Writer Identification and Writer Retrieval Based on NetVLAD with Re-ranking

Shervin Rasoulzadeh and Bagher Babaali

School of Mathematics, Statistics and Computer Science, College of Science, University of Tehran, Tehran, Iran
{s.rasoulzadeh,babaali}@ut.ac.ir

Abstract. This paper addresses writer identification and writer retrieval which is considered as a challenging problem in the document analysis and recognition field. In this work, a novel pipeline is proposed for the problem at hand by employing a unified neural network architecture consisting of the ResNet-20 as a feature extractor and an integrated NetVLAD layer, inspired by the vector of locally aggregated descriptors (VLAD), in the head of the latter part. Having defined this architecture, the triplet semi-hard loss function is used to directly learn an embedding for individual input image patches. Subsequently, generalized max-pooling technique is employed for the aggregation of embedded descriptors of each handwritten image. Also, a novel re-ranking strategy is introduced for the task of identification and retrieval based on $k$-reciprocal nearest neighbors, and it is shown that the pipeline can benefit tremendously from this step. Experimental evaluation has been done on the three publicly available datasets: the ICDAR 2013, CVL, and KHATT datasets. Results indicate that while we perform comparably to the state-of-the-art on the KHATT, our writer identification and writer retrieval pipeline achieves superior performance on the ICDAR 2013 and CVL datasets in terms of mAP.

Keywords: Writer Identification · Writer Retrieval · NetVLAD · Re-ranking · Document Analysis · Deep Learning

1 Introduction

Along with biometrics identifiers such as DNA, fingerprints, etc, handwriting is considered as a special case of behavioral biometrics [36]. Handwriting analysis helps to extract attributes such as writer from a handwritten document. Several factors may lead to handwriting variability such as having diseases (Parkinson could be recalled as an example), education, time and effort spent, different pens, etc [11]. Thus, Handwriting analysis is considered a complex and challenging task. To overcome these challenges and to provide an automatic handwriting analysis system, one needs to differentiate between online and offline data/systems. Online text analysis systems capture the whole procedure of writing with special devices and the input consists of temporal data such as pen-tip
positions. On the other hand, offline data is usually static and typically is in the format of an image. Also, methods for handwriting analysis are categorized into two categories: text-dependent and text-independent methods. In the text-dependent methods, each handwriting must contain a fixed content, while in the text-independent methods no assumptions are made on the content of handwriting and any arbitrary text could constitute the handwriting. In this work, it is intended to provide an offline text-independent handwriting analysis system concentrated on the problem of writer recognition, specifically writer identification and writer retrieval.

Writer retrieval is the task of ranking document images from a large corpus of data with similar handwriting to the query sample, see Figure 1. Experts in relative fields then analyze these rankings and thus new documents from the same writer can be found. Historians and paleographers benefit the most from this scenario. When analyzing historical documents, a vast amount of data should be dealt with, where examining them individually is not possible or a very time-consuming task. However, this scenario helps to find the writer of a historical document in a shortlist without having to go through all documents in the database. In contrast to writer retrieval, writer identification is the task of finding the writer of a query sample assuming a set of documents where each one’s writer is known in advance, see Figure 2. This scenario is often applicable in forensic sciences, e.g., finding the writer of a threatening letter. The methods for the two scenarios are quite similar. Both generate a feature vector describing the handwriting of each document with respect to its writer. These feature vectors are compared using a similarity measure criterion such as cosine distance, $\chi^2$-distance, etc, and then accordingly rankings are computed. For the retrieval scenario, these rankings are further analyzed and a shortlist of documents written by the query’s writer will be returned. On the other hand, for the identification scenario, the writer of the sample with the shortest distance to the query sample is returned as output.

Our proposed pipeline uses a ResNet-20 convolutional neural network with the NetVLAD layer, inspired by Vectors of Locally Aggregated Descriptors.
Fig. 2: Overall schema of writer identification.

(VLAD), to extract local descriptors and their corresponding embeddings describing small windows in a document image. Afterward, generalized max-pooling (GMP) aggregation is employed to achieve a single global image descriptor for each image from the computed embeddings. Dimensionality reduced global image descriptors resulted from applying the PCA, are then compared and rankings are computed. In the final stage, we make use of a re-ranking strategy based on query expansion and $k$-reciprocal nearest neighbors to improve the retrieved rankings before evaluation.

The structure of this work is as follows: In Section 2 some deep-learning-based related work in the field of writer identification and retrieval, as well as re-rankings, are discussed. Afterwards, in Section 3 we investigate our proposed pipeline in great detail. Section 4 introduces datasets and finally, we evaluate and compare our proposed pipeline against state of the art on three different datasets (ICDAR 2013, CVL, and KHATT) in Section 5.

2 Related Work

Nearly all common writer identification and retrieval datasets consist of writer disjoint train and test sets. Hence, end-to-end training cannot be applied.

One of the first writer recognition methods using deep learning techniques was proposed by Fiel and Sablatnig [16]. They trained the "CaffeNet" CNN on the line and word segmentations. Feature vectors extracted from the penultimate layer of the CNN are compared using the $\chi^2$-distance. At the time, their results showed superior performance on the IAM [31] and ICFHR’12 datasets while being inferior on the ICDAR 2013 [28] dataset. Christlein et al [6] used activation features from a CNN as image local descriptors. Afterward, global image descriptors are formed through the GMM supervector encoding. Their approach improved 0.21 % in terms of mAP on the ICDAR13 dataset. In another recent method by Christlein et al [9] LeNet and ResNet architectures are employed to extract local descriptors followed by VLAD encoding to compute global image descriptors for document images. They experimented with both exemplar support vector machines (ESVMs) and nearest neighbors to evaluate their pipeline.
To the best of our knowledge, their approach has set new standards on the ICDAR 2013 and CVL [25] datasets.

In [24] Jordan et al. experimented with reciprocal relationships in two ways. First, integrated them into the Jaccard distance and computed the final rankings based on a weighted sum of the original distance and the Jaccard distance. Second, encoded them into new feature vectors and hence expanded the positive set for ESVMs. As a result, both of their techniques outperformed the baseline on the ICDAR 2017 dataset [17].

