Nonparametric Estimation of Fisher Vectors to Aggregate Image Descriptors *

Hervé Le Borgne Pablo Muñoz Fuentes

Abstract

We investigate how to represent a natural image in order to be able to recognize the visual concepts within it. The core of the proposed method consists in a new approach to aggregate local features, based on a non-parametric estimation of the Fisher vector, that result from the derivation of the gradient of the loglikelihood. For this, we need to use low level local descriptors that are learned with independent component analysis and thus provide a statistically independent description of the images. The resulting signature has a very intuitive interpretation and we propose an efficient implementation as well. We show on publicly available datasets that the proposed image signature performs very well.

1 Introduction

Contemporary works on image classification (but also for object recognition and image retrieval) showed the efficiency of approaches based on the computation of local features (such as SIFT descriptors [14]).However, since the number of points of interest may vary from one image to another, the dimension of the vector representing an image (*i.e* number of keypoints $\times$ local feature size) is not constant, making it unusable as input of usual machine learning algorithms used in the image classification paradigm. This drawback was circumvented through the *bag-of-visterm* (BOV) approach [20]. It consists in pre-computing a codebook of visual words then coding each image according to this visual vocabulary. The simplest approach consists here in *hard quantization* that associate each local feature to one visual word(s) of the dictionary. Significant improvement is obtained by using a soft assignment scheme [5] or sparse coding [21, 1]. The BOV signature can be refined by the *spatial pyramid matching* (SPM) scheme [10] that add spatial information at several scales. It consists of averaging the local features according to hierarchical regular grids over the image (*average pooling*) although recent work showed that one may benefit to consider their maximum instead (*maximum pooling*) [21, 1].

An alternative to the BOV scheme was recently proposed to aggregate local descriptors. In [16], the visual vocabulary is modeled with a Gaussian mixture model (GMM) then a signature is derived according to the Fisher kernel principle, consisting

*The work described in this technical report was published in the proceedings of ACIVS 2011

in computing the gradient of the loglikelihood of the problem with respect to some parameters. It allows to take advantage of a generative model of data and discriminative properties at the same time. The VLAD descriptor [8] can be seen as a derivative of the Fisher kernel too . In practice, it consists of accumulating the difference between (each component of) the local descriptors and the visual words of the codebook. To be applied to image retrieval, these both signatures were compressed using principal component analysis and product kernel [8] or a simple binarization strategy [17].

The main contribution of this paper is to propose a non-parametric estimation of the Fisher kernel principle then derive an image signature that can be used for image classification.The main difference with the work of Perronnin [16, 17] is that he used a parametric method (GMM) to estimate the density of the visual words. Hence, two parameters affect its performance: (i) the number of Gaussian component and (ii) the parameters with respect to which the gradient is computed. By using a non-parametric estimation of the Fisher kernel principle, our approach circumvent these issues, and may as well provide a more accurate model of the loglikelihood.

However, the theoretical framework we proposed has to be applied to local features with statistically independent dimensions. Such descriptors were proposed in [11], who directly estimated the density of each dimension of the descriptors with parametric and non parametric methods. Our work differs from them since the non-parametric estimation is only a step in our framework, and the signature we finally obtain from the Fisher kernel derivation is different.

A second contribution of this work is to propose an efficient implementation of the image signature obtained according to the proposed theoretical framework. We propose some choices to make the computation of the signature simpler than in theory and show that classification results are maintained or even improved.

In section 2 we review the Fisher kernel framework, the non-parametric density estimation used and derive the proposed image signature. We present in section 3 some experimental results on several publicly available benchmarks and compare our approach to recent work in image scene classification.

2 Aggregating the image feature

2.1 Fisher kernel, score and vector

Let $X = \{x_t, t = 1 \dots T\}$ a set of vectors, for instance used to describe an image (*e.g* the collection of local features extracted from it). It can be seen as resulting from a generative probability model with density $f(X|\theta)$. To derive a kernel function from such a generative model, *i.e* being able to exhibit discriminative properties as well, Jaakola [7] proposed to use the gradient of the log-likelihood with respect to the parameters, called the *Fisher score*:

$$U_X(\underline{\theta}) = \nabla_{\underline{\theta}} \log f(X|\underline{\theta}) \tag{1}$$

