# Towards End-to-End Deep Learning-based Writer Identification

Zhenghua Wang[1] Andreas Maier[1] Vincent Christlein[1]

**Abstract:** Writer identification is an important task to gain knowledge about life in the past, which is commonly solved by paleographic experts. In this work, we investigate an automatic writer identification procedure based on deep learning. So far, the most approaches are based on two or more different pipeline steps and only few of them can be trained in an end-to-end manner. In this paper, we propose a fully end-to-end deep learning-based model, which consists of a U-Net for binarization, a ResNet-50 for feature extraction, and an optimized learnable residual encoding layer to obtain global descriptors. We evaluate the proposed end-to-end model on the ICDAR17 competition dataset on historical document writer identification (Historical-WI) dataset. Moreover, we investigate the performance of our optimized encoding layer on three texture datasets. While the optimized encoding layer does not work well in the task of writer identification, it provides better performance on the texture datasets. Furthermore, we show that a pre-trained U-Net can improve the performance for writer identification.

**Keywords:** writer identification; writer retrieval; deep learning; end-to-end

## 1 Introduction

Writer identification aims to retrieve the writer of a query document image in a dataset. It is playing a more and more important role for history sciences and especially for paleography, where it can help to search through a large dataset. The typical scenario is to obtain a short list, e. g. 20 samples that are most similar to the query sample of the whole dataset. In this way, the respective scientist only needs to check this short list, because with a high probability, it contains the correct writer.

Typical retrieval methods make use of the penultimate layer of a trained Convolutional Neural Network (CNN). However, script has not an object-structure, thus common writer identification methods [Ch15; Ch17; KFS18] rely on local descriptors, which are nowadays often learned by a CNN. Overall, these methods consist of three main stages: (1) pre-processing, (2) local feature extraction to obtain local descriptors, and (3) global descriptor computation to obtain a global representation. The pre-processing stage aims at segmenting the text from image patches. Afterwards, local features for each image patch are extracted. Finally, in the encoding stage, a global feature descriptor is computed from all local descriptors of the document image. Until now, each stage of the proposed method is

[1] Friedrich-Alexander-Universität Erlangen-Nürnberg, Pattern Recognition Lab, Martensstr. 3, 91058 Erlangen, Germany, firstname.lastname@fau.de

doi:10.18420/inf2020_126

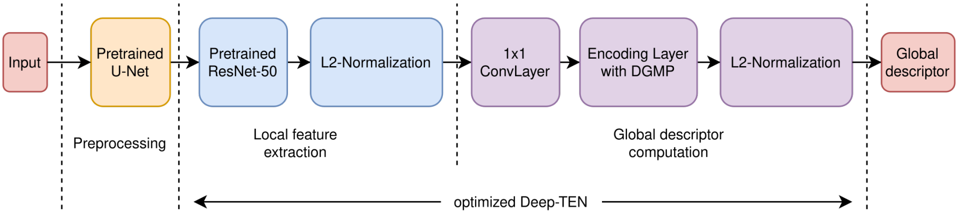

Fig. 1: Architecture of proposed end-to-end deep learning-based writer identification network.

optimized individually. In this paper, we aim to offer and evaluate a fully end-to-end deep learning-based model for writer identification.

In detail, our contributions are as follows: (1) We pre-train a U-Net [RFB15] on the Document Image Binarization Competition (DIBCO) datasets and transfer it to our end-to-end model and evaluate if a fine-tuning is beneficial. (2) We propose the use of the Deep-Ten method [ZXD17] as an encoding layer to form global descriptors. Additionally, we integrate Deep Generalized Max Pooling (DGMP) [Ch19], and evaluate it on both a writer identification dataset and texture datasets. (3) An end-to-end deep learning-based writer identification model is proposed and evaluated on the Historical-WI dataset. The end-to-end model is constructed using a pre-trained U-Net, residual network (ResNet) [He16] with 50 layers and an optimized encoding layer, see Fig. 1.

The rest of the paper is organized as follows. Sect. 2 gives an overview of the related work in encoding techniques and writer identification. The methodology of our end-to-end model is explained in Sect. 3. In Sect. 4, the evaluation protocol and results are shown. Finally, the conclusions are in Sect. 5.

## 2 Related Work

In this work, we focus on the group of codebook-based methods. Codebook-based methods create a global descriptor for each image by encoding local feature descriptors. Wu et.al [WTB14] apply SIFT [Lo99] to extract descriptors for word regions. In contrast, Christlein et al. [CBA15] calculate Contour-Zernike moments as local descriptors before aggregation using Vector of Locally Aggregated Descriptors (VLAD) [Jé12].

