Writer Identification Using GMM Supervectors and Exemplar-SVMs

Vincent Christlein, David Bernecker, Florian Hönig, Andreas Maier, Elli Angelopoulou

DOI: 10.1016/j.patcog.2016.10.005

Writer Identification Using GMM Supervectors and Exemplar-SVMs

Vincent Christlein[a,*], David Bernecker[a], Florian Hönig[a], Andreas Maier[a], Elli Angelopoulou[a]

[a] *Pattern Recognition Lab, Department of Computer Science, Friedrich-Alexander-Universität Erlangen-Nürnberg, Martensstr. 3, 91058 Erlangen, Germany*

Abstract

This paper describes a method for robust offline writer identification. We propose to use RootSIFT descriptors computed densely at the script contours. GMM supervectors are used as encoding method to describe the characteristic handwriting of an individual scribe. GMM supervectors are created by adapting a background model to the distribution of local feature descriptors. Finally, we propose to use Exemplar-SVMs to train a document-specific similarity measure. We evaluate the method on three publicly available datasets (ICDAR / CVL / KHATT) and show that our method sets new performance standards on all three datasets. Additionally, we compare different feature sampling strategies as well as other encoding methods.

Keywords: Writer identification, GMM supervectors, Exemplar-SVM

1. Introduction

Since handwritten text can be used as a biometric identifier like faces or speech, it plays an important role for law enforcement agencies in proving someone's authenticity. However, in such scenarios the decision is typically made by experts in forensic handwriting. In contrast, searching for similar scribes[1] in a large document database raises the need for an automated handwriting system (method). This topic has attracted significant attention recently, especially in the field of historical document analysis [1, 2, 3]. In this application an automatic identification for particular writers can give new insights of life in the past.

The focus of this paper is *writer identification*. Given a document, writer identification is the task of finding the specific writer (author) of the text from a set of writers which are known to the system. Depending on the data at hand, one has to differentiate between offline and online writer identification. In online writer identification the data contains temporal information about the text formation. In contrast, offline writer identification deals only with the handwritten text itself without any additional information. Offline writer identification can be further categorized into two groups [4]: *textural* methods and *allograph*-based methods. In the former group, handwriting is described by global statistics drawn from the style of the handwritten text, e. g., measurements of the ink width or the angles of stroke directions [1, 5, 6] Conversely, in allograph-based methods, the writer is described by the distribution of features extracted from small letter parts (i. e., "allographs") [7, 2, 8, 9, 10]. A vocabulary needs to be trained in advance from feature descriptors of the training set. The hereinafter presented method belongs to the allograph-based methods. Please note also that the best contenders of the ICDAR 2013 writer identification competition stem from this group [11]. Both approaches can also be combined to create a better descriptor [4, 12, 13, 14].

Given some handwritten text, we propose to characterize its scribe by means of the distribution of local feature descriptors. Hereby, the distribution is modeled by a generative model, in particular a Gaussian Mixture model (GMM). We adapt the so-called GMM-UBM method [15], a well-known approach in the field of speech processing. It has shown to yield good results, e. g., for speaker identification [16], or age determination [17].

In speech analysis, a GMM models the distribution of short-time spectral feature vectors of all speakers. Since such a GMM reflects the domain's speech style in general, it is typically denoted as Universal Background Model (UBM). Each speaker of a particular utterance is described by means of a maximum-a-posteriori (MAP) adaptation of the UBM to the feature descriptors of that utterance [15]. See Figure 1 for a schematic illustration of such representation for the case of a two-dimensional feature vector. Finally, the global feature descriptor is formed by stacking the parameters of the adapted GMM (i. e., means, covariances, and weights) in a so-called *supervector*.

For the adaptation of this approach to the image domain, we replace the short-time spectral feature descriptors with RootSIFT descriptors [18], a normalized version of *scale invariant feature transform* (SIFT) descriptors [19]. Op-

[*] Corresponding author

Email addresses: vincent.christlein@fau.de (Vincent Christlein), david.bernecker@fau.de (David Bernecker), florian.hoenig@fau.de (Florian Hönig), andreas.maier@fau.de (Andreas Maier), elli.angelopoulou@fau.de (Elli Angelopoulou)

[1] Note: "writer" and "scribe" are used interchangeably throughout this paper

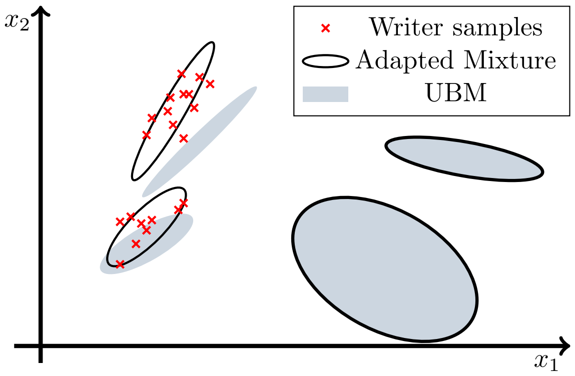

Figure 1: The Universal Background Model (blue) is adapted to samples from one document (red). Mixtures which are influenced more by the new samples are adapted more strongly than others.

tionally, the dimensionality of the feature vectors could be reduced by a principal component analysis (PCA). We show that the resulting GMM supervector encoding yields an excellent representation for individual handwriting. Additionally, we employ support vector machines (SVM) to build individual classifiers per query document. Such an SVM is a linear classifier trained by only one single positive sample and multiple negative samples, it is denoted as Exemplar-SVM [20]. Among others, Exemplar-SVMs have been used successfully for object classification [20], and scene classification [21]. For each class an ensemble of such Exemplar-SVMs is trained. The highest response of the individual Exemplar-SVMs is used to decide upon the class of an unknown image. Unlike these works, we employ a single Exemplar-SVM for each test document using all training documents as negatives. In this way, we change the similarity measure for each test document. We show that this framework outperforms the current state of the art on three publicly available datasets.

This paper is an extension of the work initially published in WACV 2014 [22]. Novel contributions include:

- the integration of Exemplar-SVMs [20], which greatly improve the recognition rate;
- a more thorough analysis of the RootSIFT descriptors, showing that their evaluation at contour edges improves the recognition rate;
- investigation of an additional encoding strategy, termed Gaussian supervector [23] besides Fisher vectors and vectors of locally aggregated descriptors (VLAD).
- evaluation on the KHATT dataset [24] containing 4000 Arabic handwritten documents of 1000 scribes in addtion to ICDAR13 and CVL.

The rest of the paper is organized as follows. Section 2 gives an overview of related work. In Section 3 we provide a detailed description of our framework. We evaluate our method on three datasets and compare our method with the current state of the art in Section 4. Section 5 gives a brief summary and outlook.

2. Related Work

The advantage of textural methods is their interpretability in comparison to allograph-based methods. Furthermore, textural methods are typically faster to compute since no dictionary needs to be trained. A recent textural approach was presented by He and Schomaker [6]. They propose to use the Δ-n Hinge feature which is a generalization of the Hinge feature [4]. The method achieves state-of-the-art results on the ICDAR13 English and Greek subsets.

A mixture of allograph and texture-based methods is presented by Newell and Griffin [12]. They exploit histograms of oriented basic image features (oBIF) and employ the delta encoding as feature descriptors, which encodes a mean oBIF histogram for each individual scribe. Despite yielding very good results on several benchmark datasets, the ICFHR 2014 competition [25] revealed that our previous work using only GMM supervectors [22] achieves higher accuracy.

Allograph-based methods rely on a dictionary trained from local descriptors. This dictionary is subsequently used to collect statistics from the local descriptors of the query document. These statistics are then aggregated to form the global descriptor that is used to classify the document. This procedure is denoted *encoding*.

