Pooling in image representation: the visual codeword point of view[1]

S. Avila[a,b,*], N. Thome[a], M. Cord[a], E. Valle[c], A. de A. Araújo[b]

[a] *Université Pierre et Marie Curie, UPMC-Sorbonne Universities, LIP6, 4 place Jussieu, 75005, Paris, France*
[b] *Federal University of Minas Gerais, NPDI Lab – DCC/UFMG, Belo Horizonte, MG, Brazil*
[c] *State University of Campinas, RECOD Lab – DCA/FEEC/UNICAMP, Campinas, SP, Brazil*

Abstract

In this work, we propose BossaNova, a novel representation for content-based concept detection in images and videos, which enriches the Bag-of-Words model. Relying on the quantization of highly discriminant local descriptors by a codebook, and the aggregation of those quantized descriptors into a single pooled feature vector, the Bag-of-Words model has emerged as the most promising approach for concept detection on visual documents. BossaNova enhances that representation by keeping a histogram of distances between the descriptors found in the image and those in the codebook, preserving thus important information about the distribution of the local descriptors around each codeword. Contrarily to other approaches found in the literature, the non-parametric histogram representation is compact and simple to compute. BossaNova compares well with the state-of-the-art in several standard datasets: MIRFLICKR, ImageCLEF 2011, PASCAL VOC 2007 and 15-Scenes, even without using complex combinations of different local descriptors. It also complements well the cutting-edge Fisher Vector descriptors, showing even better results when employed in combination with them. BossaNova also shows good results in the challenging real-world application of pornography detection.

Keywords: Image classification, Image Representation, Pattern Recognition, Bag-of-Words, Visual dictionary, coding, pooling, SVM

1. Introduction

Visual information, in the form of digital images and videos, has become so omnipresent in computer databases and repositories, that it can no longer be considered a "second class citizen", eclipsed by textual information. In that scenario, image classification and visual concept detection are becoming critical tasks. In particular, the pursuit of automatic identification of complex semantical concepts represented in images has motivated researchers in areas as diverse as Information Retrieval, Computer Vision, Image Processing and Artificial Intelligence [1, 2, 3, 4]. Though the ultimate goal of reliable concept identification remains elusive, the last decade has witnessed two important breakthroughs in that direction: the development of very discriminant low-level local features, inspired on Computer Vision approaches; and the emergence of mid-level aggregate representations, based on the quantization of those features, in the so-called "Bag-of-Words" model [5, 6]. Those advances in feature extraction and representation have closely followed a previous turning point on statistical learning, represented by the maturity of kernel methods and support vector machines [7, 8].

Our aim is content-based concept detection in images and videos, with a novel representation that enriches the Bag-of-Words model. Bag-of-Words representations can be understood as the application of

[1] This text is the latest preprint before acceptation. The final version after editing by the publisher can be found at http://dx.doi.org/10.1016/j.cviu.2012.09.007 or in Computer Vision and Image Understanding, Volume 117, Issue 5, May 2013, Pages 453-465.

[*] Corresponding author.

two critical steps [9]: coding, which quantizes the image local features according to a codebook or dictionary; and spatial pooling, which summarizes the codes obtained into a single feature vector. Traditionally, the coding step simply associates the image local descriptors to the closest element in the codebook, and the spatial pooling takes the average of those codes over the entire image.

Several trends are discernible on the mid-level representations recently proposed: the preservation of global spatial information, leading to the almost universal association to the Spatial Pyramids scheme [10]; and the concern with the integrity of the low-level descriptor information, which culminates in representations inspired from signal reconstruction. As a consequence, we have observed the steady inflation of feature vector sizes.

In this work, we propose BossaNova, a mid-level representation based on a histogram of distances between the descriptors found in the image and those in the codebook. The fundamental change is an enhancement of the pooling in order to preserve a richer portrait of the information gathered during the coding: instead of compacting all information pertaining to a codeword into a single scalar, the proposed pooling scheme produces a distance distribution. In order to accomplish that goal, BossaNova departs from the parametric models commonly found in the literature (e.g., [11, 12, 13, 14, 15]), by employing histograms. That non-parametric approach allows us to conciliate the need to preserve low-level descriptor information and keeping the mid-level feature vector at a reasonable size. A preliminary version of the representation [16] has allowed us to gain several insights into the benefits of the non-parametric choice and to explore the compromises between the opposite goals of discrimination versus generalization, representativeness versus compactness. Since BossaNova embodies the accomplishment of that preliminary work, this paper presents several new aspects :

- An extensive theoretical analysis (presented in Sections 2 and 3) that gives a unified perspective of both BoW and BOSSA models;
- A novel coding scheme based on semi-soft codeword assignment, that avoids the instability inherent to the use of hard codeword assignment on high-dimensional spaces;
- A novel normalization scheme, that renders the representation more robust to the sparsity brought by large codebooks. A new weighting scheme to balance the importance of different parts of the representation is also presented;
- A novel extension with the complementary state-of-the-art mid-level representation based on Fisher Vectors.

The remainder of this text is organized as follows. In Section 2, we formalize the Bag-of-Words model for images, and give a summary survey of the most important work which lead to its development, concluding with a brief commentary on the current state of the art. In Section 3, we give a detailed description of our approach BossaNova, both in terms of theoretical background and implementation, including a unified theoretical framework for BoW and BOSSA. In Section 4, we present our empirical results, comparing of BossaNova performance with state-of-the-art methods in several dataset, validating its enhancements over the previously proposed Bossa representation, and studying its behavior as its key parameters change. In Section 5, we explore BossaNova in the real world application of pornography detection, which because of its high-level conceptual nature, involves large intra-class appearance variations. With Section 6, we conclude the paper.

2. Related Work

In this section, we survey the literature on image representations based on the Bag-of-Words model. As its name suggests, that model is inspired from textual Information Retrieval, which contributed important ideas throughout its evolution. Here, however, we restrict our scope to works on visual information. Instead of an exhaustive survey, we opt for a more formal development: our aim is to lay out the mathematical

cornerstones common to all BoW representations, exploring how those cornerstones have been established in early works, and how they are evolving in very recent works.

In order to get the mid-level feature vector, the standard processing pipeline follows three steps [9]: (i) low-level local descriptor extraction, (ii) coding, and (iii) pooling. Classification algorithms (like support vector machines) are then trained on the mid-level feature vectors obtained.

As far as we know, the NeTra toolbox [17] was the first work to follow that scheme, proposing dense grids of color points, and unsupervised learning to build the codebook, using the LBG algorithm. The Retin system [18] is based upon a similar scheme, using local Gabor feature vectors, and learning the codebook with Kohonen self-organized maps. The technique was definitively popularized with the intuitive "Video Google" formalism [5], which makes explicit the parallels between the BoW models for visual and textual documents, while employing SIFT local features, and building the BoW using a three-step pipeline.

Let us denote the "Bag-of-Features" (BoF), i.e., the unordered set of local descriptors extracted from an image, by $\mathcal{X} = \{\mathbf{x}_j\}$, $j \in \{1, \ldots, N\}$, where $\mathbf{x}_j \in \mathbb{R}^d$ is a local feature vector and N is the number of local features (either fixed grid points, either detected points of interest) in the image.

Many feature detectors have been proposed to get salient areas, affine regions and points of interest [19] on images. However, in contrast to the task of matching a specific target image or object, methods for category classification show better performance when using a uniform feature sampling over a dense grid on the image [20].

Let us suppose we have obtained (e.g., by unsupervised learning) a codebook, or visual dictionary $\mathcal{C} = \{\mathbf{c}_m\}$, $\mathbf{c}_m \in \mathbb{R}^d$, $m \in \{1, \ldots, M\}$, where M is the number of codewords, or visual words. $\mathcal{C}$ represents the matrix $d \times M$ of all codeword coordinates, one codeword per column. Note that the codewords are in the same space of the low-level local descriptors ($\mathbb{R}^d$). Note also that, unless otherwise noted, all our vectors are column vectors.

Obtaining the codebook is essential for the "Bag-of-Words" (BoW) model, since the representation will be based on the codewords. Currently, the vast majority of methods obtains the codebook using unsupervised learning over a sample of local descriptors from the training images, usually using k-means. However, a few supervised approaches have been proposed [9, 21].

The construction of the BoW representation can be decomposed into the sequential steps of coding and pooling [9]. The coding step projects the local descriptors onto the codebook elements; while the pooling step aggregates the projected codes into a vector. The global aim is gaining invariance to nuisance factors (positioning of the objects, changes in the background, small changes in appearance, etc.), while preserving the discriminating power of the local descriptors.

The coding step can be modeled by a function f: $\mathbb{R}^d \longrightarrow \mathbb{R}^M$ as $f(\mathbf{x}_j) = \alpha_j$ (see Figure 1). It can be understood as an activation function for the codebook, activating each of the codewords according to the local descriptor. In the classical BoW representation, the coding function activates only the codeword closest to the descriptor, assigning zero weight to all others :

$$\alpha_{m,j} = 1 \text{ iff } m = \underset{k \in \{1,\ldots,M\}}{\operatorname{argmin}} \|\mathbf{x}_j - \mathbf{c}_k\|_2^2$$

where $\alpha_{m,j}$ is the m^{th} component of the encoded vector α_j. That scheme corresponds to a hard coding or hard quantization over the dictionary. The resulting binary code is very sparse, but suffers from instabilities when the descriptor being coded is on the boundary of proximity of several codewords [6].

Because of that, alternatives to that standard scheme have been recently developed. Sparse coding [9, 22] modifies the optimization scheme by jointly considering reconstruction error and sparsity of the code, using the well-known property that regularization with the ℓ_1-norm, for a sufficiently large regularization parameter λ, induces sparsity:

$$\alpha_j = \underset{\alpha}{\operatorname{argmin}} \|\mathbf{x}_j - \mathcal{C}\|_2^2 + \lambda \|\alpha\|_1$$

One of the strengths of that approach is that one can learn the dictionary with the same scheme, but optimizing over $\mathcal{C}$ and α. Efficient tools have been proposed to get tractable solutions [21].