Tang and Wu [39] proposed a novel approach with convolutional neural network (CNN) and joint Bayesian consisting of two stages: 1. feature extraction and 2. writer identification. They used CNNs to extract global features instead of small image patches. They used random word segmentations and generated 500 and 20 training samples per writer for training and testing, respectively. Finally, a Bayesian network is used for the computation of similarity between feature vectors. At the time, they achieved the best results compared to the state-of-the-art on ICDAR 2013 and CVL datasets. In another work by Xing and Qiao [41], two adjacent images patches used as inputs to their proposed network, named Deep-Writer, consisting of two branches sharing the convolutional Layers. For the final evaluation part, two softmax layers belonging to each branch were averaged to predict the writer and achieved promising results on the IAM dataset. Despite that they worked in an end-to-end manner (i.e. “the CNN is trained for a specific writer on a line-basis using some lines for training, one for validation and one for testing.” [9]), comparison of their work with other pipelines in literature is impossible.

Considering that, we could say that our proposed pipeline is mostly inspired by the works of Christlein et al. [6,9] and Jordan et al. [24]. However, with our proposed pipeline consisting of the unified neural network architecture with the NetVLAD layer, and re-ranking strategy based query expansion and \( k \)-reciprocal nearest neighbors, we could improve upon the state of the art on the ICDAR 2013 and CVL datasets.

3 The Proposed Pipeline for Writer Identification and Writer Retrieval

Our proposed pipeline consists of two parts: 1. A unified neural network architecture with ResNet-20 [20,21] and the NetVLAD layer [2], and 2. A re-ranking strategy to improve the final results. The first part itself consists of three main steps (depicted in Figure 3): The ResNet-20 with the NetVLAD layer to extract the local image descriptors and their corresponding embeddings, An orderless aggregation function to the pool obtained embeddings of each image into one global image descriptor, and the normalization and PCA [40] based dimensionality reduction of the resulted global image descriptors.
3.1 Convolutional Neural Network with NetVLAD Layer

State-of-the-art deep-learning-based methods in writer identification and writer retrieval usually employ a CNN to extract the local image descriptors which are subsequently encoded using an encoding method. An encoding consists of two steps: 1) An embedding step, where local feature vectors are projected into a possibly high-dimensional space, and 2) An aggregation step, in which embedded local feature vectors of each input image are pooled into one global image descriptor. Christlein et al. [9] computed the local feature vectors through ResNet-20 residual neural network and used the VLAD [23] encoding method for embedding and aggregation. Building on the success of Christlein et al. [9] we propose a unified Neural network consisting of ResNet-20 followed by the trainable NetVLAD layer [2], inspired by the VLAD, at the head of the last convolutional layer of ResNet-20 to learn the embedding of feature vectors in an end-to-end manner using a triplet loss [37].

ResNet-20 and NetVLAD Details of the ResNet-20 convolutional neural network and the NetVLAD layer are described in the following.

ResNet-20 Convolutional Neural Network. For network inputs, 32 × 32 image patches centered at the contour of handwriting were extracted. Same as Christlein et al. [9], we follow the architectural design of He et al. [20] on CIFAR10 dataset [26]. 6n + 2 layers are employed with n set to 3 leading to the ResNet-20 architecture. The first layer is 3 × 3 convolutions. Then an stack of 6n layers with 3 × 3 convolutions follows with every 2n layers forming an stage. At the beginning of each stage (except the first one), the feature map size is halved (downsampled) by a convolutional layer with the stride of 2, while the number of filters is doubled. Within each stage, the layers have the same number of filters. More precisely, feature maps and filters for stages are of sizes {32, 16, 8} and {16, 32, 64}, respectively. Shortcut connections are connected to the pairs of 3 × 3 layers leading to a total 3n shortcuts. The network ends with the global average pooling layer with a size of 8 and an N-way fully connected layer. However, We discard the last fully-connected layer and pass the 1 × 1 × 64 output feature vector of global average pooling layer to the NetVLAD to further learn the VLAD embeddings of these feature vectors. (See Figure 4).
NetVLAD Layer. The idea behind the vectors of locally aggregated descriptors (VLAD) \cite{Arandjelovic2014} is to compute the embeddings by means of residuals \(x_i - c_j\) for each local image descriptor \(x_i\). Finally, embedded local image descriptors of each image are accumulated by an orderless aggregation function. This characterizes the distribution of the vectors with respect to the cluster centers. The VLAD embedding can be regarded as a simplified version of the Fisher Vectors \cite{Arandjelovic2014}. More precisely, given \(N\) local image descriptors \(\{x_i | x_i \in \mathbb{R}^D, i = 1, \ldots, N\}\) and a dictionary of \(K\) cluster centers \(\{c_j | c_j \in \mathbb{R}^D, j = 1, \ldots, K\}\), the VLAD embedding function is computed as follows:

\[
\phi_{\text{VLAD},k}(x_i) = \alpha_k(x_i)(x_i - c_k) \quad (1)
\]

\[
\alpha_k(x_i) = \begin{cases} 
1, & \text{if } k = \arg\min_{j=1,...,K} \|x_i - c_j\| \\
0, & \text{else} \end{cases} \quad (2)
\]

Then \(\phi_{\text{VLAD}}(x_i) = (\phi_1(x_i), \ldots, \phi_K(x_i))\) represents the full embedding for each local image descriptor \(x_i\).

Arandjelovic et al \cite{Arandjelovic2014} introduced a trainable generalized VLAD layer, named NetVLAD, which is pluggable into any CNN architecture. Clusters hard-assignments \(a_k(x_i)\) of local image descriptors in the original VLAD are the source of discontinuities which prevent differentiability in the backpropagation procedure. The authors replaced them with soft-assignment to make it amenable to backpropagation:

\[
\tilde{\alpha}_k(x_i) = \frac{e^{-\alpha \|x_i - c_k\|^2}}{\sum_{j=1}^K e^{-\alpha \|x_i - c_j\|^2}} \quad (3)
\]

where \(\alpha\) is a parameter that control the decay of response with the magnitude of distance. Intuitively, Equation (3) assigns the weights of local image descriptors \(x_i\) proportional to their nearness to clusters \(c_j\). Moreover, factorizing \(e^{-\alpha \|x_i\|^2}\) results in:

