Writer Identification in Handwritten Musical Scores with Bags of Notes

Albert Gordo*, Alicia Fornés, Ernest Valveny
Computer Vision Center, Universitat Autònoma de Barcelona, Barcelona, Spain

Abstract

Writer Identification is an important task for the automatic processing of documents. However, the identification of the writer in graphical documents is still challenging. In this work, we adapt the Bag of Visual Words framework to the task of writer identification in handwritten musical scores. A vanilla implementation of this method already performs comparably to the state-of-the-art. Furthermore, we analyse the effect of two improvements of the representation: a Bhattacharyya embedding, which improves the results at virtually no extra cost, and a Fisher Vector representation, that very significantly improves the results at the cost of a more complex and costly representation. Experimental evaluation shows results more than 20 points above the state-of-the-art in a new, challenging dataset.

Keywords:
Writer Identification, Handwritten Musical scores, Bag of Notes

*Corresponding author

Email addresses: agordo@cvc.uab.es (Albert Gordo), alicia.fornes@cvc.uab.es (Alicia Fornés), ernest.valveny@cvc.uab.es (Ernest Valveny)

1. Introduction

Document Image Analysis and Recognition (DIAR) is an important field in Pattern Recognition, whose aim is to analyze contents of document images. One of the most active research areas of DIAR is handwriting analysis, which covers the tasks of recognition, interpretation, identification and verification of documents.

The aim of writer identification is to determine the author of a piece of handwriting from a set of writers. Writer Identification is an important task for the automatic processing of documents, allowing, for instance, the analysis of digital libraries, where document classification is performed based on the authorship of the document. It can also be applied to forensic document examination, where the personal identification is based on handwriting individuality instead of face, fingerprint, iris or voice.

While in handwriting recognition and interpretation it is important to filter out the variations in handwriting style in order to determine the meaning, these variations in handwriting style are fundamental for handwriting identification and verification. Hence, there are two important factors in writer identification: individual characteristics (within-writer variability) and class characteristics (between-writers variation). The goal is to find an optimal trade-off between intra-class compactness (minimizing individual characteristics) and inter-class separability (maximizing class characteristics).

Writer identification in handwritten text documents has been extensively researched [1, 2, 3, 4, 5]. However, the identification of the writer in graphical documents is still challenging. Graphical documents use graphical languages, which consist of symbols and combination rules, to describe ideas in a compact way. In this scenario, the identification of the writer can be performed by analyzing the shape of the symbols appearing in graphic documents.

Music Scores are a particular kind of graphic document, which include not only graphic elements, but also text (e.g. tempo markings, lyrics, *etc.*). Although the research is mainly focused on Optical Music Recognition (OMR) [6, 7, 8], writer identification in music scores seems to have awaken some interest [9, 10, 11]. In fact, many historical archives contain a large number of musical compositions without information about the composer. In this sense, the research in writer identification can help musicologists in two different ways: First, when dealing with original drafts (manuscripts written by the composer, not the scribe or copyist), in addition to melody, harmony and rhythm, scholars in musicology also analyze the handwriting style by comparing the shape of music symbols (notes, clefs, rests, accidentals, *etc.*). Second, in case of manuscripts that are written by scribes of copyists, the identification of the writer can help in the geographic location and dating of the music composition as a first step in the identification process. In both cases, the handwriting analysis is a time consuming task (especially when dealing with big databases), and for this purpose, a system for automatic writer identification in music scores could be very useful.

Although several methods for writer identification of music scores exist in the literature [12, 11, 13], we will show in Section 4 that their performance drops significantly in a challenging dataset such as the *CVC-MUSCIMA* dataset [14] recently released.

In a previous paper [15], we presented a method based on the Bag of Visual Words (BOV) framework –called the Bag of Notes (BON) approach–

that achieved very good performance in a simpler dataset [12]. In this paper, we will apply some modifications to the BON approach of [15] that permit to keep a high accuracy in the new *CVC-MUSCIMA* dataset. We will start with a vanilla version of the method, using a hard codebook obtained with k-means and a linear classifier. Results on the small dataset of [12] are very similar to those obtained with the more elaborated schema of [15], which uses probabilistic codebooks and an RBF kernel, and are comparable to the state-of-the-art. Then, we will improve the basic version and analyse the effect of the modifications in the newer, more challenging dataset of [14]. This basic version has been improved in two different ways: first, we show how square-rooting this vanilla descriptor (which, as noted in [16] corresponds to an explicit embedding of the Battacharyya kernel) can substantially improve the vanilla results at almost zero cost. Second, we extend the method to make use of the Fisher Vector (FV) framework [17]. In a nutshell, the FV can be seen as an extension of the BOV; the FV encodes not only the number of features assigned to each word, but also higher order statistics as the position and the sparseness. This framework has been successfully used in different applications such as image classification [18], retrieval [19], handwritten word spotting [20], document classification [21], or aesthetics assessing [22]. In the experiments we will see that the modified BON approach can achieve 99% accuracy in the new challenging dataset, whereas the performance of all other methods is significantly lower.

The rest of the paper is organized as follows. In Section 2 we review existing work on writer identification of music scores. The Bag of Notes approach is described in Section 3; first we describe the vanilla version, and then we extend it using the Battacharyya embedding and the Fisher Kernel framework. Experimental results are presented and discussed in Section 4. Finally, Section 5 concludes the paper and proposes some future work.

2. Previous Work

To the best of our knowledge, the first attempts of writer identification on music scores correspond to the *eNoteHistory* project [23, 24, 9], in which the researchers developed a prototype for analyzing the music score and then extracted structural information of the music symbols and notes. However, no quantitative results have been published, and as far as we know, this work has not been continued.

In [10, 12], two different writer identification approaches for old handwritten music scores were presented, inspired on some writer identification methods applied to text documents. Results were reported on a small dataset of 20 different writers and 200 images in total [12]. The first method extracts features for every music line, whereas the second one extracts textural features from music textures. The experimental results showed that both methods achieved similar identification rates, 75% and 73% respectively. The authors also presented a combination of both approaches [25], obtaining a significant increase in the classification accuracy, with a final score of 92%. However, the adaptation of writer identification approaches for text to graphic documents does not seem to be the best choice, and specific methods designed for graphical documents may obtain better results.

As a matter of fact, the previous results were outperformed by a novel and more simple method, which was based on the analysis of the shape of music symbols [13]. It consists in the analysis of the shape of certain music symbols (clefs and notes) by the use of symbol recognition methods. Experimental results on the same dataset demonstrated that this approach is not only simpler than the previous ones (which are based on the adaptation of text to graphic documents), but also the results are slightly outperformed (93%).