Another possibility is soft coding [6]. It is based on a soft assignment to each visual word, weighted

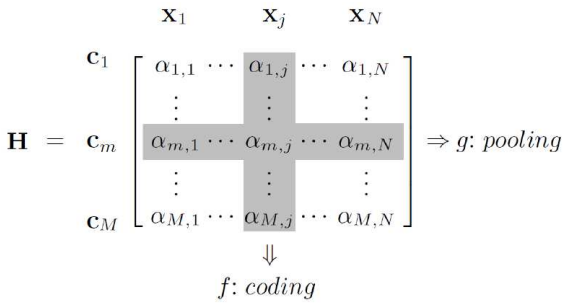


Figure 1: Matrix representation $\mathbf{H}$ of the BoW model, with columns $\mathcal{X}$ related to the low-level local descriptors, and rows $\mathcal{C}$ related to the codebook. The coding function f for a given descriptor $\mathbf{x}_j$ corresponds to column j, and may be interpreted as how much that descriptor activates each codeword. The pooling function g for a given visual word $\mathbf{c}_m$ corresponds to a summarization of row m and may be interpreted as the aggregation of the activations of that codeword. The final representation is a vector $\mathbf{z}$ (not shown), containing those aggregated activations, for each codeword.

by distances/similarities between descriptors and codewords. Soft assignment results in dense code vectors, which is undesirable, among other reasons, because it leads to ambiguities due to the superposition of the components in the pooling step. Therefore, several intermediate strategies – known as "semi-soft" coding – have been proposed, often applying the soft assignment only to the k nearest neighbors (k-NN) of the input descriptor [23].

The pooling step takes place after the coding, and can also be represented by a function, such as g: $\{\alpha_\mathbf{j}\}_{j\in 1,\ldots,N} \longrightarrow \mathbb{R}^M$ as: $g(\{\alpha_j\}) = \mathbf{z}$ which can be used to get a single scalar value on each row of the $\mathbf{H}$ matrix (see Figure 1). Traditional BoW considers the *sum pooling* operator:

$$g(\{\alpha_j\}) = \mathbf{z} : \ \forall m, \ z_m = \sum_{j=1}^{N} \alpha_{m,j} \tag{1}$$

When using sparse or soft coding, the *max pooling* is often preferred[2]:

$$\mathbf{z} : \ \forall m, \ z_m = \max_{j\in\{1,\ldots,N\}} \alpha_{m,j}$$

The vector $\mathbf{z} \in \mathbb{R}^M$ is the final image representation, used for classification. Extensions to the traditional pooling operation have been also proposed recently. The most powerful technique is the Spatial Pyramid Matching (SPM) strategy [10]. It is a pooling that considers a fixed predetermined spatial image pyramid. The previously described pooling is operated over each block of the pyramid, then concatenated into a large vector (#blocks $\times$ M).

Boureau et al. [24] stepped forward in considering both SPM and local pooling over the codes. That latter work also gives a new perspective to other recent powerful approaches VLAD [14] or Super-Vector Coding [15] as specific pooling operations. In those aggregated methods, locality constraints are incorporated during the pooling step: only descriptors belonging to the same clusters are pooled together.

Another BoW improvement belonging to the aggregated coding class is the Fisher Kernel approach proposed by Perronnin et al. [11]. It is based on the use of the Fisher kernel framework popularized by Jaakkola and Haussler [25], with Gaussian Mixture Models (GMM) estimated over the whole set of images. That approach may be viewed as a generalization to the second order of the Super-Vector approach [15]. Indeed, the final image representation is also a vector concatenating vectors over each mixture term. Picard

[2]Depending on the sparse optimization scheme, the $\alpha_{m,j}$ values may be negative. If that occurs, the following pooling is usually applied: $\mathbf{z} : \ \forall m, \ z_m = \max_{j\in\{1,\ldots,N\}} \|\alpha_{m,j}\|$.

and Gosselin [26] generalize it to higher orders, but computational complexity, vector size and difficulty in estimating higher-order moments with confidence, limit the practicality of pushing the orders beyond the second.

In a previous work [16], we had proposed another extension to pooling, called BOSSA, by considering no more a scalar output for each row as in Equation 1, but a vector, summarizing the distribution of the $\alpha_{m,j}$. That strategy allows keeping more information, related to the confidence of the detection of each visual word $\mathbf{c}_m$ in the image.

We propose in this work a new pooling method, called BossaNova, that generalizes our previous pooling strategy, with a new assignment and normalization strategy that makes the representation more effective, while keeping all advantages of BOSSA.

3. BossaNova Scheme

BossaNova is based upon a new pooling strategy, and integrates several improvements over the original BOSSA representation [16]. We open this section by reviewing the pooling formalism for the three representations: BoW, BOSSA and BossaNova, allowing us to contrast the differences between the latter two and the former. We then detail the improvements of BossaNova over BOSSA, including the weighting scheme to balance the word-count (BoW) and the distances-histogram (BOSSA) parts of the vectors, the semi-soft coding scheme and the improved normalization. The implementation details are then briefly discussed. Finally, we conclude this section with an analysis of how BossaNova and Fisher Vectors can be expected to complement each other well when combined into a single feature vector.

3.1. New Pooling Formalism

As hinted in the previous section, for representations based on the BoW model, the pooling step is critical. It compacts all the information contained in the individually encoded local descriptors into a single feature vector, thus producing a mid-level feature convenient for use with classifiers like SVM.

When pooling, there is a compromise between the invariance obtained and the ambiguities introduced. Invariance to different backgrounds or object positioning is obtained because the final codewords will be activated despite the precise positioning of the descriptors. However, since all activations are combined, ambiguities can arise, if different concepts represented in the image (e.g., a person and a car) end up activating sets of codewords that overlap too much. The following step of classification will have difficulty in separating those concepts.

One way to mitigate that problem is to preserve more information about the encoded descriptors during the pooling step. Instead of a simple sum of the activations, like in the classical BoW, more detailed information can be kept.

In BOSSA and BossaNova, we propose estimating the distribution of the descriptors around each codeword. We choose a non-parametric estimation of the descriptors distribution, by computing a histogram of distances between the descriptors found in the image and each codebook element.

More formally, and keeping the same notations used in Section 2 and in Figure 1, the proposed pooling function g estimates the probability density function of α_m: $g(\alpha_m) = \mathrm{pdf}(\alpha_\mathbf{m})$, by computing the following histogram of distances $z_{m,k}$:

$$
\begin{aligned}
g : \mathbb{R}^N &\longrightarrow \mathbb{R}^B \\
\alpha_\mathbf{m} &\longrightarrow g(\alpha_m) = z_m \\
z_{m,k} &= \mathrm{card}\left(\mathbf{x}_j \mid \alpha_{m,j} \in \left[\frac{k}{B}; \frac{k+1}{B}\right]\right) \\
&\quad \frac{k}{B} \geq \alpha_m^{min} \ \text{ and } \ \frac{k+1}{B} \leq \alpha_m^{max}
\end{aligned}
\tag{2}
$$

where B denotes the number of bins of each histogram z_m, and $[\alpha_m^{min}; \alpha_m^{max}]$ limits the range of distances for the descriptors considered in the histogram computation. On BOSSA, only the upper range was limited,

but we have since observed that, due to a known effect of the "curse of dimensionality", distances between descriptors seldom fall below a certain range, making some bins of BOSSA histograms always zero. The double range makes better use of the representation space.

The function g represents the discrete (over B bins) density distribution of the distances $\alpha_{m,j}$ between the codeword $\mathbf{c}_m$ and the local descriptors of an image. That is illustrated in Figure 2. Note that $\alpha_{m,j}$, introduced in Figure 1, traditionally quantifies a similarity between the descriptor $\mathbf{x}_j$ and the codeword $\mathbf{c}_m$, while in our pooling formalism, it represents a dissimilarity (indeed, a distance). That choice makes illustrations clearer and more intuitive, and no generality is lost, since estimating a similarity pdf for $\alpha_{m,j}$ from our model is straightforward.

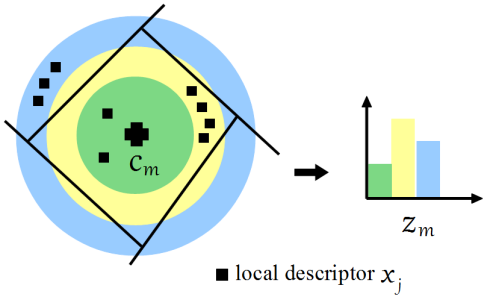


Figure 2: For each center $\mathbf{c}_m$, we obtain a local histogram z_m. The colors indicate the discretized distances from the center $\mathbf{c}_m$ to the local descriptors shown by the black dots. For each colored bin $z_{m,k}$, the height of the histogram is equal to the number of local descriptors $\mathbf{x}_j$, whose discretized distance to codeword $\mathbf{c}_m$ fall into the k^{th} bin. We can note that if $B = 1$, the histogram z_m reduces to a single scalar value counting the number of feature vectors $\mathbf{x}_j$ falling into center $\mathbf{c}_m$. Therefore, the proposed histogram representation can be considered as a consistent generalization of BoW pooling step.

3.2. BossaNova Improvements

The novel pooling strategy presented in previous section is the basis of both BOSSA and BossaNova. The latter, however, presents several improvements over the former, which we explore now. The effectiveness of each of those improvements is evaluated in Section 4.2.