\[
\tilde{\alpha}_k(x_i) = \frac{e^{w_k^T x_i + b_k}}{\sum_{j=1}^K e^{w_j^T x_i + b_j}} \quad (4)
\]

where \(w_k = 2\alpha c_k\) and \(b_k = -\alpha \|c_k\|^2\). However, in \cite{Arandjelovic2015} the authors propose decoupling dependencies of parameters \(c_k, w_k,\) and \(b_k\) as it will brings greater flexibility to the model. In this manner, NetVLAD layer consists of three independent set of learnable parameters. We crop the ResNet-20 at the last convolutional layer and view it as a \(D\)-dimensional (here \(D = 64\)) local image descriptor. As depicted in Figure 4 the NetVLAD layer can be decomposed into CNN layers connected in an acyclic graph. Equation (4) represents the soft-max activation function. So the soft-assignments of local image descriptor \(x_i\) to clusters \(c_k\) can be viewed as applying a \(1 \times 1\) convolution layer with \(K\) filters representing \(w_k\) and biases as \(b_k\) followed by the soft-max activation function to obtain final soft-assignments \(\tilde{\alpha}_k(x_i)\). The final output is \(K \times D \times 1\)-dimensional representing the full embedding for local image descriptor \(x_i\).

The authors in \cite{Arandjelovic2015} regard the output of the last convolutional layer with \(H \times W \times\)
$D$ map as a set of $D$-dimensional descriptors at $H \times W$ spatial locations in input image which are further embedded and pooled by the NetVLAD layer. However, by using ResNet-20 with image patches of size $32 \times 32$ as feature extractor, the output of the last convolutional layer becomes $1 \times 1 \times 64$ map which we consider as 64-dimensional local image descriptor extracted from the input image patch. Passing this descriptor (with $H = 1$ and $W = 1$) enables NetVLAD layer to learn the respective local image descriptor embedding. So in this manner, the NetVLAD layer functions to learn the generalized VLAD embeddings. Prior to forwarding the embeddings extracted from the NetVLAD layer to the triplet loss, they are $\ell_2$-normalized.

**Learning from Semi-Hard Triplets.** A quite well strategy to learn the parameters of the network to reach a good encoding is through triplet loss function. We wish to learn VLAD embedding representation $\phi_{\text{VLAD}}(x)$ constrained to lie on $K \times D$-dimensional hypersphere, i.e. $\|\phi_{\text{VLAD}}(x)\| = 1$, such that two embeddings belonging to the image(s) of the same writer be close together in the embedding space while embeddings of images with different writers lie far away from each other. The only requirement is that given two positive embeddings of the same writer and one negative embedding, the negative should be farther away than the positive by some margin $m$. This requirement can be translated into a loss between triplets. The loss will be defined over triplets of embeddings: an anchor $\phi_a$, a positive of the same writer as the anchor $\phi_p$, and a negative of a different writer $\phi_n$. For some distance on the embedding space $d$, the loss of a triplets $(\phi_a, \phi_p, \phi_n)$ is:

$$
\mathcal{L} = \max(d(\phi_a, \phi_p) - d(\phi_a, \phi_n) + \text{margin}, 0)
$$

(5)

The original NetVLAD paper utilizes the weakly supervised triplet ranking loss [2]. However, Since here the NetVLAD layer is applied to learn embeddings in a patch-wise manner, another strategy is employed. Based on the definition of loss we tend to train on semi-hard triplets [37], triplets where the negative is not closer to the anchor than the positive, but which still produce positive loss:

$$
d(\phi_a, \phi_p) < d(\phi_a, \phi_n) < d(\phi_a, \phi_p) + \text{margin}.
$$

(6)

We train parameters of the proposed pipeline on a large set of semi-hard triplets image patches triplets extracted from the respective dataset. Details and parameters of training are given in Section 4.

**Aggregation** Aggregation step is required to obtain a single vector representing each image from its embedded local descriptors. Default aggregation method is *sum-pooling*. Assuming the set of $N$ local descriptors $\mathcal{X} = \{x_i | i = 1, \cdots, N\}$ for an image, sum-pooling constructs global descriptor $\xi$ as follows:

$$
\xi = \psi(\phi(\mathcal{X})) = \sum_{x \in \mathcal{X}} \phi(x).
$$

(7)
Fig. 4: CNN architecture composed of the ResNet-20 followed by the NetVLAD layer. Numbers in each rectangle denote kernel size, number of output filters, padding, and size of stride, respectively. The $N$-way fully connected is dropped and instead the $1 \times 1 \times 64$ output vector is passed to the NetVLAD layer.

"Since we sum over all descriptors, the aggregated descriptors can suffer from interference of unrelated descriptors that influence the similarity, even if they have low individual similarity" [10] as the similarity $K(\mathcal{X}, \mathcal{Y})$ between two images represented by sets $\mathcal{X}$ and $\mathcal{Y}$ is computed as follows:

$$K(\mathcal{X}, \mathcal{Y}) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} \phi(x) \cdot \phi(y). \quad (8)$$

Hence, more frequently occurring descriptors will be more influential in the final representation and affect the final similarity between global descriptors. This phenomenon is called visual burstiness [22]. Recently, a novel approach named generalized max-pooling [32] was proposed to overcome this problem and has successfully applied in the field of writer identification and writer retrieval in works of Christlein et al. [9]. We employed this method in our pipeline as it has shown superior performance to the other two methods [33]. Generalized max-pooling balances contribution of every embedding $\phi(x) \in \mathbb{R}^{K \times D}$ where $x \in \mathbb{R}^D$. 
is local image descriptor, by solving a ridge regression problem. Therefore,

$$\phi(x)^T \xi_{gmp}(x) = C, \quad \forall x \in \mathcal{X},$$

(9)

where $\mathcal{X}$ is the set of all local descriptors of an image, $\xi_{gmp}$ denotes aggregated global image descriptor and $C$ is a constant that can be set arbitrarily since it does not influence the global image descriptors since they are subsequently normalized in the post-processing step. Equation (9) can be re-formulated for all $N$ local image descriptors of each image as below:

$$\Phi^T \xi_{gmp} = 1_N,$$

(10)

where $\Phi$ and $1_N$ denote the $(K \times D) \times N$ matrix of all local image descriptors embeddings and vector of $N$ constants set to 1, respectively. Equation (10) can be turned into a least-squares ridge regression problem [9,11]:

$$\xi_{gmp} = \arg\min_{\xi} \|\Phi^T \xi - 1_N\|_2^2 + \lambda \|\xi\|_2^2$$

(11)

with $\lambda$ being a regularization parameter. In the remainder of this work, $\psi$ denotes the aggregated global image descriptor.