This transforms the variable length of the sample X into a fixed length vector that can feed a classical learning machine. In the original work of [7] the Fisher information matrix F_λ is suggested to normalize the vector:

$$F_\lambda = E_X[\nabla_{\underline{\theta}} \log f(X|\underline{\theta}) \nabla_{\underline{\theta}} \log f(X|\underline{\theta})^T] \tag{2}$$

It then results into the Fisher vector:

$$G_X(\underline{\theta}) = F_\lambda^{-1/2} \nabla_{\underline{\theta}} \log f(X|\underline{\theta}) \tag{3}$$

2.2 Logspline density estimation

The traditional way of modeling a distribution density is to assume a classical parametric model such as normal, gamma or Weibull. For instance in [16], the vocabularies of visual words are represented with a Gaussian Mixture Models, for which the parameters (weight, mean and variance of each Gaussian) are estimated by maximum likelihood.

Alternatively, we can use a nonparametric estimate of the density, such as a histogram or a kernel-based method.A histogram density estimation can be seen as modeling the unknown log-density function by a piecewise constant function and estimating the unknown coefficients by maximum likelihood. In this vein, Kooperberg [9] proposed to model the log-density function by cubic spline (twice-continuously differentiable piecewise cubic polynomial), resulting into the so-called logspline density estimation.

More precisely, given the lower bound of data L and upper one U (L and U can be infinite), and a sequence of K values $t_1, \ldots, t_K$ such that $L < t_1 < \ldots < t_K < U$ (later referred as *knots*) and $K > 2$, we consider the space $\mathcal{S}$ consisting of the twice-continuously differentiable function f_s on (L, U), such that the restriction of f_s to each of the intervals $[t_1, t_2] \ldots [t_{K-1}, t_K]$ is a cubic polynomial and linear on $(L, t_1]$ and $[t_K, U)$. The functions of the K-dimensional space $\mathcal{S}$ are named natural (cubic) splines. Let $1, B_1, \ldots, B_p$ (with $p = K - 1$) a set of basis functions that span the space $\mathcal{S}$, chosen such that B_1 is linear with negative slope on $(L, t_1]$, $B_2, \ldots, B_p$ are constant on $(L, t_1]$, B_p is linear with positive slope on $[t_K, U)$ and $B_1, \ldots, B_{p-1}$ are constant on $[t_K, U)$. Given $\underline{\theta} = (\theta_1, \ldots, \theta_p) \in \mathbb{R}^p$ such that:

$$\int_L^U \exp\left(\theta_1 B_1(y) +, \ldots, \theta_p B_p(y) dy\right) < \infty \tag{4}$$

We can thus consider the exponential family of distribution based on this basis function:

$$f(y, \underline{\theta}) = \exp\left(\theta_1 B_1(y) +, \ldots, \theta_p B_p(y) - \mathcal{C}(\underline{\theta})\right) \tag{5}$$

Where $\mathcal{C}(\underline{\theta})$is a normalizing constant such that:

$$\int_{\mathbb{R}} f(y, \underline{\theta}) dy = 1 \tag{6}$$

As shown in [9], it is possible to determine the maximum likelihood estimate of $\underline{\theta}$ with a Newton-Raphson method with step-halving. They also proposed a knot selection methodology based on Akaike Information Criterion (AIC) to select the best model. It finally results into the estimation of the maximum likelihood estimate $\hat{\underline{\theta}}$ of the coefficients, the knots $t_1, \ldots, t_K$ and thus the wanted density.

Let notice that at convergence ($\underline{\theta} = \hat{\underline{\theta}}$), the loglikelihood is maximal, thus its derivative is null. Let $\{y_1 \dots y_n\}$ a random sample from f. We have:

$$\left.\frac{\partial \mathcal{L}(Y, \underline{\theta})}{\partial \theta_j}\right|_{\underline{\theta}=\hat{\underline{\theta}}} = 0 = \sum_{t=1}^{n} B_j(y_t) - n\frac{\partial \mathcal{C}(\hat{\underline{\theta}})}{\partial \theta_j} \tag{7}$$

Thus at convergence we have:

$$\frac{\partial \mathcal{C}(\hat{\underline{\theta}})}{\partial \theta_j} = E_f\left[B_j(y)\right] \tag{8}$$

Where $E_f[B(y)]$ denotes the Monte Carlo estimate of the expectation of $B(.)$ according to f. This will be used later in our development.