Fiel and Sablatnig [7] employ Fisher vectors as encoding method to encode local SIFT descriptors. A GMM serves as the vocabulary, i. e., a GMM is computed from SIFT descriptors of the training set. Using this vocabulary, the data of each document is encoded using improved Fisher vectors [26]. The similarity between handwritten documents is computed using the cosine distance between the corresponding Fisher vectors. They show state-of-the-art results on the ICDAR 2011 and CVL dataset. The current best performing method evaluated on the ICDAR13 Greek and English subsets and the CVL dataset is a combination of several features and Fisher vectors [10]. In that work, contour gradient descriptors are combined with K-adjacent segments (KAS), and SURF. Unlike these works, we employ MAP-adapted GMMs, i. e., each document is adapted to a global GMM. The statistics of the adapted GMM form our GMM supervector. Note that one can also compute completely individual codebooks per document or writer using k-means [9] or GMMs [27, 28]. However, the use of a universal background model is much more common in image retrieval [26, 29]. It simplifies the correspondence and distance computation, and typically outperforms solutions using individual codebooks [15, 22]

SIFT, or SIFT-like descriptors are the most common features in allograph-based methods [22, 7, 9, 14]. Wu et al. additionally make use of the scales and orientations given by the SIFT keypoints [14]. In contrast, we evaluate SIFT descriptors densely at the contours while preserving their

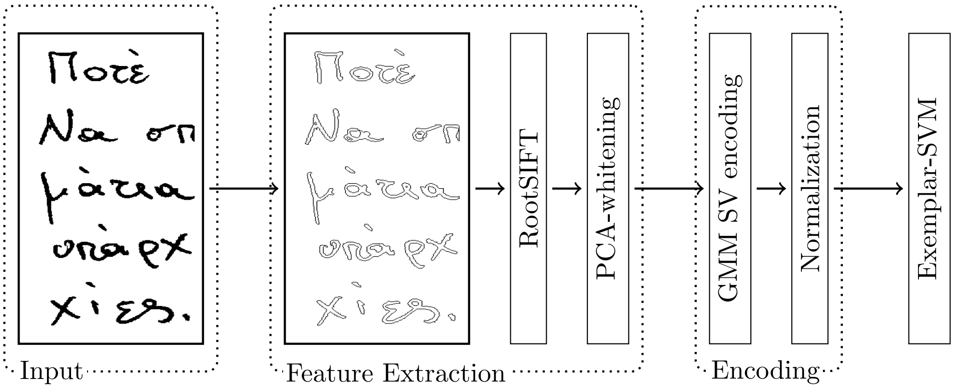

Figure 2: Overview of the entire pipeline. From the input document (left) features are extracted using RootSIFT features computed densely at the script contour. These features are subsequently PCA-whitened and their dimensionality is reduced. These local descriptors are then encoded by means of GMM supervectors. After a normalization step, they are used as input for an Exemplar-SVM. The scores of the Exemplar-SVMs are used for ranking the document.

rotational dependency. Recently, a descriptor specifically developed for script was proposed by He et al. [8], where junctions of the handwriting are extracted and subsequently encoded using self-organizing maps (SOM).

One interesting aspect of a texture-based method stems from Bertolini et al. [30]. They employ a dissimilarity framework, i. e., a single SVM is trained which classifies whether two documents are similar to each other or not. In their approach, each document is first binarized and then compressed to form a texture. Local binary patterns (LBP) and local phase quantization (LPQ) are then used to describe the textures. Each such compressed texture image is divided in 9 parts which are individually evaluated with a trained SVM. Finally, the individual probabilities are merged using different merging techniques. In our approach the image does not need to be divided in parts. Since we employ Exemplar-SVMs, an individual similarity measure for each document is computed.

Closely related to our approach is the work by Schlapbach et al. [31] on online writer identification. First, they build a UBM by estimating a GMM, and then adapt a GMM for each recorded handwriting. The similarity between two recordings is measured by using the sum of posterior probabilities of each mixture. Busch et al. [32] use MAP-adaptation for script classification in conjunction with texture features such as gray-level co-occurence matrices, Gabor and Wavelet engery features. Unlike these works, we employ RootSIFT descriptors and construct GMM supervectors from the adapted GMMs, which are further used for the classification.

Smith and Kornelson [33] compare different encoding schemes in the context of classifying whether images contain text or not. They employ SURF descriptors as their local descriptors. They show that GMM supervectors outperform Fisher Vectors in most scenarios. However, they tested the encoding methods only on an in-house dataset. We employ contour-based RootSIFT descriptors which are encoded by GMM supervectors. Additionally, we employ a different normalization scheme and train Exemplar-SVMs to encode the similarity of each test document to others.

3. Methodology

Figure 2 shows an overview of our entire encoding process. For each document local feature descriptors are computed, in particular RootSIFT descriptors evaluated at the contours. In a training step a dictionary, i. e., the UBM, is trained from the descriptors of an independent document dataset. Each document in question is then encoded using the dictionary and the local descriptors to form a high-dimensional image descriptor, which is then used for classification. The remainder of this section provides the details of the feature extraction, the construction of the UBM, the adaptation process, the normalization of the supervector and its classification using Exemplar-SVMs.

3.1. Features

SIFT descriptors are based on histograms of oriented gradients [19]. Typically they are evaluated at specific keypoint locations, which may contain information about the orientation, scale or other characteristics like the gradient norm. SIFT descriptors have proven to be strong features for image retrieval [18, 34], as well as in the related field of image forensics [35], and have already been successfully used in the context of writer identification [7, 14].

More specifically we use the Hellinger-normalized version of SIFT [18] also known as RootSIFT. In practice, each SIFT descriptor is l_1-normalized followed by an elementwise application of the square-root. For other normalization techniques the reader is referred to [36, 37].

We evaluate several different sampling strategies: a) SIFT descriptors computed at keypoints determined by the scale-space approach as proposed by Lowe [19]; b) SIFT evaluated densely at four different scales, also known as pyramid histogram of visual words (PHOW) [38]; c) SIFT evaluated at the contour points of the script.

Jégou et al. [29] showed that it can be beneficial to reduce the dimensionality of the local SIFT descriptors by means of a principal component analysis (PCA). By retaining only the dimensions related to the largest eigenvalues, possible noise contained in the lower components is removed. Furthermore, transforming the data with a PCA decorrelates the feature descriptors, so that they can be modeled more accurately by a GMM with a diagonal covariance matrix. Moreover, eigenvalue decomposition can be used to whiten the descriptors, i. e., making the covariance equal to the identity matrix. This has been shown to be beneficial for the recognitioon accuracy [39].

3.2. GMM Supervector Encoding

Encoding refers to the process of building a single global feature descriptor from many local descriptors. A widely used encoding method is known as *bag of (visual) words* (BoW).

Universal Background Model:. Similarly to k-means in the classical BoW approach, a global dictionary is constructed, which is denoted as *universal background model* (UBM). It is modeled by a Gaussian mixture model (GMM), since any continuous distribution can be modeled by a GMM with arbitrary precision. Let $\lambda = \{w_k,\, \boldsymbol{\mu}_k,\, \boldsymbol{\Sigma}_k \,|k = 1,\dots,K\}$ be the parameters of the GMM with K mixture components, where w_k, $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$ are the mixture weight, mean vector and covariance matrix of component k, respectively.

Given a feature vector $\boldsymbol{x} \in \mathcal{R}^D$, its likelihood function is defined as

$$p(\boldsymbol{x} \,|\, \lambda) = \sum_{k=1}^{K} w_k g_k(\boldsymbol{x}) \,, \tag{1}$$

where the Gaussian density g_k is:

$$g_k(\boldsymbol{x}) = g(\boldsymbol{x}\,;\,\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{\sqrt{(2\pi)^D|\boldsymbol{\Sigma}_k|}}\, \mathrm{e}^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k)} \,. \tag{2}$$

The mixture weights satisfy the constraint $\sum_{k=1}^{K} w_k = 1$ and $w_k \in \mathbb{R}_+$.

Finally, the posterior probability of a feature vector $\boldsymbol{x}_j$ to be generated by the Gaussian mixture k follows as:

$$\gamma_j(k) = p(k \,|\, \boldsymbol{x}_j) = \frac{w_k g_k(\boldsymbol{x}_j)}{\sum_{l=1}^{K} w_l g_l(\boldsymbol{x}_j)} \,. \tag{3}$$

The GMM parameters are estimated using the Expectation-Maximization (EM) algorithm to optimize a Maximum Likelihood (ML) criterion [40]. The parameters λ of the UBM are iteratively refined to increase the log-likelihood $\log p(\boldsymbol{X} \,|\, \lambda) = \sum_{m=1}^{M} \log p(\boldsymbol{x}_m \,|\, \lambda)$ of the model for the set of training samples $\boldsymbol{X} = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_M\}$. For computational efficiency, the covariance matrix $\boldsymbol{\Sigma}_k$ is assumed to be diagonal, and in the remainder of this paper, the vector of the diagonal elements of $\boldsymbol{\Sigma}_k$ is denoted as $\boldsymbol{\sigma}_k$. Note that a GMM using full covariance matrices can equally well be approximated by a GMM using diagonal covariance matrices by using a larger number of Gaussian mixtures [15].