As far as we know, the best results on writer identification for musical scores in the recent literature for the dataset of [12] are those published in [11] and [15]. In [11], a Self Organizing Maps (SOM) quantization of the musical symbols is used to construct a vocabulary where features are assigned. The descriptors are weighted with *tf-idf*, and a cosine similarity is then used along with a nearest neighbour classifier. In [15], upon which this work stands, pages are represented by a Bag of Notes, which is an adaptation from the well known Bag of Visual Words (BOV) framework [26, 27]. In a nutshell, features are represented using the Blurred Shape Model (BSM) descriptor [28]. Then, a probabilistic codebook is built using a Gaussian Mixture Model and soft assignment is used to represent the musical scores. Finally, they are classified using a Support Vector Machine (SVM) with an radial basis function kernel. These papers report scores between 96% and 97% in the aforementioned dataset.

Recently, a writer identification competition in music scores has been proposed [29]. The competition was tested on the newer *CVC-MUSCIMA* database [14], a more challenging dataset containing music scores from 50 different writers. The participants' methods were based on the computation of edge-based directional probability distribution features, grapheme features or run-length-based probability distribution features. Concerning classifica-

Table 1: Methods existing in the literature.

Method	Features	Classification	Observations
FLSB 2008 [10, 25]	Lines	k-NN	Text-based approach
FLSB 2009 [12]	Textures	k-NN	Text-based approach
FLSOB 2010 [25]	Lines-Textures	k-NN	Text-based approach
FL [13]	BSM and DTW-based	k-NN	Music approach
MMS 2010 [11]	Zoning	SOM	Text-based approach
GFVL 2010 [15]	BSM	Bag of features	Music approach
HA 2011 [29]	edge-based / grapheme	k-NN	Arabic-based approach
DS 2011 [29]	run-lengths	k-NN/ SVM / Multilayer Perceptron	Arabic-based approach

tion, their approaches used k nearest neighbor (k-NN), SVM, and multilayer perceptron classifiers. All these approaches had been applied to arabic writer identification tasks and were adapted to music scores for this competition. The best results obtained a writer identification rate of 77% in the *CVC-MUSCIMA* dataset. We summarize all these existing methods in Table 1.

3. The Bag of Notes Approach

In this section, we first show a short description of the well-known BOV framework for visual categorization. Then, we explore how this method can be easily adapted to represent music scores aimed at writer-identification, and show two improvements that significantly increase the accuracy of the method.

3.1. Bag of Visual Words

In the Bag of Visual Words (BOV) framework [26, 27], images are represented by a histogram of quantized local features. First, interest points are detected in the images, and local features, such as SIFT, are extracted from the images at those points. A visual vocabulary or codebook is then learned, offline, by clustering these local features. The *de facto* method to cluster these features is k-means. Images are then represented by a fixed-length histogram that counts the number of local features assigned to each of the codebook centres. These image descriptors can then be used for categorization tasks (typically using SVMs), retrieval, *etc*.

Several improvements to this framework have been proposed for all the stages, from interest points detection and description to vocabulary construction, whole image representation, and classification.

Through the rest of this section, we will first show how the BOV framework can be adapted in a very natural way to represent musical scores for tasks such as writer identification. Then, we will show some modifications that will significantly improve the model at a reasonable cost.

3.2. Vanilla Bag of Notes

In a BOV, images are described by the frequency of quantized low level features. In the same way, a musical score could be represented by the frequency of different features that appear in it *i.e.*, the different music symbols along their writer-dependant variations. Note that, as opposed to natural images, where spatial information is very important, such information should be avoided when representing musical scores for writer identification. Indeed, even if spatial information would be interesting for a melodic or harmonic analysis, it is not desirable for writer identification tasks, since the same writer could transcribe a wide variety of music styles. The fact that we are interested in the appearance and shape of the symbols but not in its position is one of the main indicators that the Bag of Notes can be a good approach to this problem.

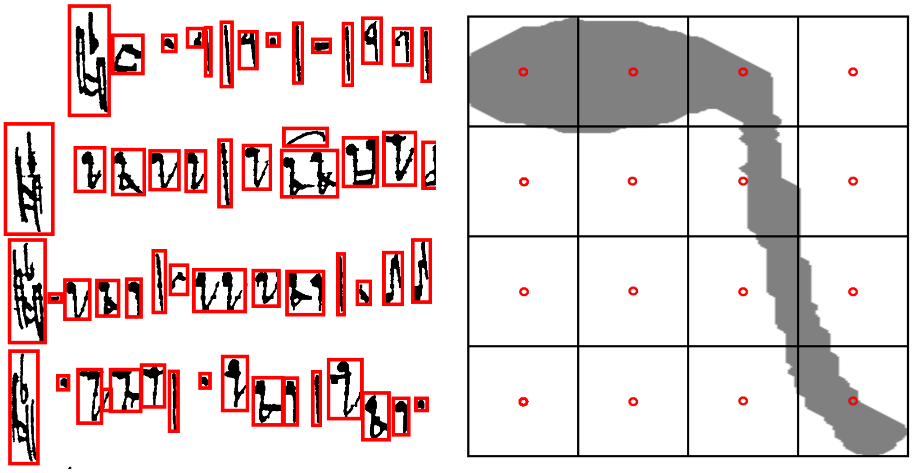
In the BOV framework, interest points are selected according to heuristics or just densely sampled, without any semantic consideration, and relying on the statistical nature of the framework to compensate for the likely mistakes. In an analogous way, in our BON we avoid performing a costly and error-prone symbol analysis such as the one of [13], and consider any remaining component after staff removal as an interest component (see Fig. 1a). This will include symbols as notes, clefs, accidentals, dynamics, *etc.*, but may also include noise as well as broken symbols, staff remains, *etc.* We will encode all the components without distinction and rely on the statistical nature of the method to weight them accordingly to their importance.

In the BOV framework, interest points have traditionally been represented with SIFT descriptors. SIFT produces an edge descriptor that is robust to residual small distortions, and is generally invariant to rotation and scaling, which makes it ideal for complex image representation. However, when trying to represent the structure of objects (such as musical symbols), descriptors such as SIFT or SURF are not the best option, and descriptors that capture some structure in a natural way are usually a better choice. In our case, we use the Blurred Shape Model (BSM) [28]. This descriptor was shown in [28] to perform better than SIFT and other descriptors in the task of symbol classification, and is very compact and fast to compute. The BSM is also scale invariant, which is a useful property in this writer identification task. Furthermore, using BSM allows us to compare our method with other approaches that used the BSM descriptor in a fairer way.

The BSM encodes the spatial probability of appearance of the shape pixels and their context information in the following way: The image is divided in a grid of $n \times n$ equal-sized subregions, and each bin receives votes from the shape points in it and also from the shape points in the neighboring bins. Thus, each shape point contributes to a density measure of its bin and its neighboring ones (see Fig 1b). The output descriptor is a vector histogram where each position corresponds to the amount of shape points in the context of the sub-region. The resulting vector histogram, obtained by processing all feature points, is normalized in the range [0..1] to obtain the probability density function (pdf) of $n \times n$ bins. In this way, the output descriptor represents a distribution of probabilities of the object shape considering spatial distortions. As a result, a robust technique in front of noise and elastic deformations is obtained.

