The *power-law* normalization is applied to each feature vector  $F_i = (v_1, \dots, v_{D_F})$  with dimension  $D_F$  by computing  $v_i = \sqrt{|v_i|} \cdot \text{sign}(v_i)$  for all  $1 \leq i \leq D_F$  followed by a re-normalization of  $F_i$  using the  $L_2$  norm.

Using Singular Value Decomposition (SVD) the covariance matrix  $C$  is then decomposed into the diagonal matrix containing the eigenvalues  $\text{diag}(\lambda_1^{-\frac{1}{2}}, \dots, \lambda_{D_F}^{-\frac{1}{2}})$  and the eigenvectors  $V^T$ . To reduce the dimensionality only the  $D'_F \leq D_F$

largest eigenvalues  $\lambda_i | 1 \leq i \leq D'_F$  and corresponding eigenvectors  $V_{D'_F}^T$  can be kept. Whitening is then performed on the centered and *power-law* normalized feature vector  $X$  of an image as follows [21]:

$$\hat{X} = \frac{\text{diag}(\lambda_1^{-\frac{1}{2}}, \dots, \lambda_{D'_F}^{-\frac{1}{2}}) V_{D'_F}^T X}{\left\| \text{diag}(\lambda_1^{-\frac{1}{2}}, \dots, \lambda_{D'_F}^{-\frac{1}{2}}) V_{D'_F}^T X \right\|}$$

As noted by Jegou et al. [21] the re-normalization factor is crucial to achieve a performance improvement (they report a performance increase of up to 10% on their dataset). As proposed by Jegou et al. [21] we use whitening to jointly decorrelate multiple vocabularies. For this, we compute multiple feature vectors with a varying number of cluster centers  $k_0, \dots, k_N$ . We start with a maximal number of clusters  $k_0$ , which is then halved for each following vocabulary. To make the results comparable with the use of a single vocabulary,  $k_0$  is derived from the total number of cluster centers  $k_\Sigma$ :

$$k_0 = k_\Sigma \frac{1 - q}{1 - q^N}$$

$$k_n = (k_n - 1) * q$$

with  $q = 1/2$ .

## III. EVALUATION

This section presents experiments, which are carried out on the dataset of the “ICDAR 2013 Competition on Writer Identification”[22]. The training set consists of 400 pages, written by 100 writers, whereas the evaluation set contains 1000 pages written by 250 writers. Each author contributed 4 pages to the dataset, two in English and two in Greek. We focus on this dataset, since it contains modern handwriting with two different alphabets. We use the training set for learning the similarity measure of the patches as well as for creating the vocabularies. The evaluation set is used for evaluation only. The evaluation is done using a leave-one-out strategy. Each document is taken once as reference document and a ranking according to the similarity of the other documents in the dataset is generated. These rankings are analyzed using the Mean Average Precision (MAP) since it also takes the position of the correct documents in the ranking into account.

First we extract the patches on both datasets, resulting in about 640k and 2.1M patches for the training and evaluation dataset, respectively. For the training of the triplets we filter the patches using the surrogate classes as described in Section II-A. In this step, we reduce the number of patches to about 300k. These patches are then used to generate triplets; for each training epoch, we use 1.28M. We do the evaluation with different vocabulary sizes, i.e. total number of VLAD cluster centers. Additionally, we evaluate using either a single VLAD vocabulary or 5 vocabularies with sizes derived as described above. As feature descriptor for each patch we use the whitened output of the trained CNN. For evaluation, we use the Euclidean as well as the cosine distance, since the

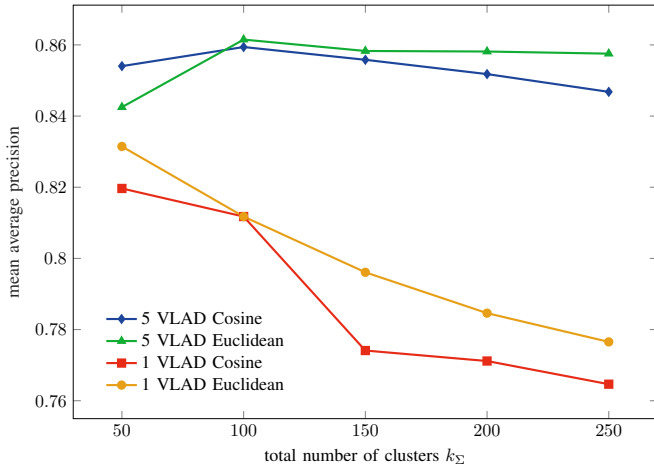


Fig. 7. Evaluation on the ICDAR13 test dataset with varying number of clusters and VLADs using cosine and euclidean distance. For training 100 surrogate classes were used.

network learns an Euclidean metric and whitened data usually has a good performance when using the cosine distance.