GMM Adaptation and Mixing:. The final UBM is adapted to each document individually, using all T local descriptors computed for a document W, $\boldsymbol{X}_W = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_T\}$. This can be seen as a MAP adaptation of the UBM to the new samples. New statistics are computed. Let

$$n_k = \sum_{t=1}^{T} \gamma_k(\boldsymbol{x}_t) \,, \tag{4}$$

then the zeroth, first and second order statistics are:

$$E_k^0 = \frac{1}{T} n_k \tag{5}$$

$$E_k^1 = \frac{1}{n_k} \sum_{t=1}^{T} \gamma_k(\boldsymbol{x}_t)\boldsymbol{x}_t \tag{6}$$

$$E_k^2 = \frac{1}{n_k} \sum_{t=1}^{T} \gamma_k(\boldsymbol{x}_t)(\boldsymbol{x}_t \odot \boldsymbol{x}_t) \tag{7}$$

where $E_k^0 \in \mathbb{R}$, $E_k^1 \in \mathbb{R}^D$, and $E_k^2 \in \mathbb{R}^D$, and $\odot$ denotes the Hadamard product.

Finally, these statistics are mixed together with the information contained in the UBM. Densities with high posteriors are adapted more strongly (cf. Figure 1). This is controlled by a fixed relevance factor r^τ for the adaptation coefficients

$$\alpha_k^\tau = \frac{n_k}{n_k + r^\tau} \tag{8}$$

for each parameter τ $\big(\tau \in \{w,\, \boldsymbol{\mu},\, \boldsymbol{\Sigma}\}\big)$. We use the same r for each τ as suggested by Reynolds et al. [15] (τ as a superscript is therefore omitted subsequently). The resulting mixture parameters follow as:

$$\hat{w}_k = \delta \left(\alpha_k E_k^0 + (1 - \alpha_k)\, w_k\right) \tag{9}$$

$$\hat{\boldsymbol{\mu}}_k = \alpha_k E_k^1 + (1 - \alpha_k)\boldsymbol{\mu}_k \tag{10}$$

$$\hat{\boldsymbol{\sigma}}_k = \alpha_k E_k^2 + (1 - \alpha_k)\left(\boldsymbol{\sigma}_k + \boldsymbol{\mu}_k^2\right) - \hat{\boldsymbol{\mu}}_k^2 \tag{11}$$

where δ is a scaling factor ensuring that the weights of all components sum up to one. Note: $\boldsymbol{\mu}^2$ and $\hat{\boldsymbol{\mu}}^2$ is a shorthand notation for $\boldsymbol{\mu} \odot \boldsymbol{\mu}$ and $\hat{\boldsymbol{\mu}} \odot \hat{\boldsymbol{\mu}}$, respectively.

Finally, the supervector $\boldsymbol{s}$ is formed by concatenating all parameters from the adapted GMM:

$$\boldsymbol{s} = \left(\hat{w}_1, \dots, \hat{w}_K, \hat{\boldsymbol{\mu}}_1^\top, \dots, \hat{\boldsymbol{\mu}}_K^\top, \hat{\boldsymbol{\sigma}}_1^\top, \dots, \hat{\boldsymbol{\sigma}}_K^\top\right)^\top \,. \tag{12}$$

The vector $\boldsymbol{s}$ represents the global feature descriptor, and consists of $(1 + 2D)K$ elements. Note that often only the adapted mean components are used, which reduces the vector size to DK. The effects of this reduction are evaluated in Section 4.4.

Normalization:. Sanchez et al. [26] propose a two step normalization for the resulting vector after an encoding with Fisher vectors. First, *power-normalization* is applied to each element, i. e.,

$$s_i = \operatorname{sign}(s_i)|s_i|^\rho \,, \forall s_i \in \boldsymbol{s} \,, 0 < \rho \leq 1 \,. \tag{13}$$

Typically, ρ is set to 0.5, which then equals the Hellinger normalization, cf. Section 3.1. Next, the vector is l_2-normalized, i. e., $\boldsymbol{s} = \boldsymbol{s}/\|\boldsymbol{s}\|_2$. Through these normalization steps image-independent information, like the background data, is discarded by reducing the influence of more frequent descriptors [26]. Furthermore, Sánchez et al. [26] showed that an l_2-normalization is, in general, beneficial when used in combination with linear classifiers.

Arandjelovic and Zisserman [34] propose to use intra-normalization for VLAD encodings. Similar to GMM supervectors, VLAD is composed of multiple components. They suggest to apply a component-wise l_2-normalization which is followed by a global l_2-normalization. This helps to reduce the influence of dominant components.

Both normalization strategies will be evaluated when applied on the proposed GMM supervectors. Moreover, we evaluate two different variants of the GMM supervectors by applying a feature mapping. This can also be seen as a form of normalization. Hereby, a feature mapping inspired by the symmetrized Kullback-Leibler divergence is applied [41]. We refer to this mapping as KL-normalization. It is computed as:

$$\tilde{\boldsymbol{\mu}}_k = \sqrt{w_k}\boldsymbol{\sigma}_k^{-\frac{1}{2}} \odot \hat{\boldsymbol{\mu}}_k \tag{14}$$

$$\tilde{\boldsymbol{\sigma}}_k = \sqrt{\frac{w_k}{2}}\boldsymbol{\sigma}_k^{-1} \odot \hat{\boldsymbol{\sigma}}_k \,. \tag{15}$$

In the case of mean-adaptation only, the resulting supervector follows as:

$$\tilde{\boldsymbol{s}}_{\mathrm{m}} = \left(\tilde{\boldsymbol{\mu}}_1^\top, \ldots, \tilde{\boldsymbol{\mu}}_K^\top\right)^\top \,, \tag{16}$$

or as suggested by Xu et al. [41], one can build a $2DK$ long supervector:

$$\tilde{\boldsymbol{s}}_{\mathrm{mv}} = \left(\tilde{\boldsymbol{\mu}}_1^\top, \ldots, \tilde{\boldsymbol{\mu}}_K^\top, \tilde{\boldsymbol{\sigma}}_1^\top, \ldots, \tilde{\boldsymbol{\sigma}}_K^\top\right)^\top \,. \tag{17}$$

In this way, properties of the UBM are incorporated implicitly into the normalized global descriptor ($\tilde{\boldsymbol{s}}_{\mathrm{m}}$ and $\tilde{\boldsymbol{s}}_{\mathrm{mv}}$) that are normally not reflected in the supervector. Note that the KL-Kernel has also been used in conjunction with GMM supervectors and SVMs in the field of speaker verification [16].

3.3. Other Encoding Methods

Apart from the different variants of the GMM supervectors (choice of features and normalization strategies), several other encoding methods exist. The most popular one is certainly *vector quantization*, however it has been shown that it is inferior to other encoding methods [42, 39]. We will compare the proposed method with other encoding methods concentrating on those which are derived from a GMM. More specifically, we will evaluate (improved) *Fisher vectors* (FV) [26] and *vector of locally aggregated vectors* (VLAD) [29], in particular a probabilistic variant of VLAD [29, 39, 33]. We will also evaluate another encoding method derived from a GMM, namely the *Gaussianized vector representation* (GVR) [23]. In the following paragraphs we will briefly present those three encoding methods.

Fisher Vectors:. This representation is in many ways similar to GMM supervectors [26]. The distribution of samples is also described by a generative model (i. e., a GMM). Each sample is then transformed to the gradient space of the model parameters. The Fisher vectors are derived from Fisher kernels, in particular the Fisher score of the samples normalized by the square-root of the Fisher information matrix [26].

Similar to the proposed MAP-adapted GMM supervectors, Fisher vectors encode statistics up to the second order:

$$\hat{\boldsymbol{\mu}}_k = \frac{1}{T\sqrt{w_k}} \sum_{t=1}^{T} \gamma_t(\boldsymbol{x}_t)\big((\boldsymbol{x}_t - \boldsymbol{\mu}) \oslash \boldsymbol{\sigma}_k\big) \,, \tag{18}$$

$$\hat{\boldsymbol{\sigma}}_k = \frac{1}{T\sqrt{2w_k}} \sum_{t=1}^{T} \gamma_t(\boldsymbol{x}_t)\Big(\big((\boldsymbol{x}_t - \boldsymbol{\mu}_k) \odot (\boldsymbol{x}_t - \boldsymbol{\mu}_k)\big) \oslash \boldsymbol{\sigma}\Big) \,, \tag{19}$$

where $\odot$ and $\oslash$ denote the element-wise multiplication and division, respectively. Finally, the concatenation of $\hat{\boldsymbol{\sigma}}_k$ for $k = 1, \ldots, K$ form the $2DK$-dimensional Fisher vector.

Probabilistic VLAD:. The non-probabilistic version of the VLAD representation [29] achieved state of the art results on several benchmark datasets, especially when its representation was improved with intra-normalization [34] or residual normalization [43].