These feature vectors should then be clusterized, *e.g.* with k-means or a Gaussian Mixture Model (GMM), to obtain a codebook and then represent the images as in the original BOV framework. Finally, these representations are classified with a classifier such as an SVM. A flowchart of the process can be seen in Figure 2.

We would like to note that, at least intuitively, this representation could be understood as counting the frequency of certain music symbols in the page, and, as such, would be more useful for rhythm analysis purposes than for writer identification. However, we believe it is important to remark that this representation can indeed be used for writer identification: i) since we have many more vocabulary words than music symbols, we will have several words representing different styles of each possible symbol. ii) In one of the improvements we propose, we go beyond counting, and capture higher order



(a) (b)

Figure 1: a) Detected symbols. Every connected component is considered as a valid symbol. That may include spurious noise, combined notes, etc. b) A note adjusted in a BSM grid. Every black pixel of the note contributes to all the surrounding centroids of the grid in a soft manner.

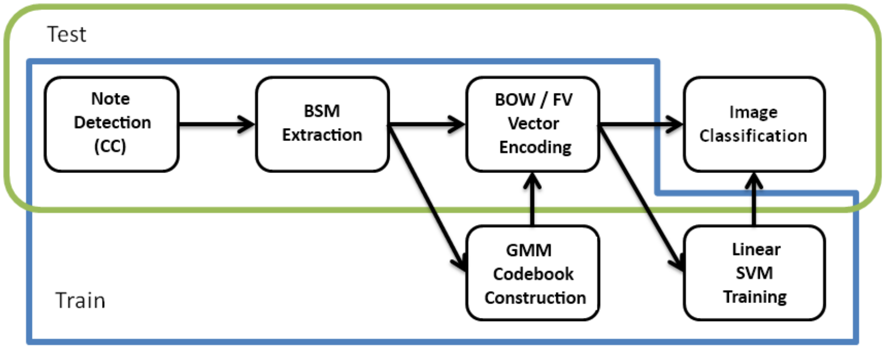


Figure 2: Flowchart of the basic Bag of Notes framework

statistics such as the position and sparseness of the symbols with respect to the vocabulary. This captures what makes the symbols different, and so it contains much more discriminative information. iii) By using a classifier such as a SVM, we expect to learn what makes authors different. This last part is particularly important. As we will see during the experimental section, methods using classifiers such as k-NN, where no learning has been performed, obtain much worse results than those using classifiers that involve learning.

3.3. Improving the vanilla Bag of Notes

In this subsection, we deal with ways to improve this vanilla BON at a reasonable cost.

Bhattacharyya explicit embedding: One typical way to improve the results in the BOV framework consists in using non-linear kernels such as Bhattacharyya, intersection or χ^2. However, using a linear kernel allows us to use solvers based on Stochastic Gradient Descent or the cutting-plane algorithm, which drastically reduce the training time of the classifier. In [16, 30], different ways to explicitly embed the data are proposed. In this case, using a linear kernel in the embedded space approximately corresponds to using the non-linear kernel in the original space. In particular, both [16, 30] note how square-rooting the original feature vectors corresponds to an exact explicit embedding for the Bhattacharyya kernel, *i.e.*, if $k_b(x, y) = \sqrt{x'y}$ and $\phi(x) = \sqrt{x}$, then $\phi(x)'\phi(y) = \sqrt{x}'\sqrt{y} = \sqrt{x'y} = k_b(x, y)$. Square rooting the feature vectors has some important properties, such as reducing the weight of the most common words, which can lead to very noticeable improvements. This is an extremely simple and effective embedding, and, as shown in [16], the results are competitive with other more complex kernels / embeddings such as χ^2 or intersection. As we will see in the experimental section, simply square-rooting the feature vectors before classification will provide a very substantial gain in the accuracy.

We note that, in general, square-rooting the vectors has been known as an ad-hoc way to improve the results when dealing with BOW histograms. However, in [31] it was shown that square-rooting the vectors can be understood as approximating a non-iid (independently and identically distributed) BOW model. In an nutshell, the BOW models assume that the words (or image patches, music symbols, *etc.*) are iid. However, this is an oversimplification, since in practice the data is not iid: if we randomly covered some parts of an image (*cf*. Fig 1 of [31]), we would still be able to approximately guess their contents based on the surrounding information. Similarly, if on one of our music scores we observe some symbols that have details that distinguish one of the writers, then we expect other symbols in that page to also have those details. By square-rooting the vectors, we are approximating a model that does *not* assume that the data is iid, and so is more consistent with the data we have. We believe this is particularly important in our case. In standard BOV applications, every image contains thousands or tens of thousands of patches, and so the abundance of data can compensate for the defects of the model. In our case, however, the number of components per page is limited (from a few tens to a hundred) and so having a more realistic model becomes more important.

The Fisher Vector: The Fisher Kernel [32] is a well-known kernel based on the Fisher information matrix that exploits generative information of the model. Intuitively, it describes how a new sample should be adapted in order to better fit the model. The Fisher Kernel considers information about how every patch differs from the learned codebook with respect to the means and variances, and so it encodes precisely what makes a particular patch different from the "average" patches. This seems particularly relevant to our interests, since capturing what makes similar symbols different is one of the important things that define the writing styles. Unfortunately, calculating the Fisher Kernel is a costly operation and its direct application is generally unfeasible.

One possible solution to overcome this problem is the Fisher Vector (FV) framework [17, 18]. The FV corresponds to an explicit embedding of the Fisher Kernel in the particular case of a Gaussian Mixture Model generative model. In this case, it encodes not only the frequencies of the words as in the BOV, but also their *position* and *sparseness* with respect to the means and variances of the codebook words in a probabilistic way. This produces a very high dimensional signature with great discriminative power, even for very small codebooks. In the very recent [33], the FV encoding was shown superior to other state-of-the-art encoding methods such as the Super-Vector encoding (SV) or Locality-constrained Linear Coding (LLC).

Here, we will first review the Fisher Kernel in its natural form of [32], and then we will follow [17] to show the closed form of the FV in the case of using a Gaussian Mixture Model generative model.

Let $T = \{x_1, x_2, \ldots, x_T\}$ be the set of T local descriptors from an image. We assume that the generation process of X can be modeled by a probability density function u_λ with parameters λ. X can be described by the gradient vector $G_\lambda^X = \frac{1}{T}\nabla_\lambda \log u_\lambda(X)$.