2.3 Signature derivation

Let consider that any image is described according to some D-dimensional vectors. Each feature dimension x^i can be thought of as arising as a random sample from a distribution having a density h^i. We can model the log-density function by a cubic spline, such as explained in section 2.2. Hence, it exists a basis $1, B_1^i, \dots, B_{p^i}^i$ of $\mathcal{S}$ such that:

$$h^i(x^i, \underline{\theta^i}) = exp\left(\sum_{j=1}^{p^i} \theta_j^i B_j^i(x^i) - \mathcal{C}^i(\underline{\theta^i})\right) \tag{9}$$

Let $\{y_t, t = 1 \dots T\}$ a set of vectors extracted from a given image, seen as T independent realizations of the D-dimensional random vector Y (for simplicity, Y denotes both the image and the corresponding random vector). The log-likelihood is thus:

$$\mathcal{L}(Y, \underline{\theta}) = \sum_{t=1}^{T} log\left(h(y_t, \underline{\theta})\right) \tag{10}$$

Where $h(y_t, \underline{\theta})$ denotes the density of Y. If one assumes the independence of all feature dimensions (this point is discussed in section 2.6), we have:

$$h(\underline{y_t}, \underline{\theta}) = \prod_{i=1}^{D} h^i(y_t^i, \underline{\theta^i}) \tag{11}$$

Thus:

$$\mathcal{L}(Y, \underline{\theta}) = \sum_{t=1}^{T}\sum_{i=1}^{D} log\left(h^i(y_t^i, \underline{\theta^i})\right) \tag{12}$$

Each density h^i can be estimated on the same basis as the one determined during learning. In other words, each h^i is expressed as in (9) with specific value for the coefficients θ_j^i. Hence, from (12) it follows:

$$\frac{\partial \mathcal{L}(Y, \underline{\theta})}{\partial \theta_j^i} = \sum_{t=1}^{T} B_j^i(y_t^i) - \frac{\partial \mathcal{C}^i(\underline{\theta^i})}{\partial \theta_j^i} \tag{13}$$

The first component of equation (13) is simply the expectation of $B_j^i(y)$ according to the density of the considered image (*i.e* estimated from the samples $\{y_t, t = 1 \dots T\}$). The second component is a function that depend on θ_j^i. If one assumes that the considered samples follow a probability law quite close to the one estimated during learning, we can apply equation (8), and finally:

$$\left.\frac{\partial \mathcal{L}(Y, \underline{\theta})}{\partial \theta_j^i}\right|_{\theta_j^i \approx \widehat{\theta_j^i}} = E_{h^i}\left[B_j^i(y)\right] - E_{f^i}\left[B_j^i(y)\right] \tag{14}$$

Where $h^i(.)$ is the density of the image descriptor and $f^i(.)$ the density class descriptor (dimension i), this last being estimated from local descriptor extracted from several learning images. The full gradient vector $U_Y(\underline{\theta})$ is a concatenation of these partial derivatives with respect to all parameters. Its number of components is $\sum_{i=1}^{D} p^i$, where p^i is the number of non-constant polynomial of the basis of $\mathcal{S}$ for dimension i.

The equation (14) leads to a remarkably simple expression, for which the physical interpretation is quite straightforward. It simply reflects the way a specific image (with density h^i) differs from the average world (*i.e* density f^i), *through* a well chosen polynomial basis, at each dimension (figure 1). The *average world* $\left(E_{f^i}\left[B_j^i(y)\right]\right)$ can be seen as a sort of codebook. If one uses linear polynomials $\left(B_j^i(y) = \alpha_j y^i\right)$, equation (14) relates to the VLAD signature, with an important difference since all vectors are used (i) during learning to estimate the codeword (ii) during test to compute the signature, while (i) K-means uses the closest vectors of a codeword (cluster center) to re-estimate it at each step (ii) VLAD uses as well only nearest neighbours to compute the signature component (see eq. (1) in [8]).

Figure 1: Example images from the *scene15* database (1^{st} column), with the response of an ICA filter (dimension i) to a particular image, *i.e* f^i in equation (14) (2^{nd} column) and the average density over the category, *i.e* h^i in equation (14) (3^{rd} column).

2.4 Signature normalization

In his seminal work, Jaakola [7] proposed to normalize the Fisher score by the Fisher information matrix. In [16], it was noted that such an operation improved the efficiency of the method in term of discrimination, by normalizing the dynamic range of the different dimensions of the gradient vector.