In contrast to the hard assignment of codewords by determining the nearest cluster centers, we use a probabilistic version of VLAD [29], which uses weighted distances to nearby cluster centers. This allows for a better comparison to the other GMM-based representations, since the same posteriors can be used.

$$\boldsymbol{v}_k = \sum_{t=1}^{T} \gamma_k(\boldsymbol{x}_t)(\boldsymbol{x}_t - \boldsymbol{\mu}_k) \,. \tag{20}$$

For the non-probabilistic version, the $\boldsymbol{\mu}_k$ would be the cluster centers obtained by k-means. $\gamma_k(\boldsymbol{x})$ would be a Dirac function returning 1 if $\boldsymbol{\mu}_k$ is the nearest cluster center to $\boldsymbol{x}_t$ and 0 otherwise. Similarly to the other representations, each $\boldsymbol{v}_k$ is stacked together to form a supervector representation containing DK elements.

Gaussianized Vector Representation:. This representation is another form of supervector encoding [23]. It can be seen as an extension of the probabilistic VLAD and is defined as:

$$\boldsymbol{z}_k = (n_k\boldsymbol{\sigma}_k)^{-\frac{1}{2}}\boldsymbol{v}_k \,, \tag{21}$$

where $\boldsymbol{v}_k$ is computed as in the soft VLAD representation, Equation (20), and $\boldsymbol{\sigma}_k$ is the diagonal of the covariance matrix of the UBM. Thus, more information about the background writers is incorporated, similarly to the KL-normalization. Again, all K components form the supervector representation.

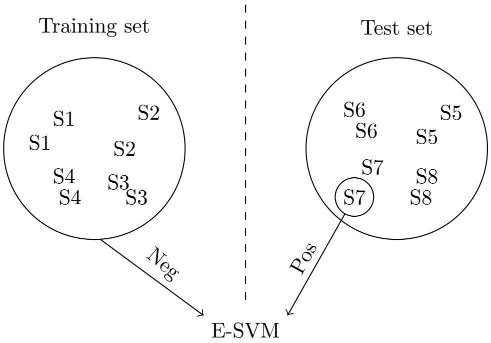

Figure 3: For each document of the test set an individual Exemplar-SVM is trained. The GMM supervector of this document is used as positive sample, while all the encodings of the training set are used as negatives.

3.4. Exemplar-SVMs

Instead of using one SVM per object category, Malisiewicz et al. [20] proposed to use an ensemble of Exemplar-SVMs for object detection. This means, that for each instance of all object classes in the training set an individual (linear) SVM is trained. For training each of these Exemplar-SVMs the current sample is used as the only instance of the positive class and all other training samples as negatives. The large margin formulation follows similarly to the standard SVM:

$$\operatorname*{argmin}_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2 + c_p h(1-\boldsymbol{w}^\top \boldsymbol{x_p} - b) + c_n \sum_{\boldsymbol{x_n}\in\mathcal{N}} h(1+\boldsymbol{w}^\top \boldsymbol{x_n} + b)\,, \tag{22}$$

where $h(x) = \max(0, x)$ is the hinge loss function, $\boldsymbol{x_p}$ is the single target positive sample and $\boldsymbol{x_n}$ are the descriptors of the negative training set $\mathcal{N}$, respectively. c_p and c_n are regularization parameters for balancing the positive and negative costs. This has the effect that a single SVM does not have to be able to recognize different views of the same object, but can concentrate on classifying a single view. The authors of [20] showed that an ensemble of such Exemplar-SVMs generalizes well, although each single Exemplar-SVM has a very strict decision boundary. As each classifier solves a simplified problem compared to a full category classifier, a simple regularized linear SVM is sufficient.

Note that Exemplar-SVMs can be reformulated to Exemplar-One-Class SVMs [44]. This has the advantage that no individual class weight has to be calibrated. Very popular is also the approximation of Exemplar-SVMs by Exemplar-LDA [45, 46, 44], where the training set is approximated by a Gaussian. Furthermore, Exemplar-SVMs can also be used as feature encoders [47, 44], where the normalized computed weight vector $\boldsymbol{w}$ is directly used as new feature descriptor for the specific exemplar.

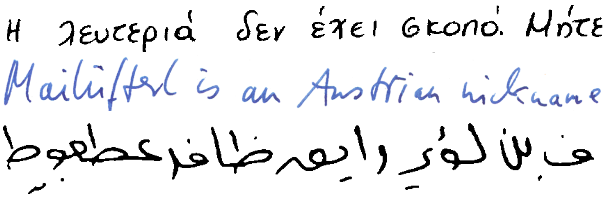
Η λευτεριά δεν έχει σκοπό. Μήτε
Mailüfterl is an Austrian nickname

Figure 4: Example lines of the three datasets, from top to bottom: ICDAR13, CVL and KHATT.

For our application we have to modify this approach, since the training and testing subsets of a typical writer identification dataset are disjoint, i. e., the writers of the test set are not part of the training set. Therefore, the normal recognition pipeline, i. e., learning a multi-class classifier by using samples from the training set which then predicts the class of the samples in the test set, can not be applied. However, by using Exemplar-SVMs we can circumvent this problem. We do not train Exemplar-SVMs on the classes (=writers), of the training set at all. Instead, we train an Exemplar-SVM during test time for each query document by using the query document as positive sample, and all training samples are used as negatives. This is illustrated in Figure 3. Each other document is scored against the Exemplar-SVM of the query document and ranked according to the scores. The author associated with the document having the highest score is with high probability also the author of the query document. Intuitively, this can be seen as an adjustment of the similarity measure. Instead of finding the nearest neighbor according to the cosine distance, a document specific similarity is learned.

The global feature vectors are high-dimensional, in our case 6400-dimensional. In such a space, all points tend to lie at the periphery of the manifold. On one hand this is the curse of dimensionality, on the other hand it is a blessing since the exemplar needs to be separable enough from the negative descriptors [46]. For the Exemplar-SVM computation, we employ LIBLINEAR [48] that relies on coordinate descent. Another possibility would be to use stochastic gradient descent (SGD) as suggested by Zepeda et al. [47]. However, we found LIBLINEAR to be fast and robust. Computing the 1000 E-SVMs of the ICDAR13 benchmark dataset takes about 2.3 minutes using a standard PC (Intel Xeon E3-1276 3.60GHz), see Section 4.8.

4. Evaluation

In the following paragraphs we document which datasets and evaluation metrics we use for evaluating our approach. Subsequently, we show the impact of the feature sampling as well as the GMM parameters, normalization, and Exemplar-SVMs. Finally, we compare our method to other GMM-based encoding methods and the state of the art method for writer identification.

4.1. Benchmark Datasets

We use the publicly available CVL, ICDAR13, and KHATT datasets for evaluation. From the example lines in Figure 4, one can see the large variation in visual appearance between these datasets.

ICDAR13 [11] was part of the ICDAR 2013 writer identification competition. It consists of two disjoint datasets, an experimental dataset for training and a benchmark dataset for testing. The experimental dataset stems from the ICFHR 2012 writer identification contest [49] and consists of 100 scribes. The benchmark set contains 250 scribes. In both subsets each scribe contributed four documents. Two documents were written in Greek, the other two in English. The documents of the dataset are all binarized.

CVL [50] consists of 310 scribes. Twenty-seven of them contributed seven documents each, which form the training set. The other 283 scribes contributed five documents each, which form the test set. For each scribe, one document is written in German and the remaining ones are written in English. Note that we binarized the documents for the evaluation using Otsu's method [51] to be more similar to the ICDAR13 dataset.

KHATT [24] was part of the ICFHR 2014 Arabic writer identification competition. KHATT consists of Arabic handwritten documents from 1000 scribes, where each scribe wrote four documents. The database is divided into three disjoint sets for training (70%), validation (15%) and testing (15%), respectively. The document images are in grayscale.

4.2. Evaluation Metrics

For evaluation, each document is tested against all remaining ones. The results for writer identification are expressed in terms of mean average precision (mAP) and TOP-k rates for different ranks k.

Mean average precision is a measure used in the context of information retrieval. Let us first specify average precision (aP). Consider a query that returns Q documents in a ranked sequence. Out of the Q documents, R are relevant, i. e., written by the queried author. aP is calculated by

$$\mathrm{aP} = \frac{1}{R}\sum_{k=1}^{Q} \Pr(k)\cdot \mathrm{rel}(k)\,, \qquad (23)$$

where rel(k) is a binary function that is 1 when the document at rank k is relevant, and 0 otherwise. Pr(k) is the precision at rank k of the query (i. e., number of relevant documents in the first k query items divided by k). The mAP is computed as the average over all aP values of all possible queries. In this way, if relevant documents are found at a lower rank, higher values are assigned. Note that the recently employed writer retrieval criterion [7, 50] is closely related to the mAP.