The gradient of the log-likelihood describes the contribution of the parameters to the generation process. The dimensionality of this vector depends only on the number of parameters in λ, not on the number of patches T. A natural kernel on these gradients is [32]:

$$K(X,Y) = G_\lambda^{X'} F_\lambda^{-1} G_\lambda^Y \tag{1}$$

where F_λ is the Fisher information matrix of u_λ:

$$F_\lambda = E_{x\sim u_\lambda}\left[\nabla_\lambda \log u_\lambda(x) \nabla_\lambda \log u_\lambda(x)'\right]. \tag{2}$$

As F_λ is symmetric and positive definite, it has a Cholesky decomposition $F_\lambda = L'_\lambda L_\lambda$ and $K(X,Y)$ can be rewritten as a dot-product between normalized vectors $\mathcal{G}_\lambda$, with $\mathcal{G}_\lambda^X = L_\lambda G_\lambda^X$:

We will refer to $\mathcal{G}_\lambda^X$ as the Fisher Vector of X. Learning a kernel classifier using the kernel (1) is equivalent to learning a linear classifier on the Fisher vectors $\mathcal{G}_\lambda^X$.

We follow [17] and choose u_λ to be a Gaussian mixture model (GMM):

$$u_\lambda(x) = \sum_{i=1}^{k} w_i u_i(x). \tag{3}$$

We collectively denote $\lambda = \{w_i, \mu_i, \Sigma_i, i = 1, \ldots, K\}$ where w_i, μ_i, and Σ_i are respectively the mixture weight mean vector and covariance matrix of Gaussian u_i. We assume that the covariance matrices are diagonal and we denote by σ_i^2 the variance vector. The GMM u_λ is trained on a large number of images using Maximum Likelihood (ML) estimation. It is supposed to describe the content of any image. We assume that the x_t's are generated independently by u_λ and therefore:

$$G_\lambda^X = \frac{1}{T}\sum_{t=1}^{T} \nabla_\lambda \log u_\lambda(x_t). \tag{4}$$

We consider the gradient with respect to the mean and standard deviation parameters (the gradient with respect to the weight parameters bring little additional information).

Let D denote de dimensionality of the descriptors x_t, let $\gamma_t(i)$ be the occupancy probability of descriptor x_t to Gaussian i, and let $\mathcal{G}^X_{\mu,i}$ (resp. $\mathcal{G}^X_{\sigma,i}$) be the D-dimensional gradient with respect to the mean μ_i (resp. standard deviation σ_i) of Gaussian i. Mathematical derivations [17] lead to:

$$\mathcal{G}^X_{\mu,i} = \frac{1}{T\sqrt{w_i}} \sum_{t=1}^{T} \gamma_t(i) \left(\frac{x_t - \mu_i}{\sigma_i} \right), \mathcal{G}^X_{\sigma,i} = \frac{1}{T\sqrt{2w_i}} \sum_{t=1}^{T} \gamma_t(i) \left[\frac{(x_t - \mu_i)^2}{\sigma_i^2} - 1 \right], \tag{5}$$

where the division between vectors is as a term-by-term operation. The final gradient vector $\mathcal{G}^X_\lambda$ is the concatenation of the $\mathcal{G}^X_{\mu,i}$ and $\mathcal{G}^X_{\sigma,i}$ vectors for $i = 1 \ldots K$ and is therefore $2KD$-dimensional.

In [18], the Fisher Kernel framework was further improved for categorization tasks. It was shown that an $L2$ normalization of the $\mathcal{G}^X_\lambda$ vectors reduces the weight of the background in the signatures and yields better classification results. It was also shown that a by-dimension square root of the signature vectors will help to ‘unsparsify’ the signatures and will therefore obtain better results when using the dot-product as a similarity measure. Both improvements will be used in our experiments.

4. Experiments

Through this section we will analyse the performance of the vanilla Bag of Notes method as well as the proposed improvements: Bhattacharyya explicit embedding and FV embedding.

First, we will compare the performance of the vanilla Bag of Notes with the other methods on the small dataset of [12]. As we will see, the vanilla Bag of Notes already produces very competitive results. Then, we will report results on the *CVC-MUSCIMA* dataset and show how the improvements lead to better accuracy scores. Finally, we will compare our results with the state-of-the-art in music score writer identification in the *CVC-MUSCIMA* dataset.

4.1. Datasets

For our experiments, we will use two different datasets:

The *small* dataset introduced in [12]. This dataset consists of 200 music pages from 20 different writers. These pages are extracted from a collection of music scores of the 17th, 18th and 19th centuries, which have been obtained from two archives in Catalonia, Spain: the archive of Seminar of Barcelona and the archive of Canet de Mar. This dataset has been commonly used in the past, and most of the methods report results in it.

We would like to make two remarks about this dataset. First, one particular piece is never transcribed by more than one author. Second, in some cases, more than one of the pages transcribed by one particular author belong to one particular music piece, *i.e.*, some pieces span through many pages. This has rised some criticism in the past, since it is not clear what it is being evaluated: who transcribed the page, or the rhythm of the page. Indeed, a method that ignored the writing style and focused on the rhythm would still score high in this dataset, independently of the writer.

The standard evaluation procedure of this dataset is a 5-fold partition. Each test partition contains exactly one document per writer (20 documents total) and the train partition contains the remaining documents. This is repeated 5 times with different partitions and the results are averaged.

The *CVC-MUSCIMA* dataset [14]**.** This dataset contains 20 different musical sheets of different pieces and styles, transcribed by 50 different writers, making a total of 1,000 binary images. As opposed to the previous dataset, in this case all the writers transcribe the same pieces. The images are presented in two flavours, with and without staff lines. This is to ensure that writer identification results are not dependant of the quality of the staff removal algorithms used. In our case, we will only use the staffless images.

The dataset also defines two sets of partitions of the data, each one containing ten folds:

Set A, or constrained. In the first set of partitions, the training pieces of a given fold are the same for each writer, and so none of the pieces of the test set have been used during the training stage. If the first music page of one writer is in the train set of a given fold, all the first music pages of the remaining writers will also be in the train set of that particular fold.

Set B, or unconstrained. In the second set of partitions this constraint is not satisfied, and pieces that appear in the training set of one author will appear in the test set of a different one (for example, the first music page will appear in the train set of one author and in the test set of another.

These partitions are particularly devised to attest that we are indeed performing writer identification instead of rhythm classification. Indeed, if the method was performing rhythm classification, it is reasonable to think that, in set B, unconstrained, test pieces from one author would be matched with the exact same pieces that appear in the train set of a different author, and so the classification results would be significantly lower than on set A, where this confusion is not possible. At the same time, a writer identification rate in set B similar to the one in set A will show that the system is classifying according to the handwriting style and not being particularly affected by the kind of music notes and symbols appearing in the music sheet.