**Normalization and Dimensionality Reduction** While working with global image descriptors obtained in the previous step, two challenges arise: (1) visual burstiness might corrupt visual similarity measure between the global image descriptors, i.e. the cosine distance used to rank images, and (2) these global descriptors lie in a very high-dimensional space and pipeline might benefit from projecting them to a lower-dimensional space. We address these challenges with an additional normalization and dimensionality reduction step.

**Power Normalization.** A normalization method to counter visual burstiness is named **power normalization** [35] that proposes to apply function $f$ component-wise to global image descriptor $\psi$,

$$f(\psi) = \text{sign}(\psi_i)|\psi_i|^p, \quad \forall i, 1 \leq i \leq n$$

(12)

where $p$ is a normalization parameter and is generally set to 0.5. Power normalization is followed by $\ell_2$-normalization.

**Principal Component Analysis.** Due to the nature of VLAD encoding, global image descriptors lie in a very high-dimensional space. Principal component analysis (PCA) [40] is used to dimensionality reduce the encoding representations. However, this introduces a new parameter, dimension, to the pipeline denoting the number of components to keep. After performing the PCA, $\ell_2$-normalization along each sample is necessary.
3.2 Re-ranking

Writer identification and retrieval systems are evaluated using leave-one-image-out cross validation. Each image is once used as query $q$ and the pipeline returns a ranked list $L(q)$ of all other images in test set (a.k.a. gallery). These ranked lists are obtained by computing the pairwise distance between query $q$ and each $p \in L(q)$ using a similarity measure criterion, i.e. cosine distance. Given two vectors $p$ and $q$, the cosine distance is defined as:

$$d_{cos}(p,q) = 1 - \frac{pq}{\|p\|\|q\|}.$$  (13)

Our goal is to re-rank each $L(q)$ based on knowledge lied in it, so that more relevant samples rank top in the list and thus, boost the performance of writer identification and retrieval.

**Nearest Neighbors** $k$-nearest neighbors $k\text{NN}(q)$ (top-$k$ ranked samples of ranked list) of query $q$ is defined as:

$$k\text{NN}(q) = \{p_1, p_2, \cdots, p_k\}, \quad |k\text{NN}(q)| = k,$$  (14)

Where $|$ denotes the cardinality of the set. The $k$-reciprocal nearest neighbors $kr\text{NN}(q,k)$ is defined as:

$$kr\text{NN}(q,k) = \{p_i | p_i \in k\text{NN}(q) \land q \in k\text{NN}(p_i)\}.$$  (15)

In other words, two samples $q$ and $p$ are considered as $k$-reciprocal nearest neighbors, when both appear within the top-$k$ ranked samples of each other. According to the previous descriptions, $k$-reciprocal nearest neighbors are more related to query $q$ than $k$-nearest neighbors.

**Query Expansion and $k$-Reciprocal Nearest Neighbors** A common approach to boost the performance of the information retrieval systems is automatic query expansion (QE) [24]. With an initial ranked list $L(q)$ computed, query expansion reformulates each query sample $q$ and obtains the improved ranked list by re-quering using the newly formed query instead of $q$.

Chum et al [12] proposed the following query expansion approach. For query $q$, a new query sample can be formed by taking the average over top-$n$ spatially verified samples $F$ from ranked list $L(q)$,

$$q_{avg} = \frac{1}{|F| + 1} \left( q + \sum_{f \in F} f \right)$$  (16)

where $f_i$ and $n$ denote the $i$th sample in $F$ and total number of samples in $|F|$, respectively.

In our problem, the features do not encode any global spatial information and
thus, we have no spatial verification at hand. Averaging over top-\(k\) samples in \(L(q)\) is not much reliable since the top-\(k\) samples might contain false matches. We propose to use a more constrained strategy by taking an average over query \(q\) and its \(k\)-rNNs in the initial ranked list to minimizing the risk of including false matches. Hence, the newly formed query \(q_{\text{new}}\) is computed as follows:

\[
q_{\text{new}} = \frac{1}{|kr\text{NN}(q)| + 1} \left( q + \sum_{r \in kr\text{NN}(q)} r \right).
\]

(17)

This however introduces a new hyper-parameter \(k\) to the pipeline. In the following, our proposed pipeline(with re-ranking) is denoted as ”Proposed(+krNN-QE\(_k\)).”

4 Evaluation

4.1 Datasets

Our primary dataset is the ICDAR 2013. However, we compare our results against the state-of-the-art on the CVL and KHATT datasets as well. These datasets contain documents written in English, Greek, German, and Arabic. Successful results on these datasets also show that the proposed pipeline is language independent.

**ICDAR 2013** : The ICDAR 2013 dataset was introduced for the ICDAR 2013 competition on writer identification [28]. It consists of four samples per writer, two of which are written in English, while the two others are in Greek. This dataset contains two disjoint train (a.k.a experimental) and test (a.k.a benchmarking) subsets. The train set consists of 100 writers, while the test set contains 250 writers. An example document from ICDAR 2013 dataset can be seen in Figure 5.

**CVL** : The CVL dataset [25] (version 1.1) consists of 27 writers contributing to seven texts (one in German and six in English) in the officially provided train set. The test set consists of 283 writers, where each writer has copied five texts (one in German and four in English). Figure 6 depicts one sample document from this dataset.

**KHATT** : The KHATT dataset [30] used as a database for ICFHR 2014 [38] Arabic writer identification competition. It consists of 1000 writers contributing to four samples. Two of which contain similar texts, while the other two were unique to each writer. The dataset is provided by three disjoint, i.e. writer independent sets, with a 4: 1: 1 ratio for train, test, and validation subsets, respectively. Along with two other datasets, a sample document image of this dataset is shown in Figure 7.
4.2 Metrics

Results reported in terms of the Hard Top-$N$ and mAP which are quite common metrics in information retrieval tasks. These metrics defined in the following.