3.2.1. Weighting BoW and BOSSA

The main result of the pooling step is a local histogram z_m for each codeword $\mathbf{c}_m$. We concatenate those histograms to form the feature vector. In addition, we propose incorporating an additional scalar N_m for each codeword, counting the number of local descriptors falling close to that codeword. That value corresponds to a classical BoW term, accounting for a raw measure of the presence of the visual word $\mathbf{c}_m$ in the image. Previously [16], we simply concatenated the BoW and BOSSA components, implicitly assigning equal importance to the components z_m and N_m.

We propose here to weight z_m and N_m, setting thus the relevance of each term in BossaNova. We apply a weight factor s to each N_m value, rewriting our image representation $\mathbf{z}$ as:

$$\mathbf{z} = \left[[z_{m,k}] \, , sN_m \right]^T , \quad (m,k) \in \{1;M\} \times \{1;B\} \tag{3}$$

As illustrated in Figure 3, $\mathbf{z}$ is a vector of size $D = M \times (B+1)$. The weighted factor s is learned via cross-validation on a training/validation sub-set.

Equation 3 lets us interpret BossaNova as an improvement over the BoW representation, through the use of an additional term coming from the a more informative pooling function. Recently, that idea of enriching BoW representations with extra knowledge from the set of local descriptors has been explored on several representations. It can be found, for example, on Fisher Vectors [12] and Super-Vector Coding [15]. Those works, however, opt by parametric models that lead to very high-dimensional image representations.

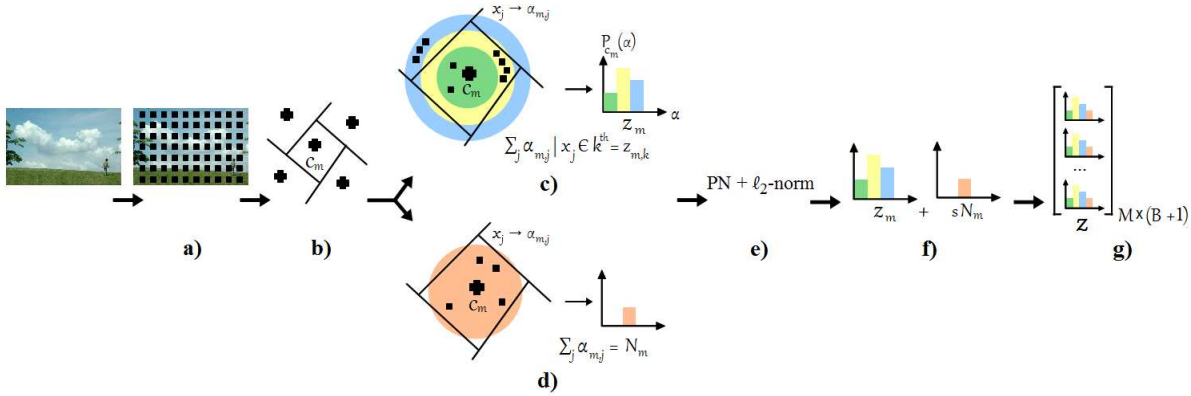


Figure 3: Overview of BossaNova vector construction. **(a)** Extraction of the low-level local descriptors (SIFT) over a dense grid. **(b)** Codebook / visual vocabulary creation with k-means on a sample of one million descriptors. **(c)** Our pooling strategy: computation of local histograms z_m for each $\mathbf{c}_m$ codeword. Localized soft-assignment ("semi-soft assignment") is used for coding. **(d)** Counting the number of feature vectors $\mathbf{x}_j$ falling into each codeword $\mathbf{c}_m$ (again, using semi-soft assignment). **(e)** Two-step normalization: power normalization followed by ℓ_2-normalization. **(f)** Weighting of the histogram (z_m) and counting components (N_m), by applying a weight factor s on the latter. **(g)** Final BossaNova representation .

By using a simple histogram of distances to capture the relevant information, our approach remains very flexible and keeps the representation compact. In the experiments (Section 4), we show that we can reach performances close to the ones of the Fisher Vectors, with a much smaller descriptor.

3.2.2. Localized Soft-Assignment Coding

Our previous work [16] employed hard assignment on the coding step, for both the BOSSA (histograms) and BoW (raw counts) components of the feature vector. In BossaNova, we propose a soft-assignment coding strategy, for both components. Soft-assignment is chosen because it has been shown to considerably enhance the results over hard assignment, without incurring the computational costs of sparse coding [9, 22]. In addition, a recent evaluation [23] reveals that well-designed soft coding can perform as well or even better than sparse coding.

Soft-assignment coding attenuates the effect of coding errors induced by the quantization of the descriptor space. Different soft coding strategies have been presented and evaluated by Gemert et al. [6], the most successful approach being the one they call "codeword uncertainty". Other authors [23, 24, 27] point out the importance of locality in the coding, an issue we will address in Section 3.4, and that leads us to a localized, "semi-soft" coding scheme.

Thus, like Liu et al. [23], we consider only the k-nearest visual words in coding a local descriptor, and we perform for those neighbors a "codeword uncertainty" soft assignment. Let us consider a given local descriptors $\mathbf{x}_j$, and its k closest visual words $\mathbf{c}_m$. The soft assignment $\alpha_{m,j}$ to the visual word $\mathbf{c}_m$ is computed as follows:

$$\alpha_{m,j} = \frac{exp^{-\beta_m d_2(\mathbf{x}_j, \mathbf{c}_m)}}{\sum_{m'=1}^{K} exp^{-\beta_{m'} d_2(\mathbf{x}_j, \mathbf{c}_{m'})}} \tag{4}$$

where $d_2(\mathbf{x}_j, \mathbf{c}_m)$ is the (Euclidean) distance between $\mathbf{c}_m$ and $\mathbf{x}_j$. The parameter β_m regulates the softness of the soft-assignment (the bigger it is, the hardest the assignment). The main difference between our approach and the one of Liu et al. [23] is that we allow β_m to vary for each codeword, while they use a global β parameter, determined by cross-validation. Since our codewords $\mathbf{c}_m$ correspond to cluster centers obtained by a k-means algorithm, we take advantage of the standard deviation σ_m of each cluster $\mathbf{c}_m$ to setup $\beta_m = \sigma_m^{-2}$.

3.2.3. Normalization

In BossaNova, the third improvement over BOSSA [16] is a two-step signature normalization.

The first step in that normalization is motivated by the following observation: as the number of visual words increases, BOSSA becomes sparser. That is also the case for most BoW-like representations: Perronnin et al. [12] have also observed that effect, which is indeed a direct consequence of the ratio between the number of local descriptors and the mid-level representation vector size. They observe that similarities become less reliable when the vector signatures become too sparse, proposing a power normalization to alleviate that drawback. Therefore, we choose to incorporate that normalization into the BossaNova representation.

Formally, the power normalization consists of applying the following operator in each histogram bin $z_{m,k}$:

$$h(z_{m,k}) = \text{sign}(z_{m,k})|z_{m,k}|^{\delta}, \qquad 0 < \delta \leq 1 \tag{5}$$

In our experiments, we consider $\delta = 0.5$, which has shown in preliminary experiments to provide better performance.

The second step is an ℓ_2-normalization applied to the final vector. In contrast, BOSSA did not implement the power normalization and employed an ℓ_1 block-norm strategy instead of the ℓ_2. Our experiments show that the change has improved the results.

3.3. Implementation Details

The key parameters in our representation are the number of visual words M, the number of bins B in each histogram z_m, the minimum distance α_m^{min} and the maximum distance α_m^{max} in the $\mathbb{R}^d$ descriptor space that define the bounds of the histogram (see Equation 2).

The codebook size M has a similar meaning as in standard BoW approaches. Histogram size B defines the granularity to which $\text{pdf}(\alpha_{\mathbf{m}})$ is estimated. The choices of M and B are co-dependent, and $M \cdot B$ determines the compromise between accuracy and robustness. The smaller $M \cdot B$ is, the less the representation is accurate, the larger $M \cdot B$ is, the less confidence we have on the estimate of each bin of the histogram representing the underlying distribution. In addition, too large $M \cdot B$ values may lead to excessively sparse vector representations. In our experiments, we use $M \sim 4000$ and B in the range $[2; 6]$.

The bounds α_m^{min} and α_m^{max} define the range of distances for the histogram computation. Local descriptors outside those bounds are ignored. For α_m^{max}, the idea is to consider only descriptors that are "close enough" to the center, and to discard the remaining ones. For α_m^{min}, the idea is to avoid the empty regions that appear around each codeword, in order to avoid wasting space in the final descriptor.

The fact that descriptors seldom, if ever, fall close to the codewords is a counter-intuitive consequence of the geometry of high-dimensional spaces. Figure 4 illustrates the phenomenon, displaying the average density of SIFT descriptors on the neighborhood of codewords, in MIRFLICKR dataset. It is clear that the number of SIFT descriptors for $\alpha_m^{min} < 0.4 \cdot \sigma_m$ is negligible (see Section 4.3.3).

Note that the parameters may act jointly to the locality constraints defined in Section 3.2.2: a descriptor $\mathbf{x}_j$ that is the k-NN from a center $\mathbf{c}_m$ is not considered for generating the signature if $d_2(\mathbf{x}_j, \mathbf{c}_m) > \alpha_m^{max}$.

In BossaNova, α_m^{min} and α_m^{max} are set up differently for each codeword $\mathbf{c}_m$. Since our visual dictionary is created using k-means, we take advantage of the knowledge about the "size" of the clusters, given by the standard deviations σ_m. We set up the bounds as $\alpha_m^{min} = \lambda_{min} \cdot \sigma_m$ and $\alpha_m^{max} = \lambda_{max} \cdot \sigma_m$, as shown in Figure 5. In practice, the three parameters of the BossaNova become B (M being fixed), λ_{min} and λ_{max}.