To normalize the dynamic range of each dimension of the gradient vector $U_Y(\underline{\theta})$ (each $U_Y(\theta_j^i)$ is given by the equation (14)), we need to compute the diagonal terms of the Fisher information matrix $F_{\underline{\theta}}$. Considering the expression of each $U_Y(\theta_j^i)$ given by he equation (14), the diagonal terms of $F_{\underline{\theta}}$ are:

$$F_{\theta_j^i} = E\left[\left(E_{h^i}\left[B_j^i(y)\right] - E_{f^i}\left[B_j^i(y)\right]\right)^2\right] \tag{15}$$

The dynamic range being computed on the learning database (density f^i), it is thus

the variance of $B_j^i(y)$. From equation (3) the final fisher vector is:

$$U_Y^n(\theta_j^i) = \frac{E_{h^i}\left[B_j^i(y)\right] - E_{f^i}\left[B_j^i(y)\right]}{\sigma_{f^i}\left[B_j^i(y)\right]} \tag{16}$$

Where $\sigma_{f^i}[.]$ is the standard deviation computed according to density f^i. Hence, our normalized signature $U_Y^n(\underline{\theta})$ can be regarded as the "standardizing" transformation of the raw description of the image given by the polynomial activity $B_j^i(y)$. The "normality" is here the learning database, from which we compute an average $E_{f^i}\left[B_j^i(y)\right]$ and a standard deviation $\sigma_{f^i}\left[B_j^i(y)\right]$ at each dimension.

2.5 Efficient implementation

We discuss in this section the choice of the knots $t_1, \ldots, t_K$ and the set of basis functions $1, B_1, \ldots, B_p$ (with $p = K-1$) that span the space $\mathcal{S}$ defined in section 2.2, used to compute our signature according to equation (14).

In his article on the logspline density estimation theory [9], Kooperberg proposed an automatic method to place the knots according to an AIC criterion. However, preliminary experiments have convinced us that such a process is not necessary, and that a simpler strategy is more efficient. For this, we fix a given number of shot and place them according to statistic order of the learning data. For instance, if $K = 9$, knots are places according to the decile of data. Hence, at each dimension, the amount of information is regularly distributed between knots. For low-level features such as those presented in section 2.6, the knots are approximately placed according to a logarithmic distribution.

Once the knots $(t_1, \ldots, t_K)$ are fixed, a natural choice for the set of basis functions $(1, B_1, \ldots, B_p)$ that span the space $\mathcal{S}$ is to consider the polynomials of the form:

$$\begin{array}{l} B_0(y) = 1 \\ B_1(y) = y \\ B_k(y) = \left(\frac{|y-t_k|+y-t_k}{2}\right)^3 \quad \text{for } k > 1 \end{array} \tag{17}$$

With such a basis, $B_0 = 1$ has no influence in the computation of the signature (see equation (14)), while $B_1(.)$ leads to compute the difference of the mean between the considered image (density h^i) and the class (density f^i). Further polynomials $B_k(.)$ are the positive part of a cubic function that is null at knot t_k.

Obviously, such a representation is strongly redundant. Hence, we propose two simplifications to this implementation. First, for polynomials B_k with $k > 1$, we only consider the values y between knot t_k and t_{k+1}, in order to avoid redundancy. The difference in equation (14) is thus computed on each interval defined by the knots. This can be seen as a sort of filter activity quantization, the limits of the cells being the knots. Secondly, for computational efficiency, we only consider the positive part

(absolute value) without computing the power three. The chosen basis thus becomes:

$$\begin{array}{l} B_0(y) = 1 \text{ (not used)} \\ B_1(y) = y \\ B_{k>1}(y) = \left\{ \begin{array}{ll} \frac{|y-t_k|+y-t_k}{2} & \text{for } y < t_{k+1} \\ 0 & \text{for } y > t_{k+1} \end{array} \right. \end{array} \tag{18}$$

Such an implementation is equivalent to compute only $(y-t_k)$ on the interval $[t_k, t_{k+1}]$ since the polynomial is null elsewhere and $y > t_k$ on the interval.

The third proposed simplification is to use a binary weighting scheme. It consists in not considering the value of $|y - t_k|$ in the computation but only its existence. In other word, one can only count $+1$ each time a pixel activity y is between t_k and t_{k+1}. Such a binary weighting scheme is commonly used in the design of BOV, in particular when the codebook is large [20].