**Hard Top-$N$.** The strictest evaluation metric is hard top-$N$. A returned list $L(q)$ for query sample $q$ is considered as acceptable if all of the top-$N$ ranked samples in $L(q)$ belong to the same class as sample $q$'s class i.e., written by same writer. The ratio of the number of acceptable returned lists and the number of query samples is reported as Hard Top-$N$ accuracy.

**Mean Average Precision.** Another commonly used measure to evaluate an information retrieval task is the mean average precision (mAP) which considers the ranking of correct samples. It is calculated as the mean over all examined query samples $q$ of set $Q$:

$$
mAP = \frac{\sum_{q \in Q} AveP(q)}{|Q|},
$$

where $AveP(q)$ is the average precision for a given query $q$ defined as below:

$$
AveP(q) = \frac{\sum_{k=1}^{n} (P(k) \times rel(k))}{\text{number of relevant documents}},
$$

where $n$ is the total number of retrieved samples, $rel(k)$ is a binary function returning 1 if sample at rank $k$ of $L(q)$ is relevant and 0 otherwise, and $P(k)$ is the precision at rank $k$ (fraction of relevant items up to first $k$ retrieved samples in $L(q)$).
4.3 Experiments and Results

Most datasets in the field of writer identification and writer retrieval come with disjoint train and test subsets and because of that, an end-to-end procedure would not be applicable. Therefore, our pipeline is composed of two phases: (1) training phase and (2) testing phase, each described below.

Training phase. To train and validate the neural network, 32 × 32 patches centered on the contour of handwritten images in the ICDAR 2013 train set get extracted. We sample around 5000000/25000 image patches for train/validation where they are subsequently passed forward to the network. ResNet weights are initialized by He-initialization [19] and Xavier-initialization [18] used to initialize Conv layer of the NetVLAD. As the ICDAR 2013 train set consists of 100 writers, the number of cluster centers has been set to 100 in the NetVLAD layer. The proposed neural network is optimized using Adamax with respect to triplet semi-hard loss with margin \( m = 0.1 \), decay rates \( \beta_1 = 0.9 \) and \( \beta_2 = 0.99 \) for 1st moment estimate and exponentially weighted infinity norm, respectively. Training is stopped after 5 epochs since the loss value stagnated at this point. The learning curves visualization is depicted in Figure 8.

Testing Phase. Once the proposed network trained, we pass image patches of the test set of the dataset under evaluation to obtain embedded feature vectors where they are eventually pooled to compute global image descriptor of each image. For generalized max-pooling, following the works of Christlein et al. [9] we set \( \lambda = 1000 \). The next step after normalization (\( \ell_2 \) and SSR applied in order) is dimensionality reduction through PCA. PCA used for whitening and projecting global descriptors to a user-defined number of dimensions (Figure 9 shows the mAP for the different number of dimensions on ICDAR 2013 train and test subsets) after which they subsequently get \( \ell_2 \)-normalization. The final step is following the evaluation procedure, i.e (re-)ranking documents based on their similarity. To have a clear view of how the proposed pipeline can benefit from re-ranking, Table 1 provides a comparison between the proposed pipeline with initial rankings (denoted as ”ResNet-20+NetVLAD+NN”) and re-ranking with different values for \( k \) (denoted as ”ResNet-20+NetVLAD+krNN-QE_k”) on the ICDAR 2013 test set. As observations reveal that \( \text{dimension} = 128 \) and \( k = 2 \) give the best results on the test set of this dataset, we follow our evaluation with these two parameters set.
Fig. 8: (a) Training learning curve in terms of batch number, and (b) based on epoch number. Orange and Blues denotes train and validation curves, respectively.

Fig. 9: Comparison of effect of number of components in PCA in terms of mAP on the ICDAR 2013 (a) train and (b) test subsets.

Table 1: Comparison of evaluation with initial rankings against re-ranking with different values of $k$ on ICDAR 2013 test set.

| Model                        | Top-1 | Hard-2 | Hard-3 | mAP  |
|------------------------------|-------|--------|--------|------|
| ResNet-20+NetVLAD+NN         | 98.60 | 84.10  | 65.60  | 93.01|
| ResNet-20+NetVLAD+krNN-QE$_k=1$ | **98.70** | 90.30  | 86.70  | 96.48|
| ResNet-20+NetVLAD+krNN-QE$_k=2$ | 97.90 | 90.50  | **86.80** | **96.58** |
| ResNet-20+NetVLAD+krNN-QE$_k=3$ | 96.40 | 91.00  | 83.50  | 96.36|

**Visualization of Embeddings** t-SNE [29] is an unsupervised, non-linear technique used mostly for exploration and visualization of high-dimensional data. We perform t-SNE on the ICDAR 2013 train subset to get an intuition of how the computed and dimensionality reduced global descriptors are arranged in the 128-D space. The t-SNE plot of the embeddings space is shown in Figure 10. The plot shows that the embeddings learned by the proposed pipeline probably have very well discriminative properties as all of the four global descriptors representing each writer’s document images approximately lie near each other in the projected 2-D space then forming a cluster. In other words, the nearest neigh-
bors of each global image descriptor are likely to form the same writer in the original space. On the other hand, Figure 11 depicts 1000 global image descriptors of dimension 128 of the ICDAR 2013 test set before and after the re-ranking part. As the Figure illustrates, the discriminability of clusters is increased and the clusters belonging to each writer are more distinguishable from each other compared to without re-ranking employed.

Fig. 10: t-SNE plot of the ICDAR 2013 train set global descriptors. There are 100 colors each representing document images of unique writer.

**Comparison with State of the Art** To compare our results to the state-of-the-art writer recognition pipelines, we deliberate on the three contemporary datasets (ICDAR 2013, CVL, and KHATT). Results are given in terms of Top-1, Hard-2, Hard-3, and mAP evaluation metrics. The Top-1 metric demonstrates the probability of first retrieved items belonging to the same writer as the query. While Hard-2 and Hard-3 are the probabilities representing that all the first two and first three retrieved documents stem from the same writer as the query’s, respectively. In most scenarios, the mAP criterion is a more proper complement since it considers the ranking of all the documents. The ResNet-20 convolutional neural network with the NetVLAD layer is employed to extract local descriptors and their corresponding VLAD embeddings. Once embeddings are computed, generalized max-pooling is used to form a global descriptor for each document image. $\ell_2$-normalization, SSR, and PCA are subsequently applied to the global descriptors. Note that in all experiments we trained ResNet-20+NetVLAD on the ICDAR 2013 train subset, and used the trained network to evaluate each dataset’s test set. In the following, we first summarize our results on these datasets, and secondly, the impression of different parts of the pipeline on the obtained results are investigated. Finally, we also present the overall computational cost of three respective datasets.