Nowadays, deep learning-based approaches are proposed for computing local feature descriptors. Fiel et al. [FS15] first propose a writer identification method based on document line segmentation and CNN activation features. In a concurrent work, Christlein et al. [Ch15] suggest to generate a global descriptor by computing Gaussian mixture model (GMM) supervectors to encode CNN activations. Christlein et al. [Ch17] also propose an unsupervised writer identification method. Keglevic et al. [KFS18] propose to train a DenseNet with triplet loss function [SKP15] to learn a similarity measurement between writers.

However, the pre-processing and global descriptor computation (encoding) stages of all above models are non-deep learning-based and optimized individually. In our work, we incorporate the binarization step into an end-to-end trainable network, by using a U-Net. A similar method was also proposed by Tensmeyer et al. [TM17], who adopt a fully convolutional network (FCN) [LSD15] for the binarization of handwriting images. As an extension of FCN, U-Net provides better segmentation performance with a small training dataset [RFB15]. For the encoding stage, Arandjelovic et al. [Ar16] introduce a new generalized VLAD layer, which is trainable on any CNN network. Zhang et al. [ZXD17] improves upon this approach by proposing a general residual encoding layer integrating the dictionary learning and residual encoding into a single learnable layer, called Deep Texture Encoding Network (Deep-TEN) model. Moreover, Christlein et al. [Ch19] suggest to apply Deep Generalized Max Pooling (DGMP) instead of global max pooling [SF16] or global average pooling [He16] when aggregating local embeddings. DGMP balances the activations of specific locations to address over-represented activations. In our work, we optimize the encoding layer of Deep-TEN model by fusing the idea of DGMP and add it on top of the local feature extraction layer.

## 3 Methodology

Our end-to-end model consists of three deep learning-based parts, a pre-trained U-Net [RFB15] for document images binarization, ResNet-50 [He16] architecture for local descriptors extraction and an optimized encoding layer for global descriptors computation.

### 3.1 U-Net Pre-training

While layout can give also clues about a writer, we want to rely solely on the script for writer identification. We suggest to apply the common deep learning-based segmentation network *U-Net* before the local feature extraction. Since the Historical-WI dataset does not provide the ground truth for training such a binarization network, we pre-train a U-Net on the DIBCO datasets,[2] which are document image datasets with ground truth for the training of segmentation networks. Afterwards, the U-Net can either be hold fixed or fine-tuned further. Instead of fine-tuning, we could also train the network from scratch but early experiments showed that this was not beneficial.

A huge number of parameters in the transferred network might cause overfitting when it is fine-tuned [Yo14]. Especially in our end-to-end model, the following ResNet-50 already contains a large number of parameters. Therefore, we shrink the size of the transferred U-Net by reducing the number of channels in the first convolutional layer of standard U-Net and maintaining its ratio in the contracting and expansion path [MS18]. Through experimental

[2] We used all available DIBCO datasets below `https://vc.ee.duth.gr/dibco2019/` (excluding year 2019)

validation, we reduce the number of channels in the first convolutional layer to 16 without deteriorating the network performance.

### 3.2 Optimized Learnable Residual Encoding Layer

After the local feature extraction, a global descriptor for each document image is created using robust residual encoders. In our work, we employ Deep-TEN [ZXD17], which can be seen as a learnable version of VLAD [Jé12], to build our end-to-end writer identification system. Additionally, we integrate DGMP [Ch19] to weigh each local embedding.

The computation of the global representation can be split into two stages: an *embeddig* and an *aggregation* phase [Mu16]. The embedding function $\phi$ maps each local descriptor to a high-dimensional space, while the aggregation function $\psi$ computes a single global descriptor from the local embedded descriptors, typically by means of sum-pooling. Assuming we have $N$ local descriptors $\boldsymbol{X} = \left\{\boldsymbol{x}_i \in \mathbb{R}^D, i = 1, ..., N\right\}$, the global descriptor is defined as:

$$\boldsymbol{\xi} = \psi\left(\phi\left(\boldsymbol{X}\right)\right) \ . \tag{1}$$

Specifically, if we employ the residual encoding model for the embedding phase, and given a learned codebook $\mathbf{C} = \left\{\boldsymbol{c}_k \in \mathbb{R}^D, k = 1, ..., K\right\}$ with $K$ codewords, we obtain:

$$\phi\left(\boldsymbol{x}_i\right) = a_{ik}\boldsymbol{r}_{ik} \ , \tag{2}$$

where $\boldsymbol{r}_{ik} = \boldsymbol{x}_i - \boldsymbol{c}_k$ represents the residual vector for each local descriptor, $a_{ik}$ is the weight for assigning the local descriptor to the codewords.