2.6 Independent low level features

According to equation (11) the signature derivation requires to use independent low-level features, such that the image description density could be expressed as a factorial code. Such features can be obtained with Independent Component Analysis (ICA)[2, 6] that is a class of methods that aims at revealing statistically independent latent variables of observed data. In comparison, the well-known Principal Component Analysis (PCA) would reveal uncorrelated sources, *i.e* with null moments up to the order two only. Many algorithms were proposed to achieve such an estimation, that are well reviewed in [6]. These authors proposed the fast-ICA algorithm that searches for sources that have a maximal nongaussianity. When applied to natural image patches of fixed size (*e.g* $\Delta = 16 \times 16 = 256$), ICA results into a generative model composed of localized and oriented basis functions [6]. Its inverse, the separating matrix, is composed of *independent* filters $w_1, \ldots, w_D$ (size Δ) that can be used as feature extractors, giving a new representation with mutually independent dimensions. The number of filters (D) extracted by ICA is less or equal to the input data dimension (Δ). This can be reduced using a PCA previously to the ICA. The responses of the D filters to some pixels $(p_1, \ldots, p_T)$ of an image $I(.)$ are thus independent realizations of the D-dimensional random vector Y. As a consequence, the density can be factorized as expected:

$$h_{ica}\left(I(p_t)\right) = \prod_{i=1}^{D} h_{ica}^{i}\left(I(p_y)\right) = \prod_{i=1}^{D} w_i * I(p_t) \tag{19}$$

Where $*$ is the convolution product. These independent low-level features can be further used according to the method presented into section 2.3 since they verify equation (11).

3 Experiments

3.1 Datasets and experimental setup

The first dataset (*scene15*, see figure 1) [10] was recently used in several works on scene classification [15, 19, 18, 21, 1]. It is composed of 4485 images with 200 to 400 images for each category. A given image belong to one category exactly. The original sources of the pictures include the COREL collection, personal photographs and Google image search. We followed the experimental setup of[10] using 100 images per class for training and the rest for testing. As well, only the gray level of the images is considered, making color-based descriptors inoperable on this dataset. The performance is measured using the classification rate to be comparable to previous works.

The second dataset (*vcdt08*) was used as a benchmark in the Visual Concept Detection Task of the campaign ImageCLEF 2008 [3]. It is composed on 1827 images for training and 1000 for testing, all image being annotated according to 17 categories.The categories are not exclusive *i.e* an image can belong to several of these categories and sometimes to none of them. The performance is measures using the Equal Error Rate *i.e* the point such that the false acceptance rate is equal to the false rejection rate. Hence, this value is between 0 and 1 such that the smaller the EER the better the system: ideally, false acceptance and false rejection are null.

Two machine learning algorithms were used in the following experiments. the first is a SVM with a linear kernel [4] that has a fast convergence. We consider the use of a linear kernel as relevant because our method lead to a large signature (from hundreds to thousands dimensions) and that in such a high dimensional space, data are quite sparse and it is thus easier to find an hyperplane that separate them. The second learning algorithm is the fast shared boosting (FSB) [12] that is specifically designed to tackle the problem of overlapping classes (such as in the dataset *vcdt08*), by using weak classifiers that shares features among classes. Another benefit of this algorithm is its fast convergence when the umber of classes grows. When images can belong to several classes, it was shown that it has usually better results than a classic one-versus-all strategy.

3.2 Signature implementation

We study the effect of the simplifications proposed in section 2.5 to efficiently implement the proposed signature. In this experiment, only $D = 64$ filters are extracted with fastICA [6] from 40000 patches of size 16×16. Multi-class classification on non-exclusive classes is done with the FSB. All the experiments in this section were conducted with 32 knots.

The "normal" implementation uses the polynomial basis given at equation (17). Simplifications (section 2.5) include: (i) computing the polynomial *between knots* instead of *above* them (ii) use the *absolute value* instead of the *third power* (iii) in the case of the absolute value one can use a *binarized* weighting scheme instead of the *values*.

Table 1 shows the performances of all these possible implementations on the *vcdt08* benchmark. As expected, reducing information redundancy into the signature by com-

	Above knots	Between knots
$\|.\|^3$	0.294	**0.254**
$\|.\|$ val	0.286	0.257
$\|.\|$ bin	0.256	**0.254**

Table 1: Performances on the *vcdt08* benchmark for several implementations of our method (see text). The lower the EER, the better the method.