ICDAR 2013. The document images in this dataset are already in binarized format out-the-box. We train and test and the officially provided train and test subsets of this dataset. Comparison of our results on the ICDAR 2013 dataset
is provided in the first section of Table 2. Our pipeline with the proposed re-ranking strategy achieves the overall best result (in terms of mAP) with a 4.2% difference against the previous best [9] reported. Another interesting observation is that the Hard-3 criterion is improved by 13.3%, which is an indicator of the huge benefit that re-ranking brings to the pipeline. This can also be observed on the Hard-2 criterion where around 3.0% improvement has been achieved. However, the slightly inferior performance obtained in terms of Top-1 which opens the room for further investigation. All in all, comparing results show that our proposed pipeline outperforms all the previous work in this field. Although our pipeline shares some similarities with the pipeline of Christlein et al. [9], our results are far better due to the direct integration of the encoding layer to the ResNet-20, and the re-ranking step.

CVL. Since the official train subset of this dataset is rather small (189 samples), we employed the already trained pipeline on the ICDAR 2013 train set and used the official test subset of the CVL dataset containing the subset of writers where each contributed exactly five forms (CVL-283) for evaluation. Also, we binarized document images using Otsu’s method [34] to be more similar to the ICDAR 2013 dataset. The obtained results along with the state-of-the-art counterparts provided in the second section of Table 2. Our proposed approach with re-ranking sets new standards by showing superior performance in terms of Hard-3 and
Table 2: Results and comparison with state-of-the-art on the evaluation datasets. Articles are sorted by their publication date.

| Dataset | Method                                  | Top-1 | Hard-2 | Hard-3 | mAP  |
|---------|-----------------------------------------|-------|--------|--------|------|
| ICDAR 2013 | RootSIFT+SV+NN [5]                       | 97.1  | 42.8   | 23.8   | 67.1 |
|         | Contour-Zernike+VLAD+NN [7]              | 99.4  | 81.0   | 61.8   | 88.0 |
|         | LeNet-5+SV+NN [6]                        | 98.9  | 83.2   | 61.3   | 88.6 |
|         | CaffeNet+NN [16]                         | 88.5  | 63.2   | 36.5   | -    |
|         | RootSIFT+SV+ESVM [8]                     | 99.7  | 84.8   | 63.5   | 89.4 |
|         | ResNet-20+VLAD+ESVM [9]                  |       |        |        |      |
|         | ResNet-20+NetVLAD+krNN-QE\(k=2\)         | 97.4  | 92.8   | 90.3   | 97.4 |
| CVL     | RootSIFT+SV+NN [5]                       | 99.2  | 98.1   | 95.8   | 97.1 |
|         | Contour-Zernike+VLAD+NN [7]              | 99.4  | 98.9   | 97.4   | 97.9 |
|         | LeNet-5+SV+NN [6]                        | 99.4  | 98.8   | 97.3   | 97.8 |
|         | CaffeNet+NN [16]                         | 88.5  | 63.2   | 36.5   | -    |
|         | RootSIFT+SV+ESVM [8]                     | 99.2  | 98.4   | 97.1   | 98.0 |
|         | ResNet-20+VLAD+ESVM [9]                  | 99.5  | 99.0   | 97.7   | 98.4 |
|         | ResNet-20+NetVLAD+krNN-QE\(k=2\)         | 99.2  | 98.9   | 98.0   | 98.6 |
| KHATT   | RootSIFT+SV+NN [5]                       | 99.3  | 96.8   | 94.5   | 97.5 |
|         | Contour-Zernike+VLAD+NN [7]              | 99.4  | 98.9   | 97.4   | 97.9 |
|         | RootSIFT+SV+ESVM [8]                     | 99.5  | 96.5   | 92.5   | 97.2 |
|         | ResNet-20+VLAD+ESVM [9]                  | 99.6  | 97.6   | 94.5   | 98.0 |
|         | ResNet-20+NetVLAD+krNN-QE\(k=2\)         | 98.6  | 95.3   | 93.0   | 97.7 |

mAP and can compete with the state-of-the-art, but slightly behind in terms of Top-1 and Hard-2 metrics.

**KHATT.** The last dataset that we were experimented on is the KHATT dataset. Likewise the CVL dataset, document images are binarized with the Otsu method. The respective results are depicted in the third section of Table 2. We believe that our evaluation strategy on this dataset is slightly different from the state-of-the-art, as they (probably) used the official train set of the respective dataset. Though likewise the CVL dataset, we used the trained network on the ICDAR 2013 officially provided train split and the officially provided test split (600 images from 150 writers) of the respective dataset served as our test set. It is quite interesting that despite training on a dataset containing English and Greek documents, the results are quite promising and comparable to the state-of-the-art.

**Influence of Different Parts of the Pipeline** Most of the writer recognition pipelines are made out of three major parts: (1) Feature Extraction, (2) Encoding, and (3) ranking strategy. As we have demonstrated in Table 3 we have specified each method name by these three parts.