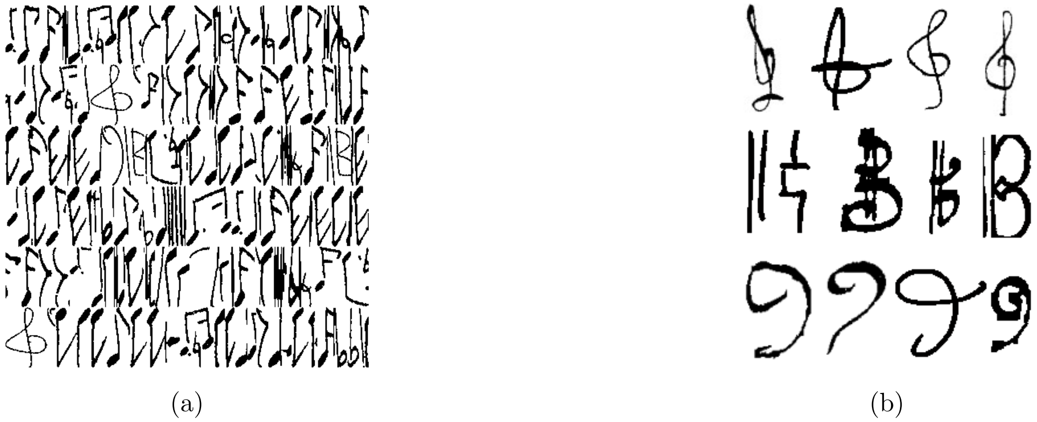
In each partition, 50% of the documents of each writer belong to the training set and the other 50% belong to the test set. Furthermore, effort has been put in guaranteeing that each piece appears approximately 50% of the time in training and 50% in test.

Evaluation on the *CVC-MUSCIMA* dataset will be done averaging the accuracy results of the ten defined folds.

4.2. Reference methods

Our approach has been compared with some of the methodologies in the literature that have shown the best results. The one proposed in [12] is a symbol-independent method, which adapts the writer identification method for text documents described in [34] to music scores. Both approaches treat the writer identification task as a texture identification problem. For this purpose, textures are generated from the original document, and the following textural features are extracted: Gabor filters and Grey-Scale Co-occurrence Matrices (GSCM). The authors showed in [12] that the *Resize* music textures (which means taking randomly music symbols and resizing them to a fixed height) obtained the best results of the four different possibilities when generating textures. For this reason, we have also generated the Resize textures from the music scores database (see Fig.3a).

The method proposed in [13] is, as far as we know, the only existing symbol-dependent writer identification approach in music scores. It is based



(a)

(b)

Figure 3: (a) Resize Texture image. The music symbols are randomly taken from the music page and put in a reference line with the same inter-symbol distance. The symbols are resized to a fixed size, without the preservation of the aspect ratio in the resizing process. For this reason, some symbols are distorted. (b) Music Clefs. The first row corresponds to treble clefs, the second row to alto clefs, and the last row shows some bass clefs.

on the detection of the music clefs that are appearing in the document, and then, comparing these clefs with the ones existing in the database. Since the authors claim that the music clefs (see Fig.3b) are the music symbols with the highest discriminative power (they look like drawings or signatures), they concentrate on the detection, recognition and extraction of the music clefs. For this purpose, the user must first manually segment one instance of each one of the three music clefs (treble, alto and bass clef). Afterwards, the system combines two symbol recognition methods (a DTW-based one, and the BSM descriptor) for detecting and segmenting the similar shapes existing in the database (the supposed music clefs). Once the clefs are extracted, the

Blurred Shape Model (BSM) descriptor is used for extracting the features, and finally these features are compared (using a k-NN classifier) with the features of the clefs in the database for the final identification. It must be noted that both our BON approach and the one described in [13] use the BSM descriptor.

The method proposed in [11] is quite similar in nature to the Bag of Notes approach, although it was published afterwards. Symbols are extracted with connected component analysis and quantized with Self Organizing Maps (SOM) and *tf-idf* weighting. Finally, nearest neighbor with a cosine similarity is used to classify the samples. Note however that SOM requires to set more parameters than k-means or GMM, and it is more sensitive to them. From the paper, it is not clear how the authors tuned the parameters.

In the recent writer identification competition in music scores [29], two different groups participated by sending a total of 8 different approaches. The first set of methods submitted by Hassaïne *et al.* were based on the computation of several sets of features. The first set uses the edge-based directional probability distribution features (see [35]). The second set uses grapheme features (which are fully described in [36]). A third set is composed of the combination of both kinds of features, edge-based and grapheme-based features. These methods have previously been applied for Arabic writer identification and for signature verification and have shown interesting results. The classification step is performed either using a logistic regression classifier or a k-NN algorithm.

The second participant group was composed of Djeddi and Souici-Meslati. The proposed methods (more details in [37]) are based on the computation of run-lengths features, which are determined on the binary image taking into consideration the pixels corresponding to the ink trace. The probability distribution of white run-lengths has been used in the writer identification experiments. There are four scanning methods: horizontal, vertical, left-diagonal and right-diagonal. The authors calculate the run-lengths features using the grey level run-length matrices and the histogram of run-lengths is normalized and interpreted as a probability distribution. For the classification stage, the authors have used k-NN, SVM (one *vs*. one, and one *vs*. all), and multilayer perceptron classifiers, as well as a combination of them.

The first group obtained the best performance in this competition with the combination of edge-based and grapheme features, followed by the second participant group with the combination of classifiers approach. We will compare our method with these two approaches.

4.3. Implementation details

To extract the low level features, we obtained the connected components of the binary image and represented them with a BSM descriptor. In the dataset of [12] we will use BSM descriptors of size 8×8 as in [15]. For the experiments on *CVC-MUSCIMA* we analyse the effect of the descriptor sizes and report the results. The BSM descriptors were later normalized using the $L1$ norm. We observed no significant difference when using other norms such as $L2$ or $L1_{sqrt}$. Although we experimented with other descriptors such as SIFT, we found the results with BSM to be significantly better. This is reasonable since BSM is designed to represent shapes, while SIFT is not.

The k-means and GMM clusterings were performed using INRIA's Yael library [1]. When computing the vanilla BON, the final image descriptors are normalized with the $L1$ norm. When computing the FV, we will apply two of the improvements presented in [18], namely power normalization and $L2$ normalization.

Classification will be made using a one *vs.* all SVM. Since we always use a linear kernel, we will use a solver optimized for linear problems. Particularly, we will use LIBLINEAR [2], which makes use of the cutting-plane algorithm and dramatically improves the training speed of the SVM. To set the C trade-off cost of the SVM classifier, we used the same heuristic used by the SVM$^{\text{light}}$ [3] suite. Given a set of N training vectors $X = \{x_1, x_2, \ldots, x_N\}$, we set $k = \frac{1}{N}\sum_{i=1}^{N} x_i x_i'$, and then $C = 1/k^2$. This heuristic gave excellent classifications results, *i.e.*, manually setting this parameter did not bring any substantial improvement.