Signature	simple $E_{f^i}\left[B^i_j(y)\right]$	centered see (14)	normalized see (16)
64 filters	74.20	**74.30**	73.92
128 filters	75.58	75.71	**76.08**
256 filters	78.56	**78.86**	78.42

Table 2: Classification accuracy on *scene15* for three different implementation (see text). The equation number is given for the *centered* and the *normalized* signature. The *simple* one is only the first part of equation (14). Signatures were computed with 32 knots and no grid nor pyramid.

puting it *between knots* improve the performances. Concerning the other simplification, no significant effect is noticed for signatures computed *between knots*. However, when the signature is computed with the "above knots" implementation, the binarized weighting scheme allows to reach similar performances as those obtained with the non-redundant implementation.

Other variation in the implementation are possible, since we proposed a *centered* signature (equation 14) and a *normalized* one (equation 16). We also consider here a *simple* signature that only implement the first member of the equation (14), i.e a non-centered version of it. Since of the dimension has few influence on the FSB [12], we tested these three signatures as input of a SVM, on the *scene15* dataset (exclusive classes). As well, signatures were computed for three different sets of ICA filters, of size 64, 128 and 256, with 32 knots. Optimal parameters of the SVM were determined on the learning database, with a 5-fold cross validation, and data were previously scaled according to the method proposed by the author [4].

Results presented in table 2 show that results improve when more filters are used (see next section for a discussion on this point) but very few variation from one implementation to another. This is probably due to the data scaling of the SVM we used, that itself standardized the signatures.

3.3 Signature parameters

Above implementation details (previous section), the signature we propose mainly depends on two parameters, namely the number of filters D and the number of knots K. In this section we show the influence of these two parameters on the *scene15* dataset. We used a *simple* signature and computed the polynomial activity *between* knots with

	$K = 8$	$K = 16$	$K = 32$	$K = 64$
$D = 64$	71.55	**75.11**	74.20	73.47
$D = 128$	71.82	**75.88**	75.58	74.84
$D = 256$	76.78	78.49	**78.56**	77.69

Table 3: Classification accuracy on *scene15* for three filter basis of various size D and different number of knots K.Best result for a given D is marked in bold.

Figure 2: Average EER over the 17 categories of VCDT 2008: ranking of the proposed approach (white) with respect to the other 53 runs (blue)

a *binarized* weighting scheme. The signature size is $D \times K$. Optimal parameters of the SVM were determined on the learning database, with a 5-fold cross validation, and data were previously scaled as proposed in [4].

Results are presented in table 3. For a given number of knots K, the best results are always obtained with a maximal number of filters, that is consistent with the previous results. Indeed, with patches of size 16×16 the maximal number of filters is 256. To obtain a smaller collection of ICA filter, data is reduced with a PCA during the process (see section 2.6), thereby inducing a loss of information detrimental to the resulting descriptors.

For a given number of filter D, the best results are obtained with $K = 16$ or $K = 32$. As explained above, knots can be regarded as the limits of quantization cells of the ICA filter activity. Hence, when the number of knots is large, cells are too selective ($K = 64$) while they generalize too much for small number of knots ($K = 8$).

3.4 Comparison to other works

Table 4 shows the classification results for our method in comparison to recent works evaluated on the *scene15* dataset. Note that we considered only signatures on the full image for fair comparison. A SPM scheme, dividing the image according to a spatial grid at several scales, usually allows to improve the performances [10, 21, 1]. We used the same filters as previously with $K = 16$ knots to build the signature at each dimension. For [10] the results reported here are those with the "strong features" (SIFT), without spatial pyramid, i.e equivalent to a BOV approach. With 256 filters, our method achieve an accuracy of 78.5%, that is a gain between 2% and 6% over the best accuracies reported in the literature. Let also notice that with 64 filters our method achieve 75.1%, that is comparable to the state-of-the art.

The result we obtained on *vcdt08* (table 1) are competitive as well. On this benchmark, [12] reported an EER of 0.24 while we achieve 0.254 with 64 filters and 32 knots. In figure 2 we compare the average EER of our approach with the one of the runs submitted to the campaign [3]. Although the best run reported performs better, our method is ranked among the best.

Method	Accuracy %
Lazebnik *et al.* [10]	74.8
Liu *et al.* [13]	75.16
Rasiwasia *et al.* [19]	72.2
Rasiwasia *et al.* [18]	72.5
Masnadi-Shirazi *et al.* [15]	76.74
$U_{64}^{16}(ICA)$	75.10
$U_{128}^{16}(ICA)$	75.88
$U_{256}^{16}(ICA)$	**78.49**

Table 4: Classification accuracy for *scene15*. $U_D^K(ICA)$ is our method with D filters and K knots.