3.4. BossaNova & Aggregated Methods: Complementary

Although alternative pooling strategies have recently been explored (e.g. max pooling), average pooling remains the most commonly employed scheme for aggregating local descriptors. As pointed out by Boureau et al. [24], incorporating locality constraints during coding or pooling is mandatory for extracting a meaningful image representation when using average pooling. That is especially the case for state-of-the-art local descriptors such as SIFT or HOG that cannot be averaged without considerably loosing information. For example, if we do not consider any coding step (i.e. $M = d$, $f = I_d$ in Figure 1), aggregating SIFT or HOG descriptors with average pooling would produce a global histogram of gradient orientation for the image. Thus, if care is not taken, the pooling step makes the representation uninformative for classification.

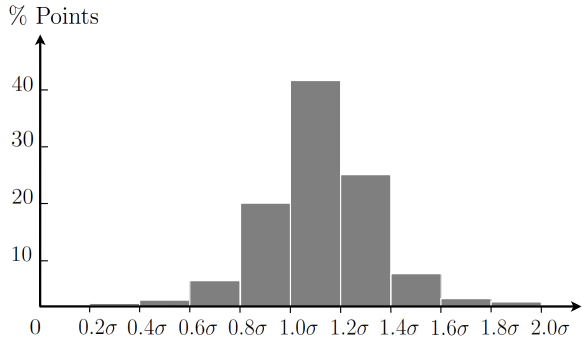


Figure 4: Average density of SIFT descriptors in the neighborhood of codewords in MIRFLICKR dataset, showing that descriptors seldom, if ever, are closer than a certain threshold to the codewords. That counter-intuitive phenomenon is a consequence of the "curse of dimensionality".

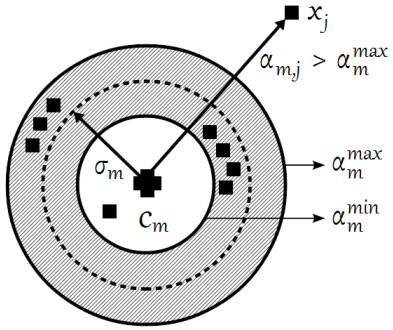
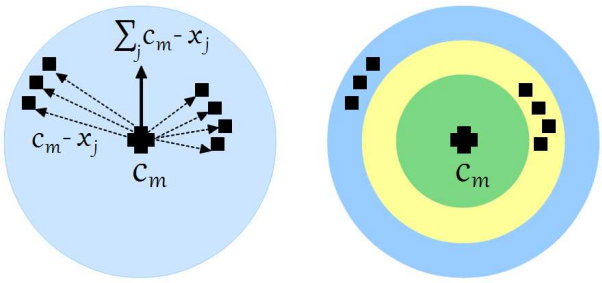


Figure 5: Illustration of the range of distances $[\alpha_m^{min}, \alpha_m^{max}]$ which defines the bounds of the histogram. The hatched area corresponds to the bounds. Local descriptors outside those bounds are ignored.

In aggregated methods such as Fisher Vectors [11], VLAD [14] or Super-Vector Coding [15], the locality constraints are mainly incorporated during the pooling step. In that class of methods, since the coding step is much more accurate (for each codeword, a vector is stored instead of a simple scalar value with standard BoW coding schemes), the authors often claim that they can afford to use a codebook of limited size (e.g. $M \sim 100$) and get very good performances. However, reducing the codebook size intrinsically increases the hypervolume of each codeword in the descriptor space. That naturally decreases the range of the locality constraints that can be incorporated during pooling: all local descriptors falling into a (now larger) codeword are averaged together. Therefore, we argue that average pooling used in aggregate methods may lack locality, as soon as the distribution of local descriptors becomes multi-modal inside a codeword. For example, Fisher Vectors model the distribution of local descriptors in each codeword with a single Gaussian. When that Gaussian assumption does not hold, the pooled representation may be unrepresentative of the local descriptor statistics. That is illustrated in Figure 6. Figure 6a shows an illustration of a cluster around codeword $\mathbf{c}_m$ with local descriptors $\mathbf{x}_j$ having two different modes (i.e. sub-clusters). When averaging the codes during pooling, we get for $\mathbf{c}_m$ a pooled vector $\sum_j \mathbf{c}_m - \mathbf{x}_j$ that is far away from any local descriptors $\mathbf{x}_j$. In contrast to that, BossaNova representation uses additional locality constraints during the pooling, since only the feature vectors $\mathbf{x}_j$ that are close to the codewords $\mathbf{c}_m$ are pooled together, as shown in Figure 6b. The pooled representation is thus able to capture the statistics of the local descriptors.

On the other hand, when the Gaussian assumption is fulfilled, aggregated methods provide powerful signatures thanks to the improved accuracy of the coding step. The two mid-level representations are thus complementary, and we can expect improving performances by combining them. In a supervised learning



a) Aggregate methods b) BossaNova

Figure 6: Aggregated methods, e.g. Fisher Vectors [12], may lack locality during pooling for small codebooks, whereas BossaNova does not. In counterpart, aggregated methods are more accurate during the coding steps, making the two representation complementary. See discussion in Section 3.4.

task, the classifier is supposed to select the most relevant pooling strategy for each cluster, in a discriminative manner. As shown in the experiments (Section 4), we report that combining BossaNova with Fisher Vectors indeed improves classification performances.

4. Experimental Results

We choose four standard datasets to perform our experiments: MIRFLICKR [28], ImageCLEF 2011 [29], PASCAL VOC 2007 [30] and 15-Scenes [10]. Each dataset is briefly described at the moment of its first use, in Section 4.1.

After describing our experimental setup, we show our results, which we organized in three groups. First, a comparison with state-of-the-art methods, including both experiments with methods we have reimplemented ourselves, and published results reported in the literature. In order to make that comparison possible, we follow carefully the experimental protocol of each dataset. In what concerns the methods we reimplemented, we compare BossaNova to our previous work, BOSSA [16], but also to one of the best methods currently available, the Fisher Vectors [12]. In order to provide a control baseline, we also employ the classical Bag-of-Words (BoW).

Next, we evaluate the impact of the three proposed improvements of BossaNova over BOSSA, analyzing the isolated and joint impact of each enhancement on the new representation.

Finally, we explore the key aspects of the parametric space of our technique.

Experimental Setup

The low-level feature extraction has a big influence in the quality of the results, and, if not controlled can easily become a nuisance factor in the experiments. Therefore, to make the comparisons fair, we use the same low-level descriptors for all techniques evaluated. For all datasets, we have extracted SIFT descriptors [31] on a dense spatial grid, with the step-size corresponding to half of the patch-size, over 8 scales separated by a factor of 1.2, and the smallest patch-size set to 16 pixels.

As a result, roughly $8,000$ local descriptors are extracted from each image of MIRFLICKR, ImageCLEF 2011 and PASCAL VOC 2007 datasets, and close to $2,000$ local descriptors from each image of 15-Scenes. The dimensionality of the SIFT is reduced from 128 to 64 by using Principal Component Analysis (PCA). That setup for local descriptor extraction proves to give very good performances in standard image datasets, as reported in [20].

To learn the codebooks, we apply the k-means clustering algorithm with Euclidean distance over one million randomly sampled descriptors. For Fisher Vectors (FV) [12], the descriptor distribution is modeled using a GMM, whose parameters (w, μ, Σ) are also trained over one million randomly sampled descriptors,

using an EM algorithm. For all mid-level representations, we incorporate spatial information using the standard spatial pyramidal matching scheme [10]. In total, 8 spatial cells are extracted for MIRFLICKR, ImageCLEF 2011 and PASCAL VOC 2007, 21 spatial cells for 15-Scenes.

One-versus-all classification is performed by SVM classifiers. We use a linear SVM for FV, since it is well known that non-linear kernels do not improve performances for those representations, see [12]. For Bag-of-Words [5], BOSSA [16] and BossaNova, we use a non-linear Gauss-ℓ_2 kernel. Kernel matrices are computed as $exp(-\gamma d(x, x'))$ with d being the distance and γ being set to the inverse of the pairwise mean distances.

Significance tests for the differences between the means were performed using a t-test, paired over the dataset classes. For the analysis of the improvements brought by each enhancement of BossaNova over BOSSA, we have also employed a factorial ANOVA.

4.1. Comparison of State-of-the-Art Methods

We compare BossaNova to other representations, perform our own re-implementation of those techniques. The methods chosen were:

- BossaNova (BN), the method proposed in this paper.
- BOSSA [16], our previous work, which BossaNova improves, chosen to empirically validate those improvements.
- Fisher Vectors (FV) [12], one of the best mid-level representations currently reported in the literature [20].
- The combination BN + FV, chosen to evaluate the methods' complementarity (Section 3.4).
- Bag-of-Words (BoW) [5]. A classical histogram of visual words, obtained with hard quantization coding and average pooling; it constitutes a control baseline for the other methods.

When available, we also report the best results available for each dataset. That allows us to evaluate other recent methods that build upon the standard baseline BoW, e.g. recent methods using sparse coding and max pooling Yang et al. [22], Boureau et al. [9].

It is important to note that, although we have chosen for BN parameters we believed were good, in the interest of a fair comparison, we have not fine-tuned it for each dataset. Therefore, the numbers reported do not represent the limit of the performance achievable by the method (in a few cases higher results are achieved in this same paper in Section 4.3, were we do explore the parameters more thoroughly). It is also important to consider that most results reported in the literature employ different low-level descriptor extraction schemes, and that this step has a large impact on the results.

Results for MIRFLICKR

The MIRFLICKR dataset [28] contains 25,000 images collected from the Flickr photo sharing social network[3]. The dataset provides metadata, in the form of associated labels and tags, but we consider only the visual content for the feature extraction. The dataset is split into a collection of 15,000 training images and 10,000 test images, as defined by the standardized challenge "Visual Concept/Topic Recognition" [28]. All images are manually annotated for 38 concepts, including categories that describe the presence of specific object (*car*, *bird*, *dog*), categories that are concrete but less spatially localized (*clouds*, *night*, *sky*), and more abstract categories (*indoor*, *food*, *structures*, *transport*). The classification performance is evaluated using the standard metric for this dataset, the Mean Average Precision (MAP).