When replicating the experiments with the Gabor filters and the GSCM, we have used the same parameters as in [12].

Unfortunately, we were not able to replicate the results of [11]. We believe the reason is that we have not been able to correctly tune the many parameters of the SOM (number of iterations, step size, bandwidth of the Gaussian, *etc.*), even when trying to do so over the test set. Instead of SOM clustering, we have used a k-means clustering. We would like to note that the method of [11] does not exploit the structural information of the SOM, and, therefore, the results of the method using SOM should be quite similar

[1] Software available at `https://gforge.inria.fr/projects/yael`

[2] http://www.csie.ntu.edu.tw/ cjlin/liblinear/

[3] http://svmlight.joachims.org/

to the results using k-means.

4.4. Results and analysis

We first report the results on the *small* dataset on Table 2. Except for our Bag of Notes and the reimplemented [11], all the results are quoted from the respective papers. We can observe how the vanilla Bag of Notes performs similarly to the best available methods. We can also observe how the reimplemented [11] using k-means obtains very similar results to those of the original method. The small difference can be due to the clustering, but also due to differences in the training protocol, low-level descriptors, normalization, *etc.*, since the authors of [11] did not disclose all the details of their evaluation protocol. Finally, we can observe how the proposed improvements obtain a 100% accuracy on this small dataset. Therefore, we will rely on the more complex *CVC-MUSCIMA* dataset to study their properties and differences.

Table 2: Comparison of results on the *small* dataset.

Method	Accuracy
Vanilla BON (128 centres)	97
BON + Bhattacharyya (128 centres)	100
FV (64 Gaussians)	100
Soft BON (128 Gaussians, unsupervised clustering) [15]	96
Soft BON (640 Gaussians, supervised clustering) [15]	97
Textural [12]	73
Music clefs [13]	93
SOM (900 centres) [11]	97
SOM (900 centres) [11] (reimplemented with k-means)	96

Tables 3 to 5 show the results of the vanilla Bag of Notes as well as the different improvements on the *CVC-MUSCIMA* dataset. Table 3 shows the results of the vanilla setup as a function of the number of words and the BSM descriptor size for both variants of the dataset. Table 4 with the Bhattacharyya embedding and Table 5 with the FV. We can draw the following conclusions from the proposed methods:

Influence of the descriptor size: In general, for BON and BON plus Bhattacharyya, descriptors of sizes 8×8 and 12×12 work better than smaller descriptors of size 4×4. The differences between 8×8 and 12×12 are usually small, and do not follow a clear pattern. We will favor 8×8 since it is faster to compute and clusterize.

In the case of FV, 8×8 seems significantly better than 4×4. Note however that, in the FV, the size of the descriptor directly affects the final signature length. Therefore, it is still convenient to use descriptors of size 4×4, since using a descriptor of size 8×8 would increase the length of the final signature by a factor of 4. In general, increasing the number of Gaussians produces better results than increasing the size of the BSM when aiming at a fixed number of dimensions.

Influence of the sets: The differences between the results of the constrained and unconstrained sets are very small. This suggests that the rhythm has not been taken into account by the classifier and that we are indeed learning writing style characteristics.

Bhattacharyya embedding *vs.* linear kernel: The use of the Bhattacharyya embedding provides a very significant improvement over the vanilla version at essentially zero cost. This is not surprising since this embedding is approximating a non-iid model, which is more consistent with the data than the vanilla BON.

Fisher Vector *vs*. linear kernel and Bhattacharyya embedding: The FV significatively improves both the vanilla and the Bhattacharyya versions for a given vocabulary size, and equals the best results of the Bhattacharyya embedding with as few as 16 Gaussians. Moreover, the FV captures information not available in the Bag of Notes, which can be particularly important in the writer identification context. However, note that using the FV can be more costly than the Bhattacharyya embedding: the FV descriptor is more complex, and requires to compute a GMM vocabulary instead of a simple k-means. The operations involved in computing the FV (*cf*. equation (5)), although simple, are more costly than just counting words and square-rooting. Finally, the Bhattacharyya vectors are sparse, while the FV is high-dimensional and dense. The complexity of many classification methods (including SVMs) depends on the number of non-sparse elements of the vectors, and so training and testing with Bhattacharyya vectors can be faster than with FVs. Therefore, the Bhattacharyya embedding can be a very reasonable alternative in situations where that complexity cannot be afforded.

4.5. Comparison with the state of the art

We now report the results that our implementations of the reference methods achieve in the *CVC-MUSCIMA* dataset.

In table 6 the writer identification results for the textural methods of [12] are shown. One can see that the best identification rate is about 25%, and it decreases to about 15% when using the independent set. These results demonstrate that the textural approach is not a good choice when using

Table 3: Hard assignment. Mean classification accuracy as a function of the number of clusters and the descriptor size.

N. Clusters		16	32	64	128	256	512
Set A (const.)	4 × 4	31.2	44.1	59.2	74.2	86.3	93.3
	8 × 8	32.2	43.7	59.6	78.1	89.2	93.6
	12 × 12	31.7	45.2	57.7	76.2	88.6	93.9
Set B (unconst.)	4 × 4	32.1	45.1	60.0	74.9	85.0	91.4
	8 × 8	31.0	46.6	61.3	78.3	88.3	93.5
	12 × 12	32.8	46.5	59.0	76.9	87.4	93.4

Table 4: Hard assignment with a Bhattacharyya explicit embedding. Mean classification accuracy as a function of the number of clusters and the descriptor size.

N. Clusters		16	32	64	128	256	512
Set A (const.)	4 × 4	33.9	48.8	68.2	83.4	93.0	97.1
	8 × 8	33.3	49.6	70.5	87.1	95.2	97.8
	12 × 12	34.5	51.8	68.7	86.8	94.8	97.8
Set B (unconst.)	4 × 4	30.4	47.1	66.4	81.9	90.9	94.5
	8 × 8	31.0	48.4	67.9	85.1	93.9	96.9
	12 × 12	33.4	48.2	66.0	84.1	93.0	96.6

bigger datasets. In fact, the results reported in [12] demonstrated that this approach had low scalability degree (from a writer identification rate of 92% with 5 writers, it decreased to 73% with 20 writers). Moreover, the results also show that it is sensitive to the rhythm of the composition, because it decreases 10 points (from 25% to 15%) when using the unconstrained set. It

Table 5: Fisher Vector. Mean classification accuracy as a function of the number of clusters and the descriptor size.

N. Clusters		4	8	16	32	64
Set A (const.)	4 × 4	79.6	91.5	96.9	98.5	99.3
	8 × 8	89.5	94.9	97.2	99.1	99.7
Set B (unconst.)	4 × 4	82.7	93.3	97.2	98.2	99.3
	8 × 8	91.2	95.5	97.7	98.9	99.5

must be said that this important dependency on the kind of symbols in the music page was already expected. The reason is quite obvious: the visual appearance of a texture image highly depends on the kind of symbols that have been used for generating the image. In other words, if we take two music pages written by the same writer but with different music symbols (which means different rhythm), and generate the music texture images, we will observe that these two texture images look extremely different.