The identification rate is given by the *soft* and *hard* TOP-k rates. The soft TOP-k rates (abbreviated as S-k) give the probability that at least one document of the same writer is among the k highly ranked documents. In contrast, the hard TOP-k rates (abbreviated as H-k) denote the probability that among the k first documents exactly k documents are from the same writer.

Descriptor	mAP
R-SIFT [22]	69.2
Dense-R-SIFT	76.0
C-R-SIFT	80.6
C-R-SIFT + PCA-64	81.8
C-R-SIFT + PCA-64 + Wh.	84.0

Table 1: Comparison of SIFT using different modalities. From top to bottom: R-SIFT computed at SIFT keypoints, R-SIFT evaluated densely over the image (Dense-R-SIFT), R-SIFT computed at the contour of the script (C-R-SIFT). C-R-SIFTs are evaluated with a PCA-dimensionality reduced version retaining 64 components (second last row) and additionally whitened (last row). The results are given in terms of mAP evaluated on the ICDAR13 training set.

In the following sections we evaluate the influence of different parts of the pipeline. We begin with the feature extraction, followed by the evaluation of different encoding methods. Finally, we assess the influence of the normalization step and compare the results of the complete pipeline with other encoding methods and the state of the art method. The UBM-GMM is learned from 150000 randomly selected descriptors of the associated training set. Taking all descriptors would be computationally prohibitive. Unless otherwise specified, we use the values of our previous work [22]: 100 components for the GMM, GMM supervectors as encoding method using a relevance factor $r = 28$ and the supervectors are normalized using power-normalization followed by an l_2-normalization. The cosine distance is used for comparing two global descriptors as a fast similarity measure (only a dot product for l_2 normalized feature descriptors) following previous work on image retrieval [26, 34].

4.3. Influence of Feature Extraction Modalities

First, we evaluate the influence of the descriptor. More specifically, we look at the influence of the sampling strategy used in conjunction with SIFT. The baseline is given by our previous results in which we used Hellinger-normalized SIFT (R-SIFT) features evaluated at SIFT keypoints [22]. We compare this baseline against a densely sampled version of RootSIFT (Dense-R-SIFT). We use the implementation provided by the VLFeat Toolbox [52] using the standard bin sizes (4,6,8,10), and a step size of 3. Another sampling strategy was inspired by the contour-gradient descriptor proposed by Jain and Doermann [9], who proposed a SIFT-like descriptor evaluated only at the contour of the script. Instead we directly use RootSIFT descriptors with their standard size, i. e., a bin size of 4. However, we omit rotational invariance, i. e., setting the descriptor upright at each position. Fiel and Sablatnig [7] showed that rotational-dependent SIFT descriptors are beneficial for writer identification. The first three rows in Table 1 show that dense sampling is better than using SIFT keypoints. Computing SIFT at the contour of the handwriting (C-R-SIFT) achieves the highest rates.

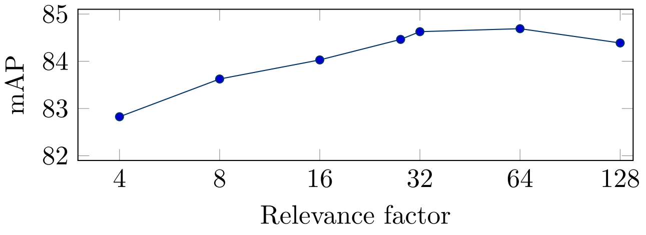

Figure 5: Evaluation of the relevance factor using the ICDAR13 training set.

Encoding	mAP
$\mathrm{SV_w}$	72.3
$\mathrm{SV_m}$	84.6
$\mathrm{SV_c}$	83.7
$\mathrm{SV_{mc}}$	84.7
$\mathrm{SV_{wmc}}$	84.7

(a) Component combinations.

Encoding	mAP
$\mathrm{SV_{wmc}}$	84.4
$\mathrm{SV_{wmc}}$ + ssr	84.7
$\mathrm{SV_{wmc}}$ + intra	84.2
$\mathrm{SV_m}$ + KL-norm	85.1
$\mathrm{SV_{mc}}$ + KL-norm	85.9

(b) Normalization comparison.

Table 2: Comparison of different GMM supervector component combinations (a): only weights ($\mathrm{SV_w}$); means ($\mathrm{SV_m}$); covariances ($\mathrm{SV_c}$); means and covariances ($\mathrm{SV_{mc}}$); weights and means and covariances ($\mathrm{SV_{wmc}}$). (b) shows different normalization techniques: all components and no normalization ($\mathrm{SV_{wmc}}$); all components and element-wise signed square root ($\mathrm{SV_{ssr}}$); all components and intra-normalization ($\mathrm{SV_{wmc}}$ + intra); KL-normalized mean components ($\mathrm{SV_m}$ + KL-norm); KL-normalized mean and covariances ($\mathrm{SV_{mc}}$ + KL-norm). All rates are given in terms of mAP evaluated on the ICDAR13 training set.

Since we seek to have a compact representation for the subsequent steps of the pipeline, we evaluate the influence of reducing the dimensionality of the RootSIFT descriptors to 64 components as well as performing an additional whitening step. Table 1 reveals that especially the whitening step is beneficial for the C-R-SIFT representation (applying a dimensionality reduction and whitening to the original RootSIFT representation gives 66.2 mAP). Note that the PCA-decorrelated versions are subsequently l_2-normalized. For the rest of the paper, we use this compact representation (C-R-SIFT + PCA-64 + Wh.).

4.4. GMM Supervector parameters

GMM supervectors depend on: a) the relevance factor r, b) the adapted components, and thus the supervector representation, and c) the applied normalization. In general also the number of Gaussians for the GMM training is important, however we have found that the accuracy is quite stable for a number of Gaussians between 50 and 150 [22], thus using 100 Gaussians for the following experiments.

Figure 5 shows the influence of different relevance factors. In contrast to the relevance factor $r = 28$ of our previous work [22], it seems that a higher relevance factor of 64 is slightly better suited for C-R-SIFT descriptors. Although the relevance factor depends on n_k (see Equation (8)), and thus is dataset dependent, we found the chosen relevance factor to be working well for other datasets, too.

Next, we compare different supervector representations, i. e., we experiment with only weights ($\mathrm{SV_w}$), means ($\mathrm{SV_m}$), covariances ($\mathrm{SV_c}$) or combinations of these three $\mathrm{SV_{mc}}$ and $\mathrm{SV_{wmc}}$. Note that they were normalized using power normalization (ssr). Table 2a shows that using mean supervectors as the sole representation is superior to supervector consisting of the adapted covariances, or weights. Higher dimensional combinations do not seem to improve the recognition rate much to justify the increase in dimensionality. Thus, we stick to the more compact representation resulting in a 6400-dimensional supervector.

We also evaluated different normalization techniques. Table 2b shows that power-normalization is superior to intra-normalization or just applying l_2-normalization. Rows four and five of Table 2b show the results of using the normalization derived from the KL-kernel. This representation seems to further improve the recognition rate. Consequently, we chose to use this normalization for the subsequent evaluations, where we use mean-adapted GMM supervectors to save training time for the Exemplar-SVM denoted as $\mathrm{SV_{m,kl}}$.

4.5. Comparison with Other Encoding Methods

We compare our proposed encoding method, i. e., GMM supervectors, with other encoding techniques that use a GMM as background model. We present the results of the ICDAR13 test set so that they can be compared to the results of the state of the art in Table 3. When comparing the different encoding methods, the GMM supervector encoding performs best, while Fisher vectors perform second best. Dimension-wise $\mathrm{SV_{wmc}}$ has the largest feature dimension of $2KD + D$, while Fisher vectors typically encode first and second order statistics resulting in a dimension of $2KD$. The other encoding methods (PVLAD, GSV, Proposed) encode only first order statistics, thus having a lower dimension of KD (i. e., 6400-dimensional). This speeds up the subsequent parts of the pipeline, especially the use of the Exemplar-SVMs.

4.6. Exemplar-SVM Analysis

For each test document an individual E-SVM is created using all the documents of the training set as negative samples. We choose to use the same class weights as proposed by Malisiewicz et al. [20], i. e., $c_p = 0.5$ and $c_n = 0.01$, where c_p is the class weight for the positive set and c_n for the negative set. We scale these parameters by a complexity parameter C which is validated using the validation sets (for the ICDAR13 set, we split the training dataset in two subsets such that 75% is used to train the SVMs and 25% for validation; for the CVL dataset we used the same C as for the ICDAR13 dataset, since the training set was too small for splitting).