**Deep-TEN:** Zhang et al. [ZXD17] suggests that the codewords are learnable parameters and the weights $a_{ik}$ are calculated by extending the soft-assignment with a learnable smoothing factor $s_k$ for each codeword:

$$a_{ik} = \frac{\exp\left(-s_k \left\|\boldsymbol{r}_{ik}\right\|^2\right)}{\sum_{j=1}^{K} \exp\left(-s_j \left\|\boldsymbol{r}_{ij}\right\|^2\right)} \ . \tag{3}$$

**Deep-TEN + DGMP:** In our work, instead of using global sum pooling for the aggregation step as Zhang et al. [ZXD17] suggests, we adopt a DGMP [Ch19] layer and integrate it in the encoding layer. Therefore, we introduce a learnable weight $\boldsymbol{\beta} \in \mathbb{R}^{1\times N}$:

$$\boldsymbol{\beta} = \left(\mathbf{K} + \lambda\mathbf{I}_N\right)^{-1}\mathbf{1}_N \ , \tag{4}$$

where $\mathbf{K}$ is the Gram matrix of embeddings, $\lambda$ is a regularization parameter, $\mathbf{I}_N$ and $\mathbf{1}_N$ are the $N \times N$-dimensional identity matrix and the $N$-dimensional vector with each element being 1, respectively. Finally, the global descriptor becomes $\boldsymbol{\xi} = \left(\boldsymbol{\xi}_1^\top, ..., \boldsymbol{\xi}_K^\top\right)^\top$, where

$$\boldsymbol{\xi}_k = \psi\left(\phi\left(\boldsymbol{X}\right)\right) = \sum_{i=1}^{N} \beta_i a_{ik}\boldsymbol{r}_{ik} \ . \tag{5}$$

# 4 Evaluation

In this paper, we first investigate how much accuracy gain can we obtain when applying the optimized encoding layer (DeepTEN+DGMP) on texture datasets compared to the original encoding layer proposed in [ZXD17]. Afterwards, we evaluate our end-to-end model on the Historical-WI dataset [Fi17].

## 4.1 Data

**Texture datasets:** (1) The MIT-Indoor [QT09] dataset consists of 67 categories and works for indoor scene recognition. Similar to the experiments mentioned in [ZXD17], we use the standard splittings, i. e. contains 80 images of each category for training and 20 for testing. (2) The outdoor dataset, Flickr Material Dataset (FMD) [Sh13], contains 10 different material categories. For each category, there are 90 images in the training set and 10 in the test set. (3) The publicly available MINC dataset is a large-scale outdoor material dataset with 23 categories. For each category, there are 2500 images. We train the models with 2250 images per category and evaluate the models with 250 images per category.

**The Historical-WI dataset [Fi17]:** It consists of 4782 handwriting document images. In this dataset, there are 1182 pages provided by 394 writers (each contributes 3 pages) for training, 3600 pages by 720 writers (each contributes 5 pages) for testing. Note that there is no writer providing pages for both training and test set, i. e. the two data splits are disjoint. Next to the color version, the dataset also provides automatically computed binarized images by means of the method of Su et al. [SLT10]. Thus, the segmentation is not always exact, especially in the case of ink artifacts and bleed-through artifacts.

## 4.2 Experiments

All experiments in this subsection are repeated three times using different random seeds. The applied ResNet50 networks are pre-trained on the ImageNet dataset. First, we evaluate the optimized encoding layer on texture datasets. Then the performance of the end-to-end model on the Historical-WI dataset is evaluated in the second part.

**Evaluation of optimized encoding layer on texture datasets:** We re-implemented the Deep-TEN model [ZXD17] (Deep-TEN (ours)) and evaluated them on three texture datasets. Afterwards, we evaluate the optimized Deep-TEN (Deep-TEN + DGMP) model by replacing the original encoding layer while keeping the rest unchanged.