Table 6: Textural Approach [12]. Mean classification accuracy using Gabor features, GSCM features, and both, for different number of neighbors (k-NN).

Sets	Features	5-NN	7-NN	9-NN	11-NN	13-NN
Set A (const.)	Gabor	20.0	21.2	22.0	23.4	21.8
	GSCM	22.0	23.8	25.6	24.8	24.8
	Both	20.8	24.0	24.60	25.2	25.4
Set B (unconst.)	Gabor	12.6	12.2	13.6	14.0	13.4
	GSCM	10.6	12.2	13.0	13.6	14.6
	Both	13.4	14.0	15.0	14.2	16.6

Table 7 show the writer identification rates of the approach proposed in [13] when applied to our database (and using a BSM with a grid of 25x25). We can see that the best results are obtained in the dependent set (with a writer identification rate of about 73%) are quite close to the ones obtained in the independent set (about 70%), demonstrating that this approach is not rhythm dependent. These results show that the use of symbol-dependent methods (specially music clefs) can help in the identification of the writer, although they require a user for manually selecting the three music clefs for each writer in the database.

Table 7: Music clefs Approach [13]. Mean classification accuracy using the music clefs and the BSM descriptor for different number of neighbors (k-NN).

Sets	3-NN	5-NN	7-NN	9-NN
Set A (const.)	72.1	72.9	72.9	72.3
Set B (unconst.)	69.5	69.9	69.7	68.9

Table 8 shows the results of our implementation of [11] using k-means instead of SOM. If we focus on the constrained set, we can observe how the results are more than 10 points below our vanilla implementation. We believe this is due using a k-NN classifier instead of a SVM. If we focus on the unconstrained set, were confusions are possible, we observe how the results drop drastically. Again, this is most likely due to using a k-NN classifier instead of a SVM.

For comparison purposes, we show on Table 9 recomputed results of BON, BON plus Bhattacharyya and FV on both sets using a k-NN classifier with a dot product / cosine similarity measure instead of a SVM, and compare it to the method of [11]. We can observe how FV is the best performing method (suggesting that writer information is more explicit in the FV representation, as expected since it encodes higher order statistics), but also how there is a huge drop on all the methods when using a k-NN classifier, particularly in the unconstrained set. This shows that albeit the descriptor does contain information about the writing style, it is of paramount importance to learn what makes writers different with classifiers such as SVM.

Table 8: Mean classification accuracy as a function of the number of clusters in our reimplementation of the method of [11]. We use k-means instead of SOM.

Sets	128	256	512	1024
Set A (const.)	49.4	64.0	74.7	81.2
Set B (unconst.)	19.9	28.3	36.0	44.8

Finally, Table 10 summarizes all the results. We can highlight the following points:

- As a general rule, the methods that directly adapt text approaches to this new context do not perform as well as methods designed for graphics. One exception are the methods of the ICDAR challenge [29], which outperform the music clefs method of [13]. Note however that the challenge methods combine several types of features and classifiers, while the music clefs method uses a simple k-NN classifier.

- Methods that perform well on the constrained set can perform badly

Table 9: Results of several methods using k-NN instead of SVM for a fixed output dimensionality. BON and [11] use the cosine as a similarity measure. BON + Bhattacharyya and FV use the dot product.

Sets	Method	128D	256D	512D
Set A (const.)	BON	61.1	71.4	77.7
	BON + Bhattacharyya	65.8	76.7	83.2
	FV	64.4	78.4	87.3
	[11] (reimplemented with k-means)	49.4	64.0	74.7
Set B (unconst.)	BON	32.7	39.1	45.8
	BON + Bhattacharyya	24.0	29.1	35.9
	FV	42.1	46.7	54.0
	[11] (reimplemented with k-means)	19.9	28.3	36.0

on the unconstrained one. Even if some writer style information is obviously being learned in those cases, it is also clear that rhythm information prevails, leading to a severe drop of the results in the unconstrained set where confusions are possible.

- Using classifiers such as SVM or using distances not based on whole page representation, such as the clef distance of [13], seem to be key points in obtaining good results in the unconstrained set.

- The proposed BON methods significantly outperform all the other methods when using the unconstrained, more difficult set.

- FV leads to almost perfect results. However, a high-dimensionality price is paid for just a small improvement. If such precision is not needed, the Bhattacharyya embedding can lead to excellent results at a fraction of the cost.

Table 10: Comparison of results on the *CVC-MUSCIMA* dataset. Methods marked with a $*$ come from the challenge of [29], and were evaluated over only one fold of the dataset.

Method	Set A	Set B
BON (512 clusters, 512D)	93.6	93.5
BON (512 clusters, 512D) + Bhattacharyya Embedding	97.8	96.9
Fisher Vector (BSM 4×4, 64 clusters, 2048D)	99.3	99.3
Fisher Vector (BSM 8×8, 64 clusters, 8192D)	99.7	99.5
Textural (13NN) [12]	25.4	16.6
Music clefs (7NN) [13]	72.9	69.7
SOM (1024D, reimplemented with k-means) [11]	81.2	44.8
PRIP02-combination [29]	-	77.0*
TUA03-SVMOAA [29]	-	76.6*

5. Conclusions

In this work, we have adapted the Bag of Visual Words framework to the task of writer identification in handwritten musical scores. A vanilla implementation of this method already performs comparably to the state-of-the-art, and further improvements yield results more than 20 points above the current state-of-the-art methods in a new, challenging dataset.

Moreover, we have shown the importance of learning classifiers when using representations that encode the whole musical score page. The Fisher Vector representation at 512 dimensions drops from a 96.9% to a 87.3% in the constrained set and from a 97.2% to a 54.0% in the unconstrained set when switching from an SVM classifier to a k-NN classifier.

We believe that these descriptors represent the rhythm of the document in a natural way, and so it is of paramount importance to unveil the writer identification information hiding in the descriptors using classifiers such as the SVM.

Finally, this opens other future work lines such as writer retrieval. Indeed, based on the results we have observed, directly using these descriptors for retrieval would lead to unsatisfying results. Frameworks such as Metric Learning could significantly help in this task.

References

[1] L. Schomaker, M. Bulacu, Automatic writer identification using connected-component contours and edge-based features of uppercase western script, IEEE *Transactions on Pattern Analysis and Machine Intelligence* 26 (2004) 787–798.

[2] A. Bensefia, T. Paquet, L. Heutte, A writer identification and verification system, *Pattern Recognition Letters* 26 (2005) 2080–2092.

[3] A. Schlapbach, H. Bunke, A writer identification and verification system using hmm based recognizers, *Pattern Analysis and Applications* 10 (2007) 33–43.