Figure 7 shows the MAP on the evaluation dataset, with a varying total number of clusters. Furthermore, we compare multiple VLADs against the usage of a single vocabulary. Additionally, the Euclidean distance and the cosine distance are used. For the training, we use 100 surrogate classes for filtering out patches as described above. This increases the performance compared to taking all patches. We determined the number of surrogate classes empirically by analyzing the results of 50, 100, 500, 1000, and 5000 classes.

We achieve the best performance of 86.1% MAP when using 5 VLADs with a total number of 100 cluster centers and the Euclidean distance. Multiple vocabularies outperform a single one in every experiment we did, especially when the total number of cluster centers is increased. Yet, for low total numbers of cluster centers this difference is modest. This can be explained by the small individual vocabulary sizes in these cases. For instance, for a total number of 50 centers, the sizes of the 5 vocabularies are just 25, 12, 6, 3, and 1. Nonetheless, using whitening to jointly decorrelate the multiple vocabularies is crucial for the performance. We also did experiments with 10 vocabularies, which did not lead to an improvement of the results.

Further, it can be seen that the Euclidean distance performs better and is more robust to changes in the total number of centers. Since this is not restricted to the usage of multiple VLADs we conclude that the Euclidean distance is better suited for our method. We also did some experiments with different sizes of the last linear layer of the network, which is our feature dimension in the embedding. When lowering the dimension to 64 or 32 the performance drops slightly. Yet, by increasing the last linear layer to 256 the improvements are not significant enough to warrant doubling of the embedding dimension. This suggests, that a dimension of 128 is a good trade off between performance and feature descriptor size.

TABLE I  
COMPARISON OF THE METHOD PROPOSED TO TWO OTHER  
STATE-OF-THE-ART METHODS.

	MAP	Top 1	hard Top 2	Top3
Christlein et al.[14]	88.0	99.4	81.0	61.8
Fiel and Sablatnig[6]	67.4	94.5	48.0	25.7
proposed	86.1	98.9	77.9	56.4

Table I shows the performance of the method proposed compared to two state-of-the-art methods on the ICDAR 13 dataset. It can be seen that our method performs slightly worse (2%) than [14], but significantly better than [6] which uses SIFT features for writer identification. All methods exhibit a performance drop when using the Top 2 criterion. Since all writers have two pages in Greek and two pages in English in the dataset, a document image written in the other language has to be found for this. Nevertheless, since the proposed method has a higher performance drop than [14] it can be concluded that the change of alphabet has a higher influence here.

The next step will be the extension of our method to an application on historic datasets. For this, the problem with different alphabets has to be addressed, since historic datasets, like the “ICDAR2017 Competition on Historical Document Writer Identification (Historical-WI)” [23], consist of different script types throughout different centuries. Furthermore, since real world data is used, it varies highly in font size, denseness of the text, and contains noise. Further improvements need to be made for the pre-processing step, in order to overcome these challenges and possibly also for the post-processing procedure. These improvements include the extension of the patch extraction. First, techniques for filtering patches that do not contain any writer information have to be investigated. Second, the influence of the distances to the two nearest surrogate class centers has to be examined. Currently we are following Christlein et al.’s [15] approach by filtering patches with a ratio greater than 0.9.

#### IV. CONCLUSION

This paper proposes a method for writer identification, which is based on learning an embedding representing the similarity of patches extracted from handwritten document images. For the extraction of the patches, the locations of SIFT features are used. To filter out unrepresentative patches in the training process, the idea of surrogate classes has been adopted by only taking patches with SIFT features near the centers of character-like clusters. The patches are then fed into a CNN network, which learns an embedding where patches from the same writer have a small distance and patches from different writers have a larger distance. For each patch the output of the last linear layer of the network is taken and a VLAD encoding is generated. In the evaluation, different numbers of centers for the VLAD are compared as well as the usage of multiple VLAD vocabularies. The evaluation is performed on the ICDAR 13 dataset, where the method



proposed, achieves nearly state-of-the-art results. Future work includes the application of the method on historic databases, from which new challenges will arise. Thus, especially the pre-processing step of extracting and filtering patches has to be improved.

#### ACKNOWLEDGMENT

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 674943. The Titan X used for this research was donated by the NVIDIA Corporation.