Tab. 1: Comparison of the optimized Deep-TEN model with the original Deep-TEN model and our re-implementation on three runs.

| Model | MIT-Indoor | | | | FMD | | | | MINC-2500 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1st | 2nd | 3rd | Avg. | 1st | 2nd | 3rd | Avg. | 1st | 2nd | 3rd | Avg. |
| Deep-TEN (orig) [ZXD17] | - | - | - | 71.3 | - | - | - | 80.2 | - | - | - | 80.6 |
| Deep-TEN (ours) | 69.5 | 69.5 | 69.6 | 69.6 | 75.4 | 74.5 | 75.1 | 75.0 | 77.2 | 77.2 | 77.0 | 77.1 |
| Deep-TEN + DGMP | 71.4 | 71.6 | 71.3 | 71.4 | 78.8 | 77.8 | 77.4 | 78.0 | 78.5 | 78.6 | 78.4 | 78.5 |

Our implementation follows the practice in paper [ZXD17]. For data augmentation, all images are normalized and resized to 400 × 400. The images in the training set are randomly flipped horizontally (50 % probability) and randomly cropped between 9 % and 100 % of the image areas. The aspect ratio is kept between 3/4 and 4/3. The standard color augmentation [KSH12] is applied. Instead of using stochastic gradient descent (SGD), we apply Adam [KB14] optimizer. The learning rate is initialized to $10^{-4}$ and multiplied 0.1 when the classification accuracy plateaus. A uniform distribution in range $\left[-\frac{1}{\sqrt{K}}, \frac{1}{\sqrt{K}}\right]$ is used as random initialization for the codewords and smoothing factor. The hyperparameter $\lambda$ for the optimized model equals to $\lambda = 10^3$. We report Top-1 accuracy for the experiment in this part.

Tab. 1 illustrates the average and standard deviation of Top-1 accuracy of original and optimized Deep-TEN models on the test sets. First, we compare our Deep-TEN implementation with the original paper [ZXD17]. While we tried to be as close as possible to the settings of the paper, we believe that there is still a difference in the evaluation protocol hindering a reproduction of the results. However, we can observe that the performance of the Deep-TEN model with an optimized encoding layer (DeepTEN + DGMP) is always better than our Deep-TEN model using sum-pooling. The biggest gain appears on the FMD dataset, where the optimized Deep-TEN improves by about 3 % on average. For the other two datasets, our optimized encoding layer performs slightly better (> 1 %).

**Evaluation of the end-to-end model on Historical-WI dataset:** Similar to the implementation in [Ch19], we subdivide the original images into 400 × 400 patches with a stride of 256. We apply a canny edge detector with a threshold (2000) to judge whether these binarized patches contain sufficient amount of text. However, for color image patches, we set the value of the threshold to 1500 since the edges are less strong. For data augmentation, all subdivided patches are randomly cropped with a size of 300 × 300. We didn't apply rotation or aspect ratios changes to the patches. For the color image patches, we normalize them to have zero-mean and unit standard deviation. We report both Top-1 accuracy and mean Average Precision (mAP) for the experiments in this part since they are frequently used in the task of writer identification and retrieval.

Tab. 2: Performance comparison of DGMP model (baseline) trained on binarized images with (a) pre-trained U-Nets with either frozen weights or fine-tuned (trained using the provided color images), (b) using Deep-TEN encoding with or without integrated DGMP layer (trained on the binarized images), and (c) using the full pipeline (trained on the color images). All three runs conducted on the Historical-WI [Fi17] testset.

| | Model | Top-1 | | | | mAP | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 1st | 2nd | 3rd | Avg. | 1st | 2nd | 3rd | Avg. |
| | DGMP (baseline) | 71.2 | 70.5 | 72.8 | 71.5 | 52.0 | 51.9 | 53.9 | 52.6 |
| (a) | U-Net (frozen) + DGMP | 72.4 | 72.9 | 71.8 | **72.4** | 53.0 | 53.7 | 52.9 | **53.2** |
| | U-Net (fine-tuned) + DGMP | 71.7 | 72.6 | 71.1 | 71.8 | 52.6 | 53.4 | 51.8 | 52.6 |
| (b) | Deep-TEN | 67.9 | 69.4 | 66.4 | 67.9 | 48.6 | 50.2 | 47.8 | 48.9 |
| | Deep-TEN + DGMP | 61.1 | 57.0 | 62.9 | 60.3 | 42.1 | 38.8 | 44.2 | 41.7 |
| (c) | U-Net (frozen) + Deep-TEN + DGMP | 63.3 | 65.6 | 63.4 | 64.1 | 44.6 | 47.2 | 44.1 | 45.3 |

The triplet loss [SKP15] is employed to train our networks, where we use hard-batch online triplet selection [SKP15]. That means each mini-batch consists of $P$ writers with $K$ patches each. In our work, we use the following parameters: $P = 7$, $K = 3$, and triplet loss margin $m = 0.1$. The hyperparameter $\lambda$ for DGMP is set to $\lambda = 10^3$. We use the Adam optimizer [KB14] with a weight decay of $10^{-3}$. The learning rate is initialized with $10^{-5}$ and multiplied by 0.1 when the mAP plateaus. All experiments are run for 80 epochs, and the models with best validation accuracy are selected for the testing phase.