[4] A. Schlapbach, M. Liwicki, H. Bunke, A writer identification system for on-line whiteboard data, *Pattern Recognition* 41 (2008) 2381–2397.

[5] I. Siddiqi, N. Vincent, Text independent writer recognition using redundant writing patterns with contour-based orientation and curvature features, *Pattern Recognition* 43 (2010) 3853–3865.

[6] W. Homenda, Optical music recognition: the case study of pattern recognition, in: *Computer Recognition Systems*, 2005, pp. 835–842.

[7] L. Pugin, J. Hockman, J. Burgoyne, I. Fujinaga, Gamera versus aruspix. two optical music recognition approaches, in: *Proceedings of the International Conference on Music Information Retrieval*, 2008, pp. 419–424.

[8] C. Dalitz, M. Droettboom, B. Pranzas, I. Fujinaga, A comparative study of staff removal algorithms, IEEE *Transactions on Pattern Analysis and Machine Intelligence* 30 (2008) 753–766.

[9] I. Bruder, T. Ignatova, L. Milewski, Integrating knowledge components for writer identification in a digital archive of historical music scores, in: *Proceedings of the Joint ACM/IEEE Conference on Digital Libraries*, 2004, p. 397.

[10] A. Fornés, J. Lladós, G. Sánchez, H. Bunke, Writer identification in old handwritten music scores, in: *Proceedings of the International Workshop on Document Analysis Systems*, 2008, pp. 347 – 353.

[11] S. Marinai, B. Miotti, G. Soda, Bag of characters and som clustering for script recognition and writer identification, in: *Proceedings of the International Conference on Pattern Recognition*, 2010, pp. 2182–2185.

[12] A. Fornés, J. Lladós, G. Sánchez, H. Bunke, On the use of textural features for writer identification in old handwritten music scores, in:

Proceedings of the International Conference on Document Analysis and Recognition, 2009, pp. 996–1000.

[13] A. Fornés, J. Lladós, A symbol-dependent writer identification approach in old handwrittenmusic scores, in: *Proceedings of the International Conference on Frontiers in Handwriting Recognition*, 2010, pp. 634–639.

[14] A. Fornés, A. Dutta, A. Gordo, J. Lladós, CVC-MUSCIMA: a ground truth of handwritten music score images for writer identification and staff removal, *International Journal on Document Analysis and Recognition* 15 (2012) 243–251. http://www.cvc.uab.es/cvcmuscima/.

[15] A. Gordo, A. Fornés, E. Valveny, J. Lladós, A bag of notes approach to writer identification in old handwritten musical scores, in: *Proceedings of the International Workshop on Document Analysis Systems*, 2010, pp. 247–254.

[16] F. Perronnin, J. Sánchez, Y. Liu, Large-scale image categorization with explicit data embedding, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2297–2304.

[17] F. Perronnin, C. Dance, Fisher kernels on visual vocabularies for image categorization, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[18] F. Perronnin, J. Sánchez, T. Mensink, Improving the fisher kernel for large-scale image classification, in: *Proceedings of the European Conference on Computer Vision*, 2010, pp. 143–156.

[19] F. Perronnin, Y. L., J. Sánchez, H. Poirier, Large-scale image retrieval with compressed fisher vectors, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 3384–3391.

[20] F. Perronnin, J. Rodriguez-Serrano, Fisher kernels for handwritten word-spotting, in: *Proceedings of the International Conference on Document Analysis and Recognition*, 2009, pp. 106–110.

[21] A. Gordo, F. Perronnin, A bag-of-pages approach to unordered multi-page document classification, in: *Proceedings of the International Conference on Pattern Recognition*, 2010, pp. 1920–1923.

[22] L. Marechesotti, F. Perronnin, D. Larlus, G. Csurka, Assessing the aesthetic quality of photographs using generic image descriptors, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[23] N. Luth, Automatic identification of music notations, in: *Proceedings of the International Conference on Web Delivering of Music*, 2002, pp. 203–210.

[24] R. Göcke, Building a system for writer identification on handwritten music scores, in: *Proceedings of the IASTED International Conference on Signal Processing, Pattern Recognition, and Applications*, pp. 250–255.

[25] A. Fornés, J. Lladós, G. Sánchez, X. Otazu, H. Bunke, A combination of features for symbol-independent writer identification in old music scores,

International Journal on Document Analysis and Recognition 13 (2010) 243–259.

[26] J. Sivic, A. Zisserman, Video google: A text retrieval approach to object matching in videos, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2003, pp. 1470–1477.

[27] G. Csurka, C. Dance, L. Fan, J. Willamowski, C. Bray, Visual categorization with bags of keypoints, in: *Workshop on Statistical Learning in Computer Vision, ECCV*, 2004, pp. 1–22.

[28] S. Escalera, A. Fornés, O. Pujol, P. Radeva, G. Sánchez, J. Lladós, Blurred shape model for binary and grey-level symbol recognition, *Pattern Recognition Letters* 30 (2009) 1424–1433.

[29] A. Fornés, A. Dutta, A. Gordo, J. Lladós, The icdar 2011 music scores competition:staff removal and writer identification, in: *Proceedings of the International Conference on Document Analysis and Recognition*, 2009, pp. 1511–1515.

[30] A. Vedaldi, A. Zisserman, Efficient additive kernels via explicit feature maps, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 3539–3546.

[31] R. G. Cinbis, J. Verbeek, C. Schmid, Image categorization using Fisher kernels of non-iid image models, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[32] T. Jaakkola, D. Haussler, Exploiting generative models in discriminative classifiers, in: *Advances in Neural Information Processing Systems*, 1999, pp. 487–493.

[33] K. Chatfield, V. Lempitsky, A. Vedaldi, A. Zisserman, The devil is in the details: an evaluation of recent feature encoding methods, in: *Proceedings of the British Machine Vision Conference*, 2011, pp. 76.1–76.12.

[34] H. Said, T. Tan, K. Baker, Personal identification based on handwriting, *Pattern Recognition* 33 (2000) 149–160.

[35] S. Al-Ma'adeed, E. Mohammed, D. Al Kassis, Writer identification using edge-based directional probability distribution features for arabic words, in: *Proceedings of the IEEE/ACS International Conference on Computer Systems and Applications*, 2008, pp. 582–590.

[36] S. Al-Ma'adeed, A. Al-Kurbi, A. Al-Muslih, R. Al-Qahtani, H. Al Kubisi, Writer identification of arabic handwriting documents using grapheme features, in: *Proceedings of the IEEE/ACS International Conference on Computer Systems and Applications*, 2008, pp. 923–924.

[37] D. Chawki, S. Labiba, A texture based approach for arabic writer identification and verification, in: *Proceedings of the International Conference on Machine and Web Intelligence*, 2010, pp. 115–120.