First we evaluated the influence of the number of available negatives used to train the Exemplar-SVMs. Figure 6

Method	S-1	S-2	S-5	S-10	H-2	H-3	mAP
C-R-SIFT + PVLAD	97.3	97.8	98.4	98.8	67.8	45.1	79.0
C-R-SIFT + GSV	97.4	98.0	98.5	99.0	66.8	45.1	78.9
C-R-SIFT + $\text{FV}_{\text{mc,ssr}}$	97.4	98.4	98.7	99.0	69.0	47.2	80.3
C-R-SIFT + $\text{SV}_{\text{wmc,ssr}}$	98.0	98.4	98.9	99.1	71.1	47.1	80.9
C-R-SIFT + $\text{SV}_{\text{m,kl}}$	98.2	98.6	98.7	98.9	71.2	47.7	81.4

Table 3: Comparison of different encoding methods evaluated on the ICDAR13 test set.

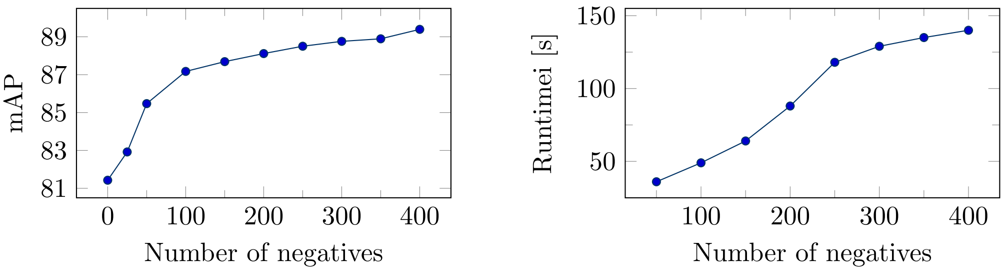

Figure 6: Evaluation of the accuracy (left) and time (right) using different number of negatives for the Exemplar-SVM training, evaluated with the ICDAR13 test set.

(left) shows that with a growing number of negative training samples the retrieval rate raises. Even a low number of negatives has a positive influence on the mean average precision. The better accuracy comes with the prize of a higher runtime, see Figure 6 (right). However, this makes just a small part of the overall runtime, cf. Section 4.8,

4.7. Evaluation of Our Entire Pipeline

We compare our baseline [22] with the proposed more compact representation, i. e., C-R-SIFT descriptors with mean-adapted GMM supervectors and KL-normalization (Proposed) and our extended pipeline, i. e., the integration of Exemplar-SVMs (Proposed + E-SVM). We show that this additional step sets new standards on all evaluated datasets.

4.7.1. Results for ICDAR13

Interestingly, Table 4 shows that all proposed encoding methods (Table 3) perform better than the methods currently considered the state of the art [11, 22, 9]. The work by Fiel et al. [7] (SIFT + FV) and our previous work [22] (R-SIFT + SV) are based on sparsely sampled SIFT and RootSIFT, respectively. In comparison with their contour-based versions, we can conclude that the feature sampling is indeed an important factor for a high mAP.

As can be seen in Table 4, using Exemplar-SVMs gives a further boost in terms of accuracy. For example, on the ICDAR13 dataset the hard TOP-2 and TOP-3 rates improve by about 13 and 15 percentage points, respectively. Thus, we are able to detect not even the document from the same language, but find with a high probability the documents of the same author even in a different script style.

As Table 5 shows, if we evaluate the languages independently, our approach without Exemplar-SVMs performs worse than the feature combination approach of Jain and Doermann [10][2]. However, using our extended pipeline, we even achieve a recognition rate of 100% for the Greek documents, and a TOP-1 accuracy of 99% for the English dataset.

4.7.2. Results for CVL

The CVL dataset is evaluated in two different ways:

A) Using solely the CVL training set for creating the background model, PCA-transformation matrix, and the computed GMM supervectors as negatives for the Exemplar-SVM.

B) As training set we merged two additional datasets: i) the complete IAM dataset [53] consisting of 1539 pages, and ii) the ICDAR 2011 benchmark dataset [54] containing 209 documents. The UBM and the PCA-transformation matrix were computed using the ICDAR13 training set.

Thus, A) gives a fair comparison to other methods, since only information from the dataset itself is used. For B) we show what is possible with additional training data, even when this data comes from different datasets. Similarly, Table 6 shows a large improvement using Exemplar-SVMs in the case of scenario B) where we enriched the training set. However, using solely the CVL training set for the training of the Exemplar-SVMs worsens the results. This is in contradiction to our Exemplar-SVM analysis, where

[2] The authors have not provided results for the complete ICDAR13 dataset

Method	S-1	S-2	S-5	S-10	H-2	H-3	mAP
SIFT + FV$_{\text{mc,ssr}}$ [7]	90.9	93.6	97.0	98.0	44.8	24.5	-
HIT-ICG	94.8	96.7	98.0	98.3	63.2	36.5	-
CS [9]	95.1	97.7	98.6	99.1	19.6	7.1	-
R-SIFT + SV$_{\text{wmc,ssr}}$ [22]	97.1	98.5	98.9	99.0	42.8	23.8	67.1
Proposed	98.2	98.6	98.7	98.9	71.2	47.7	81.4
Proposed + E-SVM	**99.7**	**99.7**	**99.8**	**99.8**	**84.8**	**63.5**	**89.4**

Table 4: Comparison of our method with the state of the art evaluated on the ICDAR13 test set. Values of HIT-ICG, [7], [9] are taken from [11].

	Greek					English				
Method	S-1	S-2	S-5	S-10	mAP	S-1	S-2	S-5	S-10	mAP
SIFT + FV [7]	88.4	92.0	96.8	97.8	-	91.4	94.2	95.8	97.2	-
HIT-ICG	93.8	96.4	97.2	97.8	-	92.2	94.6	96.4	96.8	-
CS [9]	95.6	98.2	98.6	99.2	-	94.6	97.0	98.4	98.8	-
SV [22]	97.4	98.6	99.0	99.4	98.2	96.4	97.4	98.0	98.8	97.2
Δ-n H. [6]	96.0	-	-	98.4	-	93.4	-	-	97.8	-
Comb. [10]	99.2	99.6	99.8	99.8	99.5	97.4	97.8	98.6	98.8	97.9
Proposed	98.2	98.6	99.2	99.4	98.6	95.8	96.6	97.0	97.6	96.5
Proposed + E-SVM	**100**	**100**	**100**	**100**	**100**	**99.0**	**99.2**	**99.8**	**100**	**99.3**

Table 5: Comparison with the state of the art on the ICDAR13 test set: Greek only (left) and English only (right). Values of HIT-ICG, [7], [9] are taken from [11].

even 25 negatives for the Exemplar-SVM training bring a small improvement for the ICDAR13 test set. We believe that the small number of different scribes in the training set prevents the creation of strong Exemplar-SVMs. Also the lack of a suitable validation set makes a calibration of the balancing factor C impossible. Note that our proposed method without Exemplar-SVMs does not improve over our baseline approach [22]. This might be related to the rather homogeneous CVL dataset, where a more dense sampling does not improve over a sparse sampling. However, the proposed supervector is much smaller than our baseline (KD vs. $2KD + K$). Further note that the different UBM and PCA-transformation result in slightly worse results of "proposed" in B) compared to A).

4.7.3. Results for KHATT

The same holds true for the KHATT dataset, which we additionally evaluated[3]. Our strong baseline [22] achieves slightly better results compared to the proposed system using C-R-SIFT descriptors and mean-adapted GMM supervectors (Proposed). However, when we apply the complete pipeline, i. e., using Exemplar-SVMs, we achieve recognition rates near 100%. This is related to the large training and validation sets provided by this dataset. This also indicates that larger training sets are needed for a further improvement in recognition rates.

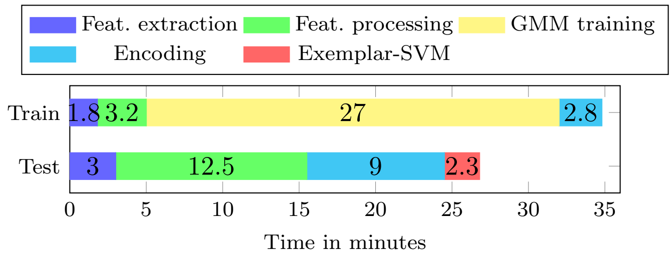

Figure 7: Runtime of the different pipeline steps, evaluated using the ICDAR13 dataset.