We have designed three experiments to evaluate the performance of our end-to-end model. The first two experiments address to evaluate the effects of each deep learning-based stage of our end-to-end network against its non-deep learning method. In the final experiment, the integrated end-to-end model is implemented and evaluated. The model proposed in [Ch19] acts as the baseline for all of our experiments in this part. The baseline model consists of fine-tuning a pre-trained ResNet-50 followed by a DGMP layer, i. e. no Deep-TEN encoding or U-Net is used. It was trained on the provided binarized images of the dataset. Note that the results differ from the ones reported in the paper [Ch19] due to a different batch size and a reduced input size.

**(a) Effect of pre-trained U-Net:** We train two models to evaluate the effects of the pre-trained U-Net. The U-Net takes a 300 × 300 input patch and outputs an equally-sized segmentation. Both models are composed of a pre-trained U-Net followed by the baseline architecture. We freeze the parameters of the U-Net in the first model while we fine-tuning the U-Net in the second model. Note that both models are trained with color images. The results are given in Tab. 2a. We see that the performance increases when the pre-trained U-Net is transferred to a writer identification network in case of frozen U-Net weights.

**(b) Effect of learnable Deep-TEN layer:** The only difference between our Deep-TEN architecture and the DGMP model (baseline) is the learnable encoding layer. In this experiment, the models are evaluated on the binarized Historical-WI dataset. Tab. 2b shows the test accuracy of the models. Both the original and the optimized encoding layer make the results worse. The pure Deep-TEN model even works better than the optimized one (Deep-TEN + DGMP). It seems that the learnable residual encoding layer does not work well in the task of writer identification. Moreover, it seems that the encoding layer does not benefit from the DGMP aggregation in this task. To this end, we can only speculate why this is the case. Maybe the Deep-TEN encoding layer encourages overfitting on the training writers and hence does not generalize well on the unseen test samples. Note that we also experimented with different learning rates, weight decay, etc.

**(c) Full model:** Finally, we evaluate the whole model, cf. Tab. 2c. While the additional binarization step improves over Deep-TEN + DGMP, the encoding stage by Deep-TEN worsened the results too much that it could possibly outperform the baseline model.

## 5 Conclusion

This paper aimed to propose a fully end-to-end deep learning-based pipeline for writer identification. To achieve this goal, we mainly have investigated two kinds of technologies, image binarization and the use of global feature encoding. The U-Net, which is pre-trained on the DIBCO dataset, works as the binarization layer in our end-to-end model. The results of our experiments show that the pre-trained U-Net outperforms the traditional method. However, a fine-tuning of the U-Net was not beneficial, thus this could also be a separate pre-processing step. Moreover, we evaluated Deep-TEN, an encoding technique to compute the global descriptor. We also incorporated DGMP aggregation mechanism. This improved encoding layer worked fine for texture classification. However, for the historical writer identification dataset, both pure and enhanced encoding layer worsen the performance. Overall, a short pipeline of just binarization, feature extraction by triplet loss and just use the weighted average by means of DGMP seems to be better than using a sophisticated encoding layer.

**Acknowledgement**: This work has been partly supported by the Cross-border Cooperation Program Czech Republic – Free State of Bavaria ETS Objective 2014–2020 (project no. 211).

## References

[Ar16] Arandjelovic, R.; Gronat, P.; Torii, A.; Pajdla, T.; Sivic, J.: NetVLAD: CNN architecture for weakly supervised place recognition. In: CVPR. Pp. 5297–5307, 2016.

[CBA15] Christlein, V.; Bernecker, D.; Angelopoulou, E.: Writer identification using VLAD encoded contour-Zernike moments. In: ICDAR. IEEE, pp. 906–910, 2015.