#### REFERENCES

- [1] F. Kleber, S. Fiel, M. Diem, and R. Sablatnig, "CVL-DataBase: An Off-Line Database for Writer Retrieval, Writer Identification and Word Spotting," in *2013 12th International Conference on Document Analysis and Recognition (ICDAR)*, 2013, pp. 560–564.
- [2] U.-V. Marti, R. Messerli, and H. Bunke, "Writer Identification Using Text Line Based Features," in *2001 6th International Conference on Document Analysis and Recognition (ICDAR)*, 2001, pp. 101–105.
- [3] M. Bulacu and L. Schomaker, "Text-Independent Writer Identification and Verification Using Textural and Allographic Features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 4, pp. 701–717, apr 2007.
- [4] R. Jain and D. Doermann, "Writer Identification Using an Alphabet of Contour Gradient Descriptors," in *2013 12th International Conference on Document Analysis and Recognition (ICDAR)*, Aug 2013, pp. 550–554.
- [5] S. Fiel and R. Sablatnig, "Writer Retrieval and Writer Identification Using Local Features," in *2012 10th IAPR International Workshop on Document Analysis Systems (DAS)*, March 2012, pp. 145–149.
- [6] —, "Writer Identification and Writer Retrieval Using the Fisher Vector on Visual Vocabularies," in *2013 12th International Conference on Document Analysis and Recognition (ICDAR)*, 2013, pp. 545–549.
- [7] A. Nicolaou, A. D. Bagdanov, M. Liwicki, and D. Karatzas, "Sparse radial sampling lbp for writer identification," in *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, Aug 2015, pp. 716–720.
- [8] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov 1998.
- [9] D. C. Cireşan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, "Flexible, high performance convolutional neural networks for image classification," in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Two*, ser. IJCAI'11. AAAI Press, 2011, pp. 1237–1242.
- [10] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, pp. 1–42, April 2015.
- [11] J. A. Sánchez, V. Romero, A. H. Toselli, and E. Vidal, "ICFHR2016 Competition on Handwritten Text Recognition on the READ Dataset," in *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, Oct 2016, pp. 630–635.
- [12] J. Chu and S. Srihari, "Writer identification using a deep neural network," in *Proceedings of the 2014 Indian Conference on Computer Vision Graphics and Image Processing*, ser. ICVGIP '14. New York, NY, USA: ACM, 2014, pp. 31:1–31:7.
- [13] S. Fiel and R. Sablatnig, "Writer Identification and Retrieval using a Convolutional Neural Network," in *16th International Conference, Computer Analysis of Images and Patterns (CAIP) 2015*, 2015, pp. 26–37.
- [14] V. Christlein, D. Bernecker, and E. Angelopoulou, "Writer Identification Using VLAD Encoded Contour-Zernike Moments," in *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, 2015, pp. 906–910.
- [15] V. Christlein, M. Gropp, S. Fiel, and A. K. Maier, "Unsupervised feature learning for writer identification and writer retrieval," in *14th IAPR International Conference on Document Analysis and Recognition, ICDAR 2017, Kyoto, Japan, November 9-15, 2017*, 2017, pp. 991–997.
- [16] L. Xing and Y. Qiao, "Deepwriter: A multi-stream deep cnn for text-independent writer identification," in *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, Oct 2016, pp. 584–589.
- [17] V. Balntas, E. Johns, L. Tang, and K. Mikolajczyk, "Pn-net: Conjoined triple deep network for learning local image descriptors," *CoRR*, vol. abs/1601.05030, 2016. [Online]. Available: <http://arxiv.org/abs/1601.05030>
- [18] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," *CoRR*, vol. abs/1412.6622, 2014. [Online]. Available: <http://arxiv.org/abs/1412.6622>
- [19] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
- [20] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid, "Aggregating local image descriptors into compact codes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1704–1716, Sep. 2012.
- [21] H. Jégou and O. Chum, "Negative evidences and co-occurrences in image retrieval: The benefit of PCA and whitening," in *Computer Vision - ECCV 2012 - 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part II*, ser. Lecture Notes in Computer Science, vol. 7573. Springer, 2012, pp. 774–787.
- [22] G. Louloudis, B. Gatos, N. Stamatopoulos, and A. Papandreou, "ICDAR 2013 Competition on Writer Identification," in *2013 12th International Conference on Document Analysis and Recognition (ICDAR)*, Aug 2013, pp. 1397–1401.
- [23] S. Fiel, F. Kleber, M. Diem, V. Christlein, G. Louloudis, N. Stamatopoulos, and B. Gatos, "ICDAR2017 Competition on Historical Document Writer Identification (Historical-WI)," in *2017 14th International Conference on Document Analysis and Recognition (ICDAR)*, 2017, pp. 1377–1382.