**Feature Extraction and Encoding.** Feature Extraction and Encoding steps are two inseparable parts of the writer recognition pipelines. In contrast to the nu-
merous hand-crafted feature methods such as Contour-Zernike and SIFT descriptors, we use features learned by a convolutional neural network. Relying on the neural networks brings the advantage of being data-driven. As data progresses through the CNN layers, a higher level of abstraction is reached automatically by the data. Also, the encoding step is required for the pipeline to computing a single representation for each document image from its many feature descriptors. There are a few encoding methods such as GMM Supervectors, VLAD, I-Vectors, etc. used in the literature where specifically VLAD encoding results were quite promising. However, none of them are deep-learning-based and we believe that constructing a unified neural network consisting of both the feature extraction and encoding step as one integrated step would bring many advantages to the system. One of the first deep-learning-based approaches is implemented by Fiel et al. [16] employed the CaffeNet which is part of the ”Caffe Deep Learning Framework”. However, they trained their convolutional neural network on the word images of the IAM dataset and extracted the features of the penultimate fully connected layer for evaluation. But they did not use any specific encoding method and this might a reason that their approach is far behind the state-of-the-art methods using an encoding step. Christlein et al. [9] already used ResNet-20 as a feature descriptor, However, we also integrate VLAD encoding step into the neural network by employing the NetVLAD layer in the head of ResNet-20. In this manner, We have one unified neural network to extract embeddings from each document image. To combine Feature extraction and encoding in one part. Table 3 compares our unified feature extraction and encoding step with others in the literature.

Table 3: Effect of feature extraction and encoding step of the pipeline on the ICDAR13 test set.

| Method                    | mAP  |
|---------------------------|------|
| RootSIFT+SV               | 67.1 |
| Contour-Zernike+VLAD      | 88.0 |
| LeNet-5+SV               | 88.6 |
| ResNet-20+VLAD            | 93.2 |
| ResNet-20+NetVLAD         | 94.1 |

**Ranking Strategy.** Routinely, in the leave-on-image-out cross-validation, the ranking of other documents are computed based on cosine similarity with the nearest neighbor approach. However, performance could be leveraged by the knowledge provided in this initial ranking in order to refine them. While it is quite common in image retrieval methods to use re-ranking strategies, It has not brought much attention to it in the field of writer identification and writer retrieval. To the best of our knowledge, Jordan et al. in [24] are the only ones who employed this scenario. However, their approach differs from ours since they used the query expansion to add more positive examples before evaluation using ESVMS. To
demonstrate the usefulness of our approach, in Table 12 we have provided how the mAP criterion changes according to the ranking strategy combined with different parameter values for $k$ on the evaluated datasets. It can be out-turned from the bar plot that our re-ranking strategy brings huge benefits to the pipeline improving the mAP criterion.

![Graph showing mAP comparison](image)

**Fig. 12:** Comparison of using NN or krNN-QE$_k$ to rank documents. Red: NN, Green: krNN-QE$_{k=1}$, Blue: krNN-QE$_{k=2}$, and Orange: krNN-QE$_{k=3}$.

**Computational Cost** Our ResNet-20 with the NetVLAD layer consists of nearly 285,000 trainable parameters. In our experiments, training time takes about 75 minutes in a Google Colab's GPU Hardware accelerator of type NVIDIA Tesla P100-PCIE-16GB. Also, Table 4 summarizes the computation time of the global descriptors of evaluated datasets. Note that, the computation involves patch extraction, embeddings extraction from the pre-trained model, and ridge regression optimization steps.

| Dataset          | Images | Elapsed time |
|------------------|--------|--------------|
| ICDAR 2013 (Train) | 400    | 2935s        |
| ICDAR 2013 (Test)  | 1000   | 10255s       |
| CVL (Test)        | 1415   | 14524s       |
| KHATT (Test)      | 600    | 5988s        |

**5 Conclusion**

In this work, we have presented a 1) novel pipeline consisting of a convolutional neural network followed by the NetVLAD layer to extract local descriptors and their corresponding VLAD embeddings in an end-to-end manner and 2)
re-ranking strategy based on query expansion and $k$-reciprocal nearest neighbors to improve initial rankings.

Our results demonstrate improvements and set new standards on both IC-DAR13 and CVL datasets. Having in mind that we evaluated KHATT data using the pre-trained network on ICDAR 2013 dataset, the results on this dataset is quite promising as well. However, there is still room for improvement and optimization of the proposed pipeline in various directions. Since we did not preprocess image patches for input to the network, the preprocessing step could be investigated in more detail. Also, deep learning-based approaches other than NetVLAD such as DeepTen \cite{42} may worth investigating. On the other hand, we have used the NetVLAD layer to extract embeddings but employing it to directly learn global image descriptors could also be beneficial. Finally, historical data are getting more and more attention in recent years, so for future works, the application of the proposed pipeline on historical data must be researched.

6 Conflict of Interest

The authors have no conflict of interest to declare.