[Ch15] Christlein, V.; Bernecker, D.; Maier, A.; Angelopoulou, E.: Offline writer identification using convolutional neural network activation features. In: German Conference on Pattern Recognition. Springer, pp. 540–552, 2015.

[Ch17] Christlein, V.; Gropp, M.; Fiel, S.; Maier, A.: Unsupervised feature learning for writer identification and writer retrieval. In: ICDAR. Vol. 1, IEEE, pp. 991–997, 2017.

[Ch19] Christlein, V.; Spranger, L.; Seuret, M.; Nicolaou, A.; Král, P.; Maier, A.: Deep Generalized Max Pooling. In: ICDAR. IEEE, pp. 1090–1096, 2019.

[Fi17] Fiel, S.; Kleber, F.; Diem, M.; Christlein, V.; Louloudis, G.; Nikos, S.; Gatos, B.: Icdar2017 competition on historical document writer identification (historical-wi). In: ICDAR. Vol. 1, IEEE, pp. 1377–1382, 2017.

[FS15] Fiel, S.; Sablatnig, R.: Writer identification and retrieval using a convolutional neural network. In: International Conference on Computer Analysis of Images and Patterns. Springer, pp. 26–37, 2015.

[He16] He, K.; Zhang, X.; Ren, S.; Sun, J.: Deep residual learning for image recognition. In: CVPR. IEEE, pp. 770–778, 2016.

[Jé12] Jégou, H.; Perronnin, F.; Douze, M.; Sánchez, J.; Pérez, P.; Schmid, C.: Aggregating Local Image Descriptors into Compact Codes. IEEE Transactions on Pattern Analysis and Machine Intelligence 34/9, pp. 1704–1716, Sept. 2012.

[KB14] Kingma, D. P.; Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980/, 2014.

[KFS18] Keglevic, M.; Fiel, S.; Sablatnig, R.: Learning features for writer retrieval and identification using triplet cnns. In: 2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR). IEEE, pp. 211–216, 2018.

[KSH12] Krizhevsky, A.; Sutskever, I.; Hinton, G. E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. Pp. 1097–1105, 2012.

[Lo99] Lowe, D. G.: Object recognition from local scale-invariant features. In: ICCV. Vol. 2, Ieee, pp. 1150–1157, 1999.

[LSD15] Long, J.; Shelhamer, E.; Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR. Pp. 3431–3440, 2015.

[MS18] Mangalam, K.; Salzamann, M.: On compressing u-net using knowledge distillation. arXiv preprint arXiv:1812.00249/, 2018.

[Mu16] Murray, N.; Jégou, H.; Perronnin, F.; Zisserman, A.: Interferences in match kernels. IEEE transactions on pattern analysis and machine intelligence 39/9, pp. 1797–1810, 2016.

[QT09] Quattoni, A.; Torralba, A.: Recognizing indoor scenes. In: CVPR. IEEE, pp. 413–420, 2009.

[RFB15] Ronneberger, O.; Fischer, P.; Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. Springer, pp. 234–241, 2015.

[SF16] Sudholt, S.; Fink, G. A.: Phocnet: A deep convolutional neural network for word spotting in handwritten documents. In: 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR). IEEE, pp. 277–282, 2016.

[Sh13] Sharan, L.; Liu, C.; Rosenholtz, R.; Adelson, E. H.: Recognizing materials using perceptually inspired features. International journal of computer vision 103/3, pp. 348–371, 2013.

[SKP15] Schroff, F.; Kalenichenko, D.; Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: CVPR. IEEE, pp. 815–823, 2015.

[SLT10] Su, B.; Lu, S.; Tan, C. L.: Binarization of historical document images using the local maximum and minimum. In: Proceedings of the 9th IAPR International Workshop on Document Analysis Systems. Pp. 159–166, 2010.

[TM17] Tensmeyer, C.; Martinez, T.: Document image binarization with fully convolutional neural networks. In: ICDAR. Vol. 1, IEEE, pp. 99–104, 2017.

[WTB14] Wu, X.; Tang, Y.; Bu, W.: Offline text-independent writer identification based on scale invariant feature transform. IEEE Transactions on Information Forensics and Security 9/3, pp. 526–536, 2014.

[Yo14] Yosinski, J.; Clune, J.; Bengio, Y.; Lipson, H.: How transferable are features in deep neural networks? In: Advances in neural information processing systems. Pp. 3320–3328, 2014.

[ZXD17] Zhang, H.; Xue, J.; Dana, K.: Deep ten: Texture encoding network. In: CVPR. IEEE, pp. 708–717, 2017.