4.8. Runtime Evaluation

We measured the runtime of different steps of our proposed pipeline, see Figure 7. GMM training takes the most time followed by the feature processing. Feature processing comprises the Hellinger normalization and PCA transformation. Interestingly, while the encoding step takes three times as long as the feature extraction part, the Exemplar-SVM part takes less time than the feature extraction using LIBLINEAR. The training of the 1000 Exemplar-SVMs of the ICDAR13 benchmark dataset takes only about 2.3 minutes. However note that the number of negatives is 400 (the ICDAR13 training dataset). With more negatives this step could take more time. The processing time for the

[3] Note that the evaluation protocol of [5] is different from ours, since the authors chose to use not the official dataset splitting: They use two documents from each author to train a multi-class SVM (resulting in 2000 documents). The system is then tested by using one document as probe and the other as query, i. e., 1000 evaluations. In contrast, we evaluate the algorithm on the official testing subset in a leave-one-document-out manner.

	Method	S-1	S-2	S-5	S-10	H-2	H-3	H-4	mAP
	SIFT + $\text{FV}_{\text{mc,ssr}}$ [7]	97.8	98.6	99.1	99.6	95.6	89.4	75.8	-
	R-SIFT + $\text{SV}_{\text{wmc,ssr}}$ [22]	99.2	99.2	99.5	99.6	98.1	95.8	88.7	97.1
	Comb. [10]	**99.4**	**99.5**	**99.6**	**99.7**	98.3	94.8	82.9	96.9
A)	**Proposed**	98.8	99.0	99.2	99.2	97.8	95.3	88.8	96.4
	Proposed + E-SVM	93.4	94.4	96.1	97.2	91.0	87.3	80.0	91.0
B)	**Proposed**	98.7	98.9	99.1	99.2	97.7	95.2	87.3	96.1
	Proposed + E-SVM	99.2	**99.5**	**99.6**	**99.7**	**98.4**	**97.1**	**93.6**	**98.0**

Table 6: Comparison with the state of the art on the CVL test set. We experimented using different negative sets for the E-SVM training: A) the CVL training set; B) the IAM datasets plus the ICDAR 2011 benchmark dataset.

Method	S-1	S-2	S-5	S-10	H-2	H-3	mAP
Edge Hinge [5]	84.1	-	91.8	92.8	-	-	-
R-SIFT + $\text{SV}_{\text{wmc,ssr}}$ [22]	97.8	99.0	99.3	99.5	90.3	75.0	92.7
Proposed	96.0	97.8	98.5	98.7	87.0	67.8	88.0
Proposed + E-SVM	**99.5**	**99.5**	**99.5**	**99.5**	**96.5**	**92.5**	**97.2**

Table 7: Comparison with the state of the art on the KHATT test set.

ICDAR13 test set was about 27 minutes, i. e., each image took about 1.6s to process. Please note that our implementation has not been optimized regarding the runtime, and only some parts were parallelized. We see room for improvement, especially with the feature processing and encoding step.

5. Conclusion

In this work, we have presented a new framework for offline writer identification setting new performance standards on three benchmark datasets. First, we proposed the use of SIFT descriptors computed densely at the script contour. We showed that this sampling strategy greatly improves the recognition rates in comparison to other strategies on the difficult bilingual ICDAR13 dataset. Similar to our previous work, we evaluated the influence of different encoding methods and showed that GMM supervectors are superior to other GMM-based encoding methods. We can further improve the recognition accuracy by using a normalization derived from the KL-kernel and at the same time reduce the dimensionality of the feature vector. Additionally, we extended our previous work [22] by using Exemplar-SVMs and showed that this step boosts the recognition rate on all datasets. However, large datasets such as KHATT benefit the most, due to the significant size of the training set.

Since feature extraction was not the focus of this paper, it would be interesting to analyze, how features, specifically designed for script, e. g., the recently developed *junclets* [8], would perform in conjunction with GMM supervectors and Exemplar-SVMs. Recent improvements in the encoding step such as higher order VLAD [55] or democratic aggregation [56], could further improve the writer identification rates. The current high identification rates also suggest the need for larger datasets. This would also widen the scope for techniques relying on more training data such as convolutional neural networks.

Acknowledgments

This work has been supported by the German Federal Ministry of Education and Research (BMBF), grant-nr. 01UG1236a. The contents of this publication are the sole responsibility of the authors.

References

[1] A. Brink, J. Smit, M. Bulacu, L. Schomaker, Writer Identification Using Directional Ink-Trace Width Measurements, Pattern Recognition 45 (1) (2012) 162–171. 1

[2] T. Gilliam, R. Wilson, J. Clark, Scribe Identification in Medieval English Manuscripts, in: Pattern Recognition (ICPR), 2010 20th International Conference on, Istanbul, 2010, pp. 1880–1883. 1

[3] D. Fecker, A. Asit, V. Märgner, J. El-Sana, T. Fingscheidt, Writer Identification for Historical Arabic Documents, in: Pattern Recognition (ICPR), 2014 22nd International Conference on, Stockholm, 2014, pp. 3050–3055. 1

[4] M. Bulacu, L. Schomaker, Text-Independent Writer Identification and Verification Using Textural and Allographic Features, Pattern Analysis and Machine Intelligence, IEEE Transactions on 29 (4) (2007) 701–717. 1, 2

[5] C. Djeddi, L.-S. Meslati, I. Siddiqi, A. Ennaji, H. E. Abed, A. Gattal, Evaluation of Texture Features for Offline Arabic Writer Identification, in: Document Analysis Systems (DAS), 2014 11th IAPR International Workshop on, Tours, 2014, pp. 8–12. 1, 10, 11

[6] S. He, L. Schomaker, Delta-n Hinge: Rotation-Invariant Features for Writer Identification, in: Pattern Recognition (ICPR), 2014 22nd International Conference on, Stockholm, 2014, pp. 2023–2028. 1, 2, 10

[7] S. Fiel, R. Sablatnig, Writer Identification and Writer Retrieval using the Fisher Vector on Visual Vocabularies, in: Document Analysis and Recognition (ICDAR), 2013 12th International Conference on, Washington DC, 2013, pp. 545–549. 1, 2, 3, 7, 9, 10, 11

[8] S. He, M. Wiering, L. Schomaker, Junction Detection in Handwritten Documents and its Application to Writer Identification, Pattern Recognition 48 (12) (2015) 4036–4048. 1, 3, 11

[9] R. Jain, D. Doermann, Writer Identification Using an Alphabet of Contour Gradient Descriptors, in: Document Analysis and Recognition (ICDAR), International Conference on, Buffalo, 2013, pp. 550–554. 1, 2, 7, 9, 10

[10] R. Jain, D. Doermann, Combining Local Features for Offline Writer Identification, in: Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on, Heraklion, 2014, pp. 583–588. 1, 2, 9, 10, 11

[11] G. Louloudis, B. Gatos, N. Stamatopoulos, A. Papandreou, ICDAR 2013 Competition on Writer Identification, in: Document Analysis and Recognition (ICDAR), 2013 12th International Conference on, Washington DC, 2013, pp. 1397–1401. 1, 7, 9, 10

[12] A. J. A. Newell, L. D. L. Griffin, Writer Identification Using Oriented Basic Image Features and the Delta Encoding, Pattern Recognition 47 (6) (2014) 2255–2265. 1, 2

[13] L. Schomaker, M. Bulacu, Automatic Writer Identification Using Connected-Component Contours and Edge-Based Features of Uppercase Western Script, Pattern Analysis and Machine Intelligence, IEEE Transactions on 26 (6) (2004) 787–798. 1

[14] X. Wu, Y. Tang, W. Bu, Offline Text-Independent Writer Identification Based on Scale Invariant Feature Transform, Information Forensics and Security, IEEE Transactions on 9 (3) (2014) 526–536. 1, 2, 3

[15] D. A. Reynolds, T. F. Quatieri, R. B. Dunn, Speaker Verification Using Adapted Gaussian Mixture Models, Digital Signal Processing 10 (1-3) (2000) 19–41. 1, 2, 4

[16] W. M. Campbell, D. E. Sturim, D. A. Reynolds, Support Vector Machines Using GMM Supervectors for Speaker Verification, Signal Processing Letters, IEEE 13 (5) (2006) 308–311. 1, 5

[17] T. Bocklet, A. Maier, E. Nöth, Age and Gender Recognition for Telephone Applications Based on GMM Supervectors and Support Vector Machines, in: Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on, Las Vegas, 2008, pp. 1605 – 1608. 1

[18] R. Arandjelovic, A. Zisserman, Three Things Everyone Should Know to Improve Object Retrieval, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, Providence, 2012, pp. 2911–2918. 1, 3

[19] D. G. Lowe, Distinctive Image Features from Scale-Invariant Keypoints, International Journal of Computer Vision 60 (2) (2004) 91–110. 1, 3