References

1. Arandjelovic, R. and Zisserman, A., 2013. All about VLAD. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (pp. 1578-1585).
2. Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T. and Sivic, J., 2016. NetVLAD: CNN architecture for weakly supervised place recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 5297-5307).
3. Asi, A., Abdalhaleem, A., Fecker, D., Märgner, V. and El-Sana, J., 2017. On writer identification for Arabic historical manuscripts. International Journal on Document Analysis and Recognition (IJDAR), 20(3), pp.173-187.
4. Chen, S., Wang, Y., Lin, C.T., Ding, W. and Cao, Z., 2019. Semi-supervised feature learning for improving writer identification. Information Sciences, 482, pp.156-170.
5. Christlein, V., Bernecker, D., Höning, F. and Angelopoulou, E., 2014, March. Writer identification and verification using GMM supervectors. In IEEE Winter Conference on Applications of Computer Vision (pp. 998-1005). IEEE.
6. Christlein, V., Bernecker, D., Maier, A. and Angelopoulou, E., 2015, October. Offline writer identification using convolutional neural network activation features. In German Conference on Pattern Recognition (pp. 540-552). Springer, Cham.
7. Christlein, V., Bernecker, D. and Angelopoulou, E., 2015, August. Writer identification using vlad encoded contour-zernike moments. In 2015 13th International Conference on Document Analysis and Recognition (ICDAR) (pp. 906-910). IEEE.
8. Christlein, V., Bernecker, D., Höning, F., Maier, A. and Angelopoulou, E., 2017. Writer identification using GMM supervectors and exemplar-SVMs. Pattern Recognition, 63, pp.258-267.
9. Christlein, V. and Maier, A., 2018, April. Encoding CNN activations for writer recognition. In 2018 13th IAPR International Workshop on Document Analysis Systems (DAS) (pp. 169-174). IEEE.
10. Christlein, V., Spranger, L., Seuret, M., Nicolaou, A., Král, P. and Maier, A., 2019, September. Deep Generalized Max Pooling. In 2019 International Conference on Document Analysis and Recognition (ICDAR) (pp. 1090-1096). IEEE.
11. Christlein, V.: Handwriting analysis with focus on writer identification and writer retrieval. PhD thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg, 2019.
12. Friedrich-Alexander-Universität Erlangen-Nürnberg, O., Philbin, J., Sivic, J., Isard, M. and Zisserman, A., 2007, October. Total recall: Automatic query expansion with a generative feature model for object retrieval. In 2007 IEEE 11th International Conference on Computer Vision (pp. 1-8). IEEE.
13. Djeddi, C., Meslati, L.S., Siddiqi, I., Ennaji, A., El Abed, H. and Gattal, A., 2014, April. Evaluation of texture features for offline arabic writer identification. In 2014 11th IAPR international workshop on document analysis systems (pp. 106-110). IEEE.
14. Fiel, S. and Sablatnig, R., 2013, August. Writer identification and writer retrieval using the fisher vector on visual vocabularies. In 2013 12th International Conference on Document Analysis and Recognition (pp. 545-549). IEEE.
15. Fiel, S.: Novel methods for writer identification and retrieval. PhD thesis, Technische Universität Wien, 2015.
16. Fiel, S. and Sablatnig, R., 2015, September. Writer identification and retrieval using a convolutional neural network. In International Conference on Computer Analysis of Images and Patterns (pp. 26-37). Springer, Cham.
17. Fiel, S., Kleber, F., Diem, M., Christlein, V., Louloudis, G., Nikos, S. and Gatos, B., 2017, November. Icdar2017 competition on historical document writer identification (historical-wi). In 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR) (Vol. 1, pp. 1377-1382). IEEE.
18. Glorot, X. and Bengio, Y., 2010, March. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the thirteenth international conference on artificial intelligence and statistics (pp. 249-256).
19. He, K., Zhang, X., Ren, S. and Sun, J., 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE international conference on computer vision (pp. 1026-1034).
20. He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
21. He, K., Zhang, X., Ren, S. and Sun, J., 2016, October. Identity mappings in deep residual networks. In European conference on computer vision (pp. 630-645). Springer, Cham.
22. Jegou, H., Douze, M. and Schmid, C., 2009, June. On the burstiness of visual elements. In 2009 IEEE Conference on Computer Vision and Pattern Recognition (pp. 1169-1176). IEEE.
23. Jegou, H., Perronnin, F., Douze, M., Sánchez, J., Perez, P. and Schmid, C., 2011. Aggregating local image descriptors into compact codes. IEEE transactions on pattern analysis and machine intelligence, 34(9), pp.1704-1716.
24. Jordan, S., Seuret, M., Král, P., Lenc, L., Martínek, J., Wiermann, B., Schwinger, T., Maier, A. and Christlein, V., 2020, July. Re-ranking for Writer Identification and Writer Retrieval. In International Workshop on Document Analysis Systems (pp. 572-586). Springer, Cham.
25. Kleber, F., Fiel, S., Diem, M. and Sablatnig, R., 2013, August. Cvldatabase: An off-line database for writer retrieval, writer identification and word spotting. In 2013 12th international conference on document analysis and recognition (pp. 560-564). IEEE.
26. Krizhevsky, A. and Hinton, G., 2009. Learning multiple layers of features from tiny images.
27. Lai, S., Zhu, Y. and Jin, L., 2020. Encoding Pathlet and SIFT Features With Bagged VLAD for Historical Writer Identification. IEEE Transactions on Information Forensics and Security, 15, pp.3553-3566.
28. Louloudis, G., Gatos, B., Stamatopoulos, N. and Papandreou, A., 2013, August. Icdar 2013 competition on writer identification. In 2013 12th International Conference on Document Analysis and Recognition (pp. 1397-1401). IEEE.
29. Maaten, L.V.D. and Hinton, G., 2008. Visualizing data using t-SNE. Journal of machine learning research, 9(Nov), pp.2579-2605.
30. Mahmoud, S.A., Ahmad, I., Al-Khatib, W.G., Alshayeb, M., Parvez, M.T., Märgner, V. and Fink, G.A., 2014. KHATT: An open Arabic offline handwritten text database. Pattern Recognition, 47(3), pp.1096-1112.
31. Marti, U.V. and Bunke, H., 2002. The IAM-database: an English sentence database for offline handwriting recognition. International Journal on Document Analysis and Recognition, 5(1), pp.39-46.
32. Murray, N. and Perronnin, F., 2014. Generalized max pooling. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2473-2480).
33. Murray, N., Jégou, H., Perronnin, F. and Zisserman, A., 2016. Interferences in match kernels. IEEE transactions on pattern analysis and machine intelligence, 39(9), pp.1797-1810.
34. Otsu, N., 1979. A threshold selection method from gray-level histograms. IEEE transactions on systems, man, and cybernetics, 9(1), pp.62-66.
35. Perronnin, F., Sánchez, J. and Mensink, T., 2010, September. Improving the fisher kernel for large-scale image classification. In European conference on computer vision (pp. 143-156). Springer, Berlin, Heidelberg.
36. Schomaker, L., 2008. Writer identification and verification. In Advances in Biometrics (pp. 247-264). Springer, London.
37. Schroff, F., Kalenichenko, D. and Philbin, J., 2015. Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 815-823).
38. Slimane, F., Awaida, S., Mezghani, A., Parvez, M.T., Kanoun, S., Mahmoud, S.A. and Märgner, V., 2014, September. Icfhr2014 competition on arabic writer identification using ahtid/mw and khatt databases. In 2014 14th International Conference on Frontiers in Handwriting Recognition (pp. 797-802). IEEE.
39. Tang, Y. and Wu, X., 2016, October. Text-independent writer identification via CNN features and joint Bayesian. In 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR) (pp. 566-571). IEEE.
40. Wold, S., Esbensen, K. and Geladi, P., 1987. Principal component analysis. Chemometrics and intelligent laboratory systems, 2(1-3), pp.37-52.
41. Xing, L. and Qiao, Y., 2016, October. Deepwriter: A multi-stream deep CNN for text-independent writer identification. In 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR) (pp. 584-589). IEEE.
42. Zhang, H., Xue, J. and Dana, K., 2017. Deep ten: Texture encoding network. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 708-717).