Table 1 shows the results over MIRFLICKR, and details the parameter settings for each method. Among the methods we have tested ourselves, all differences are significant with at least 99% confidence, except for

[3] http://www.flickr.com

BN and FV, whose difference is not significant. Results published in the literature, unfortunately, do not include significance tests or confidence intervals.

We can notice that all the recent methods improve the classification performance over the BoW baseline: the BOSSA representation published in [16] outperforms BoW with 1.2% absolute improvement (2.3% relative improvement). That illustrates the relevance of improving the pooling scheme, as we do in this paper.

Table 1: Image classification MAP (%) results of BossaNova (BN), standard implemented state-of-the-art representations and published methods on MIRFLICKR [28]. (1) BoW: $M = 4096$; (2) BOSSA: $M = 2048$, $B = 6$, $\lambda_{min} = 0$, $\lambda_{max} = 2$, as in [16], (3) FV: GMM with 256 Gaussians, as in [12]; (4) BN: $M = 4096$, $B = 2$, $\lambda_{min} = 0$, $\lambda_{max} = 2$, $s = 10^{-3}$.

	MAP (%)
Implemented methods	
BoW [5]	51.5
BOSSA [16]	52.7
FV [12]	54.3
BN (ours)	54.4
BN + FV (ours)	**56.0**
Published results	
Huiskes et al. [32]	37.5
Guillaumin et al. [33]	53.0

If we now compare the BOSSA to the proposed BossaNova, we observe an increase from 52.7% to 54.4%. That shows the benefits brought out by the weight factor, soft coding and new normalization proposed in Section 3.2 (further explored in Section 4.2). Furthermore, BossaNova is tied with Fisher Vectors, the current state-of-the-art method. Note that our representation (12,288 dimensions for each spatial cell) is about 3 times smaller than FV (32,768 dimensions for each spatial cell). Also, we observe that our method is better than Fisher Vectors for 22 out of 38 concepts[4].

Finally, we can notice the considerable improvement obtained when combining BossaNova and FV, reaching a MAP of 56.0%. This corresponds to a remarkable success of the complementariness of BossaNova and Fisher Vector representations. The combination surpasses both individual methods for 31 out of 38 concepts while performing similarly for the seven remaining concepts.

From the literature, we choose the baseline dataset result [32], and the best, as far as we know, result published [33]. The baseline performances [32] are quite low, 14% below our re-implementation of the classical BoW (Table 1). The main reason is the features employed there, global image descriptors, which are much outperformed by highly discriminant local descriptors such as SIFT.

In comparison to Guillaumin et al. [33], BossaNova performs better for 29 out of 38 concepts, and its MAP increases from 53.0% to 56.0%. It is notable BossaNova employs only SIFT to build the mid-level representation, while Guillaumin et al. [33] combines 15 different image representations, including SIFT.

To the best of our knowledge, ours is the best result reported to date on MIRFLICKR dataset, using a single low-level feature.

Results for ImageCLEF 2011

The ImageCLEF 2011[5] contains four main tasks: Medical Retrieval, Photo Annotation, Plant identification and Wikipedia Retrieval. We present our results for the ImageCLEF 2011 Photo Annotation Task [29], which consists of 18,000 Flickr images. The training set of 8,000 images includes annotations, EXIF data, and Flickr user tags, but we consider only the visual content for the feature extraction. The annotation challenge is performed on 10,000 images. The image set is annotated with 99 concepts that describe the

[4]The detailed per-class performances for all datasets are available at https://sites.google.com/site/bossanovasite/.
[5]http://www.imageclef.org/2011

scene (*indoor*, *outdoor*, *landscape*), depicted objects (*car*, *animal*, *person*), the representation of image content (*portrait*, *graffiti*, *art*), events (*travel*, *work*) or quality issues (*overexposed*, *underexposed*, *blurry*). The metric employed is the MAP.

Table 2 gives the results, both the ones implemented and tested by us, and the ones reported on literature. With at least 99% confidence, all differences were significant for the methods we have implemented ourselves. Once again, we note a considerable improvement of performance from BOSSA to BossaNova, from 32.9% to 35.3%. Furthermore, the combination of BossaNova and Fisher Vector representations outperforms the other methods.

Table 2: Image classification MAP (%) results of BossaNova (BN), standard implemented state-of-the-art representations and published methods on ImageCLEF 2011 Photo Annotation Task [29]. (1) BoW: $M = 4096$; (2) BOSSA: $M = 4096$, $B = 2$, $\lambda_{min} = 0$, $\lambda_{max} = 2$; (3) FV: GMM with 256 Gaussians, as in [12]; (4) BN: $M = 4096$, $B = 2$, $\lambda_{min} = 0.4$, $\lambda_{max} = 2$, $s = 10^{-3}$.

	MAP (%)
Implemented methods	
BoW [5]	31.2
BOSSA [16]	32.9
FV [12]	36.8
BN (ours)	35.3
BN + FV (ours)	38.4
Published results	
Mbanya et al. [34]	33.5
Le and Satoh [35]	33.7
de Sande and Snoek [36]	36.7
Su and Jurie [37]	38.2
Binder et al. [38]	38.8

We also compare our results with those of the five best systems reported in the literature. In the ImageCLEF 2011 Photo Annotation Task, each group registered for the challenge is restricted to a maximum of 5 runs. Table 2 shows the best run for each group, with the restriction to results that employed only the visual information.

The best system during the competition (Binder et al. [38]) reported 38.8% MAP, employing non-sparse multiple kernel learning and multi-task learning. They apply SIFT and color channel combinations to build different extensions of the BoW models with respect to sampling strategies and BoW mappings. The system of Su and Jurie [37] uses many features, such as SIFT, HOG, Texton, Lab–1948, SSIM, and Canny, aggregating them by a BoW into a global histogram. Fisher Vectors and contextual information were used as enhancement of the BoW models. The method of de Sande and Snoek [36] employs several color SIFT features with Harris-Laplace and dense sampling, and apply the SVM classifier. The system of Le and Satoh [35] also use numerous features. As global features, they use color moments, color histogram, edge orientation histogram and local binary patterns; and as local features, keypoint detectors such as Harris Laplace, Hessian Laplace, Harris Affine, and dense sampling are used to extract SIFT descriptors. Again, classification is performed with a SVM classifier. The approach of Mbanya et al. [34] is based on the BoW model. They apply feature fusion of the opponent SIFT descriptor and the GIST descriptor. Moreover, a post-classification processing step is incorporated in order to refine classification results based on rules of inference and exclusion between concepts. As we can notice, all those top-performing systems employ complex combinations of several low-level features to achieve their good results.

In view of that, our results of 35.3% for BN, and 38.4% for BN + FV, are remarkably good, since we employ just SIFT descriptors.

Results for PASCAL VOC 2007

The PASCAL VOC 2007 dataset [30] consists of $9,963$ images, from 20 object categories. Those images are split into three subsets: training ($2,501$ images), validation ($2,510$ images) and test ($4,952$ images). The following experimental results are obtained on trainval/test sets. To tune the C-SVM parameter, we use the train/val sets. Classification performance is measured by MAP across all classes, a metric chosen to facilitate the comparison with the literature.

Table 3 shows the results, detailing the parameter settings for each method. For the methods we (re-)implemented, all differences were significant with at least 99% confidence. Again, we observe a considerable improvement of performance from BOSSA to BossaNova, from 54.4% to 58.5%. The combination BN + FV still outperforms all other methods. For some categories its absolute improvement in MAP reached up to 10% (up to 37.7% of relative improvement), especially well-known challenging ones (e.g. *bottle*, *cow*, *sheep*).

Table 3: Image classification MAP (%) results of BossaNova (BN), standard implemented state-of-the-art representations and published methods on PASCAL VOC 2007 dataset [30]. (1) BoW: $M = 4096$; (2) BOSSA: $M = 4096$, $B = 2$, $\lambda_{min} = 0$, $\lambda_{max} = 2$; (3) FV: GMM with 256 Gaussians, as in [12]; (4) BN: $M = 4096$, $B = 2$, $\lambda_{min} = 0.4$, $\lambda_{max} = 2$ $s = 10^{-3}$.

	MAP (%)
Implemented methods	
BoW [5]	53.2
BOSSA [16]	54.4
FV [12]	59.5
BN (ours)	58.5
BN + FV (ours)	61.6
Published results	
Krapac et al. [13]	56.7
Wang et al. [27]	59.3
Chatfield et al. [20]	61.7

Table 3 also shows the comparison with published results. The comparison with Krapac et al. [13] is particularly relevant, because we employ the same low-level descriptor extraction as them, although our representation ends up being more compact. The LLC method of Wang et al. [27] is evaluated with HOG descriptors. LLC was also evaluated on extremely dense SIFT descriptors (sampling step of 3 pixels, compared to 16 used in our experiments), roughly $70,000$ per image, obtaining a MAP of 53.8% with a codebook of $4,000$ words [20].

The best reproducible results currently known are 58.2% for Super-Vector (SV) coding[6] and 61.7% for FV Chatfield et al. [20]. Those results are encouraging, since the SIFT descriptors employed on those experiments are extremely dense. As observed by Chatfield et al. themselves, denser sampling yield higher classification accuracies for all techniques, a result which we have also observed in preliminary tests on BOSSA and BossaNova.

Again, the combined BN + FV results show the complementarity of those methods. The performance is practically tied with the best reproducible results reported in the literature, but using SIFT features nearly $10\times$ less dense.