[20] T. Malisiewicz, A. Gupta, A. A. Efros, Ensemble of Exemplar-SVMs for Object Detection and Beyond, in: Computer Vision (ICCV), IEEE International Conference on, Barcelona, 2011, pp. 89–96. 2, 6, 8

[21] M. Juneja, A. Vedaldi, C. V. Jawahar, A. Zisserman, Blocks that shout: Distinctive parts for scene classification, in: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, Portland, 2013, pp. 923–930. 2

[22] V. Christlein, D. Bernecker, F. Hönig, E. Angelopoulou, Writer Identification and Verification Using GMM Supervectors, in: Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on, 2014, pp. 998–1005. 2, 7, 8, 9, 10, 11

[23] X. Zhou, X. Zhuang, H. Tang, M. Hasegawa-Johnson, T. S. Huang, Novel Gaussianized Vector Representation for Improved Natural Scene Categorization, Pattern Recognition Letters 31 (8) (2010) 702–708. 2, 5

[24] S. A. Mahmoud, I. Ahmad, W. G. Al-Khatib, M. Alshayeb, M. Tanvir Parvez, V. Märgner, G. A. Fink, KHATT: An Open Arabic Offline Handwritten Text Database, Pattern Recognition 47 (3) (2014) 1096–1112. 2, 7

[25] F. Slimane, S. Awaida, ICFHR2014 Competition on Arabic Writer Identification Using AHTID/MW and KHATT Databases, in: Frontiers in Handwriting Recognition (ICFHR), 14th International Conference on, Heraklion, 2014, pp. 797 – 802. 2

[26] J. Sánchez, F. Perronnin, T. Mensink, J. Verbeek, Image Classification with the Fisher Vector: Theory and Practice, International Journal of Computer Vision 105 (3) (2013) 222–245. 2, 4, 5, 7

[27] F. Slimane, V. Märgner, A New Text-Independent GMM Writer Identification System Applied to Arabic Handwriting, in: Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on, Heraklion, 2014, pp. 1–6. 2

[28] A. Schlapbach, H. Bunke, Off-line Writer Identification and Verification Using Gaussian Mixture Models, in: S. Marinai, H. Fujisawa (Eds.), Machine Learning in Document Analysis and Recognition, Vol. 90 of Studies in Computational Intelligence, Springer Berlin Heidelberg, 2008, pp. 409–428. 2

[29] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, C. Schmid, Aggregating Local Image Descriptors into Compact Codes, Pattern Analysis and Machine Intelligence, IEEE Transactions on 34 (9) (2012) 1704–1716. 2, 3, 5

[30] D. Bertolini, L. Oliveira, E. Justino, R. Sabourin, Texture-based Descriptors for Writer Identification and Verification, Expert Systems with Applications 40 (6) (2013) 2069–2080. 3

[31] A. Schlapbach, M. Liwicki, H. Bunke, A Writer Identification System for On-line Whiteboard Data, Pattern recognition 41 (7) (2008) 2381–2397. 3

[32] A. Busch, W. W. Boles, S. Sridharan, Texture for Script Identification, Pattern Analysis and Machine Intelligence, IEEE Transactions on 27 (11) (2005) 1720–1732. 3

[33] D. C. Smith, K. A. Kornelson, A Comparison of Fisher Vectors and Gaussian Supervectors for Document Versus Non-document Image Classification, in: SPIE 8856, Applications of Digital Image Processing XXXVI, Vol. 8856, San Diego, CA, 2013, pp. 88560N–88560N–12. 3, 5

[34] R. Arandjelovic, A. Zisserman, All About VLAD, in: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, Portland, OR, 2013, pp. 1578 – 1585. 3, 5, 7

[35] V. Christlein, C. Riess, J. Jordan, C. Riess, E. Angelopoulou, An Evaluation of Popular Copy-Move Forgery Detection Approaches, Information Forensics and Security, IEEE Transactions on 7 (6) (2012) 1841–1854. 3

[36] P. H. Gosselin, N. Murray, H. Jégou, F. Perronnin, Revisiting the Fisher Vector for Fine-grained Classification, Pattern Recognition Letters 49 (2014) 92–98. 3

[37] T. Kobayashi, Dirichlet-based Histogram Feature Transform for Image Classification, in: Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, Columbus, 2014, pp. 3278–3285. 3

[38] A. Bosch, A. Zisserman, X. Mu, X. Munoz, Image Classification Using Random Forests and Ferns, in: Computer Vision (ICCV), IEEE 11th International Conference on, Rio de Janeiro, 2007, pp. 1–8. 3

[39] X. Peng, L. Wang, X. Wang, Y. Qiao, Bag of Visual Words and Fusion Methods for Action Recognition: Comprehensive Study and Good Practice, arXiv preprint arXiv:1405.4506. 4, 5

[40] A. Dempster, N. Laird, D. Rubin, Maximum Likelihood from Incomplete Data via the EM Algorithm, Journal of the Royal Statistical Society. Series B (Methodological) 39 (1) (1977) 1–38. 4

[41] M. Xu, X. Zhou, Z. Li, B. Dai, T. S. Huang, Extended Hierarchical Gaussianization for Scene Classification, in: Image Processing (ICIP), 2010 17th IEEE International Conference on, Hong Kong, 2010, pp. 1837–1840. 5

[42] K. Chatfield, V. Lempitsky, A. Vedaldi, A. Zisserman, The Devil is in the Details: an Evaluation of Recent Feature Encoding Methods, in: J. Hoey, S. McKenna, E. Trucco (Eds.), British Machine Vision Conference, BMVA Press, Dundee, 2011, pp. 76.1–76.12. 5

[43] J. Delhumeau, P.-H. Gosselin, H. Jégou, P. Pérez, Revisiting the VLAD Image Representation, in: Multimedia (MM), 21st ACM international conference on, ACM Press, Barcelona, 2013, pp. 653–656. 5

[44] T. Kobayashi, Three Viewpoints Toward Exemplar SVM, in: Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on, 2015, pp. 2765–2773. 6

[45] B. Hariharan, J. Malik, D. Ramanan, Computer Vision – ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part IV, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, Ch. Discriminative Decorrelation for Clustering and Classification, pp. 459–472. 6

[46] M. Gharbi, T. Malisiewicz, S. Paris, F. Durand, A Gaussian Approximation of Feature Space for Fast Image Similarity, Tech. Rep. MIT-CSAIL-TR-2012-032 (2012). 6

[47] J. Zepeda, P. Pérez, Exemplar SVMs as Visual Feature Encoders, in: Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on, 2015, pp. 3052–3060. 6

[48] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, C.-J. Lin, LIBLINEAR: A Library for Large Linear Classification, Journal of Machine Learning Research 9 (2008) 1871–1874. 6

[49] G. Louloudis, B. Gatos, N. Stamatopoulos, ICFHR2012 Competition on Writer Identification Challenge 1: Latin/Greek Documents, in: Frontiers in Handwriting Recognition (ICFHR), 2012 International Conference on, Bari, 2012, pp. 829–834. 7

[50] F. Kleber, S. Fiel, M. Diem, R. Sablatnig, CVL-DataBase: An Off-Line Database for Writer Retrieval, Writer Identification and Word Spotting, in: Document Analysis and Recognition (ICDAR), 2013 12th International Conference on, Washington DC, 2013, pp. 560 – 564. 7

[51] N. Otsu, A Threshold Selection Method from Gray-Level Histograms, Systems, Man, and Cybernetics, IEEE Transactions on 9 (1) (1979) 62–66. 7

[52] A. Vedaldi, B. Fulkerson, VLFeat - An Open and Portable Library of Computer Vision Algorithms, in: Multimedia, International Conference on, ACM, Firenze, 2010, pp. 1469–1472. 7

[53] U. V. Marti, H. Bunke, The IAM-database: An English Sentence Database for Offline Handwriting Recognition, International Journal on Document Analysis and Recognition 5 (1) (2002) 39–46. 9

[54] G. Louloudis, N. Stamatopoulos, B. Gatos, ICDAR 2011 Writer Identification Contest, in: Document Analysis and Recognition (ICDAR), 2011 International Conference on, Beijing, 2011, pp. 1475–1479. 9

[55] X. Peng, L. Wang, Y. Qiao, Q. Peng, Boosting VLAD with Supervised Dictionary Learning and High-Order Statistics, in: D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (Eds.), Computer Vision – ECCV 2014, Vol. 8691 of Lecture Notes in Computer Science, Springer International Publishing, Zurich, 2014, pp. 660–674. 11

[56] H. Jégou, A. Zisserman, Triangulation Embedding and Democratic Aggregation for Image Search, in: Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, Columbus, 2014, pp. 3310–3317. 11