4 Conclusion

We proposed a non parametric estimation of the Fisher Kernel framework to derive an image signature. This high-dimensional and dense signature can reasonably feed a linear SVM. We showed the effectiveness of the approach in the context of image categorization. Experiments on challenging benchmark showed our approach outperforms recent techniques in scene classification [10, 13, 15, 19, 18]. Hence we report the higher accuracy on this state-of-the-art dataset in scene classification. We also reported competitive results on the vcdt08 benchmark of the ImageCLEF campaign [3].

In future work we would be interested in applying the non-parametric Fisher vector estimation presented in this paper on local features that only partially verify the property of statistical independence (see [11]) or not at all [14]. Another direction of research will concern the signature compression in order to apply it to image retrieval.

5 Acknowledgment

This work has been partially funded by I2S in the context of the project Polinum. We acknowledge support from the ANR (project Yoji) and the DGCIS for funding us through the regional business cluster Cap Digital (project Roméo)

References

[1] Boureau, Y.L., Bach, F., LeCun, Y., Ponce, J.: Learning mid-level features for recognition. In: CVPR. San Francisco, USA (2010)

[2] Comon, P.: Independent component analysis, a new concept? Signal Processing 36(3), 287–314 (1994)

[3] Deselaers, T., Deserno, T.: The visual concept detection task in imageclef 2008. In: ImageCLEF workshop (2008)

[4] Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: Liblinear: a library for large linear classification. Journal of Machine Learning Research 9, 1871–1874 (2008)

[5] van Gemert, J.C., Veenman, C.J., Smeulders, A.W.M., Geusebroek, J.M.: Visual word ambiguity. IEEE Transactions on Pattern Analysis and Machine Intelligence 32(7), 1271–1283 (2010)

[6] Hyvärinen, A., Karhunen, J., Oja, E.: Independent Component Analysis. Wiley-Interscience (May 2001)

[7] Jaakola, T., Haussler, D.: Exploiting generative models in discriminative classifiers. In: NIPS. pp. 1–8 (1999)

[8] Jégou, H., Douze, M., Schmid, C., Pérez, P.: Aggregating local descriptors into a compact image representation. In: CVPR. San Francisco, USA (june 2010)

[9] Kooperberg, C., Stone, C.J.: Logspline density estimation for censored data. Journal of Computational and Graphical Statistics 1, 301–328 (1997)

[10] Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR. pp. 2169–2178. Washington, DC, USA (2006)

[11] Le Borgne, H., Guérin Dugué, A., Antoniadis, A.: Representation of images for classification with independent features. Pattern Recognition Letters 25(2), 141–154 (2004)

[12] Le Borgne, H., Honnorat, N.: Fast shared boosting for large-scale concept detection. Multimedia Tools and Applications (2010)

[13] Liu, J., Shah, M.: Scene modeling using co-clustering. In: ICCV (2007)

[14] Lowe, D.G.: Object recognition from local scale-invariant features. In: CVPR 1999. vol. 2, pp. 1150–1157. Los Alamitos, CA, USA (August 1999)

[15] Masnadi-Shirazi, H., Mahadevan, V., Vasconcelos, N.: On the design of robust classifiers for computer vision. In: CVPR. pp. 779–786. San Francisco, USA (june 2010)

[16] Perronnin, F., Dance, C.R.: Fisher kernels on visual vocabularies for image categorization. In: CVPR (2007)

[17] Perronnin, F., Dance, C.R.: Large-scale image retrieval with compressed fisher kernels. In: CVPR. pp. 3384–3391. San Francisco, USA (2010)

[18] Rasiwasia, N., Vasconcelos, N.: Holistic context modeling using semantic co-occurrences. In: CVPR. vol. 0, pp. 1889–1895. Los Alamitos, CA, USA (2009)

[19] Rasiwasia, N., Vasconcelos, N.: Scene classification with low-dimensional semantic spaces and weak supervision. In: CVPR. pp. 1–6 (2008)

[20] Sivic, J., Zisserman, A.: Video google: A text retrieval approach to object matching in videos. In: ICCV. pp. 1470–1477 vol.2 (2003)

[21] Yang, J., Yu, K., Gong, Y., Huang, T.: Linear spatial pyramid matching using sparse coding for image classification. In: CVPR (2009)