Results for 15-Scenes

The 15-Scenes dataset [10] contains 4,485 images of 15 natural scene categories. Following the standard experimental setup, we randomly select 100 images per class for training and the remaining images for testing. We average the classification accuracy over 30 random train/test splits.

[6]Even if Zhou et al. [15] published on this dataset a score of 64.0% using SV coding, Chatfield et al. [20] show that the SV coding is about 58.2%, and the difference results from nontrivial optimizations not described in their paper, making it extremely hard to reproduce.

Results, both the ones implemented and tested by us, and the ones reported on the literature are shown in Table 4. Once again, we observe that BossaNova method surpasses BOSSA with a absolute improvement of 2.4% (relative improvement of 2.9%), validating the improvements of the method. In comparison to Fisher Vectors, BossaNova classification performance is peculiarly inferior: this is the dataset showing the largest difference. We must note for one single class (*industrial*) our result is much lower than expected, weighting down the averages. When combining BossaNova and Fisher Vector methods, that issue is solved, and the combination is better than FV in isolation. The combination BN + FV surpasses both individual methods for 13 out of 15 natural scene categories.

Table 4: Image classification accuracy (%) results of BossaNova (BN), standard implemented state-of-the-art representations and published methods on 15-Scenes dataset [10]. (1) BoW: $M = 4096$; (2) BOSSA: $M = 4096$, $B = 2$, $\lambda_{min} = 0$, $\lambda_{max} = 2$; (3) FV: GMM with 256 Gaussians, as in [12]; (4) BN: $M = 4096$, $B = 2$, $\lambda_{min} = 0$, $\lambda_{max} = 2$, $s = 10^{-3}$. The table shows the means and standard deviations of 30 accuracy measures.

	Accuracy (%)
Implemented methods	
BoW [5]	81.1 ± 0.6
BOSSA [16]	82.9 ± 0.5
FV [12]	88.1 ± 0.2
BN (ours)	85.3 ± 0.4
BN + FV (ours)	88.9 ± 0.3
Published results	
Yang et al. [22]	80.3 ± 0.9
Lazebnik et al. [10]	81.4 ± 0.5
Boureau et al. [9]	85.6 ± 0.2
Krapac et al. [13]	88.2 ± 0.6

We also compare our results with those of the best systems reported in the literature. BossaNova outperforms considerably the methods reported by Yang et al. [22] and Lazebnik et al. [10], using improved BoWs with sparse coding and max pooling.

If we take our best result (88.9%), we observe that it is better than the result of Krapac et al. [13], obtained with spatial Fisher Vectors. Again, that comparison is relevant since both Krapac et al. and we employ similar low-level local descriptor extractions.

Figure 7 illustrates the confusion matrix for our best classification performance. Not surprisingly, confusion occurs between indoor classes (e.g. *bedroom*, *living room*, *kitchen*), urban architecture classes (*e.g. inside city*, *street*, *tall building*) and also between natural classes (e.g. *coast*, *open country*). Our result reaches near state-of-the-art performance for that dataset.

4.2. BOSSA to BossaNova Improvements Analysis

In Section 4.1, we show that BossaNova shows good results when compared to state-of-the-art works, and, in particular, that it considerably outperforms BOSSA [16]. To further quantify the performance gains, we propose in this section to evaluate the individual performance increase brought out by each of the three proposed improvements: learning the weighting between BoW and BOSSA (Section 3.2.1), using a localized-soft coding strategy (Section 3.2.2), and applying a new normalization to the final vector representation (Section 3.2.3).

The joint activation of the three steps leads to eight different configurations where the performance of the corresponding mid-level representation is evaluated (denoted as **Weight**, **Soft** and **Norm** in Tables 6 and 5). Then, we apply a statistical t-test to attest the significance of the difference between two given configurations. We perform the test for paired samples, i.e., we evaluate the performance of two configurations on N different folds of train/test images and compute the difference between the performance metrics on each fold. The confidence interval (CI) for the average difference is computed using a Student-t model [39], and the difference

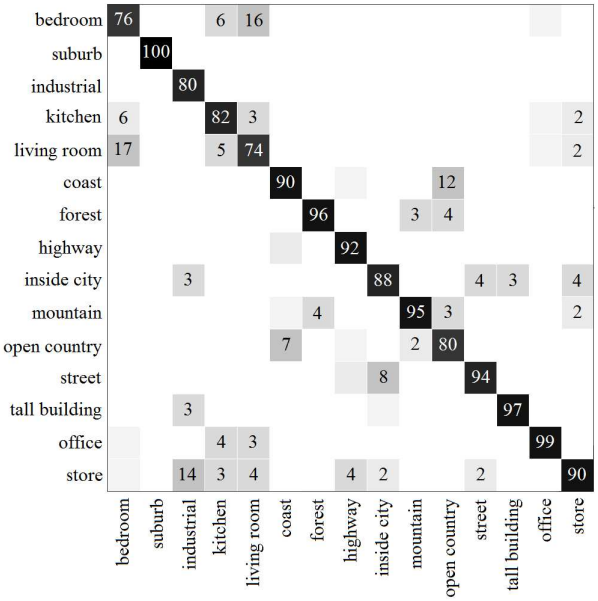


Figure 7: Confusion matrix for the 15-Scenes dataset. The average classification rates for individual classes are listed along the diagonal, and the columns are the true classes.

is considered significative if the interval does not include zero. For the tests in this section, we ask for a confidence of 95%.

Table 6 shows the evaluation of the eight different configurations on the 15-Scenes database, for $N = 30$ folds. We can see that the performances, measured by accuracy, monotonically increase from configuration **1** (BOSSA) to **8** (BossaNova). When only one improvement is added to BOSSA (configurations **2**, **3** and **4**), the performance gain is always significant. That already proves the relevance of the three modifications proposed in this paper. When two improvements are incorporated, the performances increase are significant when compared to BOSSA (**1**), but also when compared to configurations with only one improvement: configurations **5**, **6** and **7** are all significantly better than the best configuration with one improvement (**4**). When all three improvements are added, the difference is again significant: **8** is better than **6** and **7**, the best configurations including two improvements.

Testing just for the difference between BOSSA (**1**) and BossaNova (**8**) allows us to set the confidence to the large value of 99.9% and still obtain a CI of [1.18, 3.72], showing that the difference is significant.

We apply the same setup on the PASCAL VOC database. Here, the performance metric is the MAP, computed over the 20 classes for $N = 10$ folds[7]. The same conclusions apply: each improved configuration significantly outperforms its predecessor, as illustrated in Table 5.

Again, the difference between BOSSA (**1**) and BossaNova (**8**) is significant with a large confidence. For 99.9% confidence, the CI is $[3.21, 4.65]$.

For both datasets we have also tested the influence of the parameters using a factorial analysis of variance (ANOVA) [39]. In both cases, the models obtained were highly significant (with confidence above 99.9%) for all three improvements, confirming the results above. In addition, the ANOVA allows to measure the relative impact of each influence. For the more challenging VOC dataset, the soft assignment coding explains

[7]Note that in the VOC 2007 database, the train/val/test folds are generally fixed for evaluating performances. Here, we use random folds to obtain the necessary number of runs for statistical analysis.

Table 5: Impact of the proposed improvements to the BossaNova on PASCAL VOC 2007. We use $M = 4096$, $B = 2$, $\lambda_{min} = 0$, $\lambda_{max} = 2$. "Weight": the weighted factor s, No = no cross-validation, Yes = cross-validation. "Soft": soft assignment coding, No = hard-assignment, Yes = localized soft assignment. "Norm": normalization, No = ℓ_1 block normalization, Yes = power normalization + ℓ_2-normalization.

	Weight	**Soft**	**Norm**	**MAP***	**Confidence Interval (95%)****
1	No	No	No	54.9 ± 0.5	
2	Yes	No	No	55.2 ± 0.4	$2-1 = [0.25, 0.39]$
3	No	Yes	No	55.8 ± 0.5	$3-1 = [0.76, 1.12]$
4	No	No	Yes	55.6 ± 0.4	$4-1 = [0.94, 1.16]$
5	Yes	No	Yes	55.9 ± 0.4	$5-1 = [0.57, 0.85]$, $5-4 = [0.28, 0.40]$
6	Yes	Yes	No	56.4 ± 0.4	$6-1 = [1.34, 1.72]$, $6-4 = [0.62, 1.02]$
7	No	Yes	Yes	58.1 ± 0.4	$7-1 = [3.02, 3.38]$, $7-4 = [2.35, 2.63]$
8	Yes	Yes	Yes	58.8 ± 0.4	$8-1 = [3.59, 4.27]$, $8-7 = [0.45, 0.98]$

* Means and standard deviations of 10 MAP measures.
** Confidence intervals for the MAP differences. The difference is significant if its confidence interval does not contain zero (see text).

Table 6: Impact of the proposed improvements to the BossaNova on 15-Scenes. We use $M = 4096$, $B = 2$, $\lambda_{min} = 0$, $\lambda_{max} = 2$. "Weight": the weighted factor s, No = no cross-validation, Yes = cross-validation. "Soft": soft assignment coding, No = hard-assignment, Yes = localized soft assignment. "Norm": normalization, No = ℓ_1 block normalization, Yes = power normalization + ℓ_2-normalization.

	Weight	**Soft**	**Norm**	**Accuracy***	**Confidence Interval (95%)****
1	No	No	No	82.9 ± 0.5	
2	Yes	No	No	83.2 ± 0.2	$2-1 = [0.10, 0.50]$
3	No	Yes	No	83.4 ± 0.5	$3-1 = [0.24, 0.76]$
4	No	No	Yes	83.6 ± 0.1	$4-1 = [0.51, 0.89]$
5	Yes	No	Yes	83.9 ± 0.1	$5-1 = [0.80, 1.20]$, $5-4 = [0.15, 0.45]$
6	Yes	Yes	No	84.5 ± 0.4	$6-1 = [1.30, 1.90]$, $6-4 = [0.60, 1.20]$
7	No	Yes	Yes	84.5 ± 0.4	$7-1 = [1.37, 1.83]$, $7-4 = [0.42, 1.37]$
8	Yes	Yes	Yes	85.3 ± 0.4	$8-1 = [2.17, 2.63]$, $8-7 = [0.20, 1.40]$

* Means and standard deviations of 30 accuracy measures.
** Confidence intervals for the accuracy differences. The difference is significant if its confidence interval does not contain zero (see text).

almost 48% of the improvements, while the two-step normalization explains about 31%. The BoW–BOSSA weighting, in isolation, is responsible for only 3% of the variation, but there is a cross-effect between the weighting and the soft coding that accounts for another 9%. The impact of the coding is clearly the largest, but the importance of the normalization is quite surprising, especially considering the optimization of that step is often neglected in the literature.

4.3. BossaNova Parameter Evaluation

4.3.1. Codebook Size

The impact of codebook size M on BossaNova classification performance is shown on Table 7, which clearly shows that larger codebooks lead to higher accuracy. BoW performance, however, stops growing at 4096 visual words.

As stated in Section 4.1, the performances reported in Table 1 correspond to a BossaNova with good parameters, but not strongly fine-tuned. Therefore, our representation can reach an even higher score of 55.2% with a dictionary of size $M = 8192$. However, the last improvement from 4096 to 8192 is not that high, suggesting that the growth will soon stop growing. Meanwhile, the representation has doubled in size.

Hence, we define as our standard setting $M = 4096$ in order to get a good tradeoff between effectiveness and efficiency.

Table 7: Codebook size impact on BossaNova (BN) and BoW performance (MAP (%)) on MIRFLICKR dataset [28]. (1) BN: $B = 2$, $\lambda_{min} = 0$, $\lambda_{max} = 2$, $k\text{-}NN = 10$, $s = 10^{-3}$; (2) BoW: $M = 4096$.

	Codebook size			
	1024	2048	4096	8192
BN	51.8	52.9	54.4	55.2
BoW	50.3	51.3	51.5	51.1

Comparison with Hierarchical BoW

We contrast BossaNova to a Hierarchical BoW (H-BoW) since there are some similarities between our pooling approach and a 2-step descriptor space clustering. The pooling performed in BossaNova can indeed be regarded as a special form of clustering, where the second-level of clustering corresponds to regions that are equally spaced from the center. On the other hand, in a standard H-BoW, the second-level clusters are similar to the first-level ones (e.g. hyper-sphere, if ℓ_2 norm is used for clustering).

We claim that the special shape of the second-level clustering, which is based on the idea of pooling descriptors depending on their similarity to the center, is better founded that a naive 2-level clustering (with Euclidean distance). To achieve that comparison, we build a 2-level hierarchical codebook using BossaNova codebook size (M) at the first-level, and BossaNova histograms bin count plus one ($B + 1$) at the second-level. That makes the comparison fair, allocating the same size for both representations. For instance, BossaNova with a codebook of size $M = 4096$ and two bins per histogram ($B = 2$), will be compared with a H-BoW first-level of 4096 and second-level of 3 clusters (both representation are therefore of size $4096 \times 3 \times 8$, 8 being the spatial cells of the SPM scheme).

Table 8 compares BossaNova with H-BoW on the MIRFLICKR dataset. For each codebook size, we observe that BossaNova is superior to H-BoW, and that the difference tends to grow as the (first-level) codebook size grows. That confirms the relevance of the improved pooling scheme introduced in the paper.

Table 8: Comparison of BossaNova (BN) wrt Hierarchical BoW performance (MAP (%)) on MIRFLICKR dataset [28]. BN: $B = 2$, $\lambda_{min} = 0$, $\lambda_{max} = 2$, $k\text{-}NN = 10$, $s = 10^{-3}$.

	Codebook size		
	1024	2048	4096
BN	51.8	52.9	54.4
H-BoW	50.6	51.3	51.4

4.3.2. Bin quantization

We next investigate how BossaNova classification performance is affected by the number of bins (B). Using $M = 4096$, the number of bins is varied among 2, 4 and 6. The results of our experiments are shown in Table 9.

First, we observe that increasing the number of bins yields a slight amelioration in performance. However, the growth depends on the topic of MIRFLICKR dataset: for 30 out of 38 concepts the performance increases by 0.2%–1.9% and for 3 isolated concepts (*bird*(r), *car*(r), *sea*(r)) the performance decreases slightly, by 0.2%.

Once again, further investigations will certainly provide optimized parameters but with a higher complexity. We handled default parameters to 2 here in order to get compact representations.

Table 9: Bin quantization influence on BossaNova MAP (%) performances on MIRFLICKR dataset [28]. We use $M = 4096$, $\lambda_{min} = 0$, $\lambda_{max} = 2$, $k\text{-}NN = 10$, $s = 10^{-3}$.

	Number of Bins		
	$B = 2$	$B = 4$	$B = 6$
MAP	54.4	54.7	**54.9**

4.3.3. Minimum Distance α_m^{min}

We also study the effects of the minimum distance α_m^{min} on BossaNova classification performance. Using the test values of BossaNova parameters (i.e., $B = 2$, $M = 4096$, $\lambda_{max} = 2$, and for semi-soft coding $k\text{-}NN = 10$), we set λ_{min} based on Figure 4.

For $\lambda_{min} = 0.4$ and $\lambda_{max} = 2$, corresponding to 95% of the total SIFT descriptors on the whole dataset, we obtain a MAP = 54.9% which is better than the range of $\lambda_{min} = 0$ and $\lambda_{max} = 2$ (MAP = 54.4%). That is in accordance with our intuition in Section 3.3.

Interestingly, we observe considerable improvements for the most of the concepts (up to 1%) and also a decrease for some ones (up to 0.5%). That suggests that setting a λ_{min} and even λ_{max} per visual word seems to be useful to exploit as future research.

4.3.4. Scalability Issues

When applied in a classification context, the proposed BossaNova representation is used in conjunction to Gauss-ℓ_2 non-linear kernels, because we empirically notice that non-linear features maps boost performances (see Section 4). Non-linear kernels are known to be slower for training and testing. Note that for the databases we evaluate (MIRFLICKR, ImageCLEF 2011, PASCAL VOC 2007, 15-Scenes), using non-linear kernels was still reasonable for training and testing. However, it becomes intractable for large-scale problems, *i.e.* databases with more than one million images.

We want to stress that recent works focused on approximating non-linear kernels by linear ones, by providing approximated features maps [40, 41]. In most of the case, the approximated representations reach about the same level of performances than the exact kernels. Therefore, we hope that using these strategies upon BossaNova is a viable way to handle large-scale classification tasks. The precise evaluation of those compromises is part of our future works.

5. Application: Pornography Detection

We have evaluated our approach in a real-world application, pornography detection. Pornography is less straightforward to define than it may seem at first, since it is a high-level semantic category, not easily translatable in terms of simple visual characteristics. Though it certainly relates to nudity, pornography is a different concept: many activities which involve a high degree of body exposure (swimming, boxing, sunbathing, etc.) have nothing to do with it. That is why systems based on skin detection [42] often accuse false positives in contexts like beach shots or sports.

A commonly used definition is that pornography is the portrayal of *explicit* sexual matter with the *purpose* of eliciting arousal. That raises several challenges. First and foremost what threshold of explicitness must be crossed for the work to be considered pornographic? Some authors deal with that issue by further dividing the classes [43] but that not only fall short of providing a clear cut definition, but also complicates the classification task. The matter of purpose is still more problematic, because it is not an objective property of the document. Here, we have opted to keep the evaluation conceptually simple, by assigning only two classes (porn and non-porn). On the other hand, we took great care to make them representative.

The Pornography dataset contains nearly 80 hours of 400 pornographic and 400 non-pornographic videos. For the pornography class, we have browsed websites which only host that kind of material (solving, in a way, the matter of purpose). The dataset consists of several genres of pornography and depicts actors of

many ethnicities, including multi-ethnic ones. For the non-pornography class, we have browsed general-public purpose video network and selected two samples: 200 videos chosen at random (which we called "easy") and 200 videos selected from textual search queries like "beach", "wrestling", "swimming", which we knew would be particularly challenging for the detector ("difficult"). Figure 8 shows selected frames from a small sample of the dataset, illustrating the diversity of the pornographic videos and the challenges of the "difficult" non-pornographic ones.

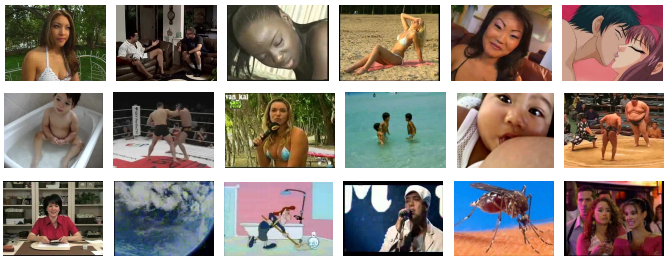


Figure 8: Illustration of the diversity of the pornographic videos (top row) and the challenges of the "difficult" non-pornographic ones (middle row). The easy cases are shown at bottom row. The huge diversity of cases in both pornographic and non pornographic videos makes that task very challenging.

We preprocess the dataset by segmenting videos into shots. An industry-standard segmentation software[8] has been used. On average there are 20 shots per video. As it is often done in video analysis, a key frame is selected to summarize the content of the shot into a static image. Although there are sophisticated ways to choose the key frame, in this proof-of-concept application, we opted to simply selected the middle frame of each video shot.

In the experiments, we follow the experimental setup applied on our previous work on BOSSA [16]. Our main aim is to compare the performance of our proposed method in this paper with our previous one. As a low-level local descriptor, we employ HueSIFT [44], a SIFT variant including color information, which is particularly relevant for our dataset. The 165-dimensional HueSIFT descriptors are extracted densely every 6 pixels. For a fair comparison, we use the same vocabulary M constructed in [16] by k-means clustering algorithm, with M fixed as 256.

For classification, we apply the setup described in Section 4 and we use a 5-fold cross-validation to tune the best C parameter. We report the image classification performance by using the MAP, and the video classification by accuracy rate, where the final video label is obtained by majority voting over the images. Table 10 shows the results of our experiments over Pornography dataset, and details the parameter settings for each method.

Table 10: Comparison of the proposed BossaNova with BOSSA and BoW methods on the Pornography dataset. MAP (%) is computed at image classification level, and Accuracy rate is reported for video classification. For each method, we use their tested configuration parameters, namely (1) BoW: $M = 256$, (2) BOSSA and BossaNova: $M = 256$, $B = 10$, $\lambda_{min} = 0$, $\lambda_{max} = 3$.

	MAP (frames)	Acc. rate (videos)
BoW [5]	91.4 ± 1	83.0 ± 3
BOSSA [16]	94.6 ± 1	87.1 ± 2
BN (ours)	$\mathbf{96.4} \pm 1$	$\mathbf{89.5} \pm 1$

Once again, BossaNova outperforms both BoW and BOSSA representations. Comparing BOSSA with BoW, we already notice a considerable improvement of 3.2% and 4.1% for image and video classification,

[8] http://www.stoik.com/products/svc/

respectively. If we now compare BossaNova with BOSSA, we also observe a considerable increase of 1.8% and 2.4% for image and video classification, respectively. That confirms the advantages introduced by BossaNova representation.

Here, it is instructive to study the fail cases. First, we inspect the misclassified non-pornographic videos. That corresponds to very challenging non-pornographic videos: breastfeeding sequences, sequences of children being bathed, and beach scenes. BoW gave a wrong classification for almost all those clips. The analysis of the most difficult pornographic videos revealed that the method has difficult when the videos are of very poor quality (typical of amateur porn, often uploaded from webcams) or when the clip is only borderline pornographic, with few explicit elements. BoW also had difficulty with those clips, misclassifying many of them.

Moreover, it is interesting to see that for all three methods the video classification scores are inferior to the image classification scores. That can be explained by the fact that some pornographic videos have the additional difficulty of having very few shots with pornographic content (typically 1 or 2 takes among several dialog shots or cut scenes), giving no allowance for classification errors.

6. Conclusion

We have introduced in this paper a visual data classification scheme based on a novel representation that enriches the Bag-of-Words model.

BossaNova representation is interesting from the conceptual, technical and empirical points of view. From a conceptual point of view, its elegant non-parametric conception avoids unnecessary hypothesis about the data distribution. From a technical point of view, the simple vector computation, the ease of implementation and the relatively compact feature vector obtained are non-negligible advantages, especially when tackling datasets which are becoming progressively larger in scale and scope. The empirical comparisons in concept detection, both in a very general task using the MIRFLICKR, ImageCLEF 2011, PASCAL VOC 2007 and 15-Scenes datasets, and in a the specialized task of pornography detection, show the advantage of BossaNova when compared to both traditional techniques and cutting-edge approaches.

In addition, BossaNova geometric properties lead us to predict an interesting complementarity with the Fisher Vector representations, which was confirmed empirically on several standard datasets.

Acknowledgements

We thank Florent Perronnin and Jorge Sánchez for the attentive support in understanding their Fisher Vector method. We thank David Picard for the Java Machine Learning library. Funding is provided by CAPES/COFECUB 592/08/10, CNPq 14.1312/2009-2, ANR 07-MDCO-007-03 and FAPESP.

References

[1] A. Smeulders, M. Worring, S. Santini, A. Gupta, R. Jain, Content Based Image Retrieval at the End of the Early Years, PAMI 22.
[2] M. Lew, N. Sebe, C. Djeraba, R. Jain, Content-Based Multimedia Information Retrieval: State of the Art and Challenges, ACM Transactions on Multimedia Computing, Communications, and Applications 2.
[3] R. Datta, D. Joshi, J. Li, J. Wang, Image Retrieval: Ideas, Influences, and Trends of the New Age, ACM Computing Surveys 40.
[4] P. Gosselin, M. Cord, S. Philipp-Foliguet, Combining visual dictionary, kernel-based similarity and learning strategy for image category retrieval, CVIU 3.
[5] J. Sivic, A. Zisserman, Video Google: A Text Retrieval Approach to Object Matching in Videos, in: ICCV, vol. 2, 2003.
[6] J. van Gemert, C. Veenman, A. Smeulders, J.-M. Geusebroek, Visual Word Ambiguity, PAMI 32.
[7] N. Sebe, I. Cohen, A. Garg, T. Huang, Machine Learning in Computer Vision, Springer Verlag, 2005.
[8] M. Cord, P. Cunningham, Machine Learning Techniques for Multimedia: Case Studies on Organization and Retrieval, Cognitive Technologies, Springer, 2008.
[9] Y. Boureau, F. Bach, Y. LeCun, J. Ponce, Learning mid-level features for recognition, in: CVPR, 2010.
[10] S. Lazebnik, C. Schmid, J. Ponce, Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories, in: CVPR, 2006.
[11] F. Perronnin, C. Dance, Fisher Kernels on Visual Vocabularies for Image Categorization, in: CVPR, 2007.

[12] F. Perronnin, J. Sánchez, T. Mensink, Improving the Fisher Kernel for Large-Scale Image Classification, in: ECCV, 2010.
[13] J. Krapac, J. Verbeeky, F. Jurie, Modeling Spatial Layout with Fisher Vectors for Image Categorization, in: ICCV, 2011.
[14] H. Jégou, M. Douze, C. Schmid, P. Pérez, Aggregating local descriptors into a compact image representation, in: CVPR, 2010.
[15] X. Zhou, K. Yu, T. Zhang, T. Huang, Image classification using super-vector coding of local image descriptors, in: ECCV, 2010.
[16] S. Avila, N. Thome, M. Cord, E. Valle, A. Araújo, BOSSA: extended BoW formalism for image classification, in: ICIP, 2011.
[17] W. Y. Ma, B. S. Manjunath, NETRA: A toolbox for navigating large image databases, ACM Multimedia Systems 7 (3).
[18] J. Fournier, M. Cord, S. Philipp-Foliguet, RETIN: A content-based image indexing and retrieval system, Pattern Analysis and Applications Journal, Special issue on image indexation 4 (2/3).
[19] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, L. V. Gool, A comparison of affine region detectors, IJCV 65 (1/2).
[20] K. Chatfield, V. Lempitsky, A. Vedaldi, A. Zisserman, The devil is in the details: an evaluation of recent feature encoding methods, in: BMVC, 2011.
[21] J. Mairal, F. Bach, J. Ponce, G. Sapiro, Online Learning for Matrix Factorization and Sparse Coding, Journal of Machine Learning Research 11.
[22] J. Yang, K. Yu, Y. Gong, T. Huang, Linear spatial pyramid matching using sparse coding for image classification, in: CVPR, 2009.
[23] L. Liu, L. Wang, X. Liu, In Defense of Soft-assignment Coding, in: ICCV, 2011.
[24] Y. Boureau, N. Le Roux, F. Bach, J. Ponce, Y. LeCun, Ask the locals: multi-way local pooling for image recognition, in: ICCV, 2011.
[25] T. Jaakkola, D. Haussler, Exploiting Generative Models in Discriminative Classifiers, in: In Advances in Neural Information Processing Systems, 1998.
[26] D. Picard, P. Gosselin, Improving Image Similarity With Vectors of Locally Aggregated Tensors, in: ICIP, 2011.
[27] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, Y. Gong, Locality-constrained linear coding for image classification, in: CVPR, 2010.
[28] M. Huiskes, M. Lew, The MIR Flickr Retrieval Evaluation, in: MIR, 2008.
[29] S. Nowak, K. Nagel, J. Liebetrau, The CLEF 2011 Photo Annotation and Concept-based Retrieval Tasks, in: CLEF, 2011.
[30] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, A. Zisserman, The PASCAL Visual Object Classes Challenge 2007 Results, 2007.
[31] A. Vedaldi, B. Fulkerson, VLFeat – An open and portable library of computer vision algorithms, in: ACM International Conference on Multimedia, 2010.
[32] M. J. Huiskes, B. Thomee, M. S. Lew, New Trends and Ideas in Visual Concept Detection: The MIR Flickr Retrieval Evaluation Initiative, in: MIR, 2010.
[33] M. Guillaumin, J. Verbeek, C. Schmid, Multimodal semi-supervised learning for image classification, in: CVPR, 2010.
[34] E. Mbanya, S. Gerke, C. Hentschel, P. Ndjiki-Nya, Sample Selection, Category Specific Features and Reasoning, in: CLEF, 2011.
[35] D.-D. Le, S. Satoh, NII, Japan at ImageCLEF 2011 Photo Annotation Task, in: CLEF, 2011.
[36] K. E. A. V. de Sande, C. G. M. Snoek, The University of Amsterdam's Concept Detection System at ImageCLEF 2011, in: CLEF, 2011.
[37] Y. Su, F. Jurie, Semantic Contexts and Fisher Vectors for the ImageCLEF 2011 Photo Annotation Task, in: CLEF, 2011.
[38] A. Binder, W. Samek, M. Kloft, C. Müller, K.-R. Müller, M. Kawanabe, The Joint Submission of the TU Berlin and Fraunhofer FIRST (TUBFI) to the ImageCLEF2011 Photo Annotation Task, in: CLEF, 2011.
[39] R. Jain, The Art of Computer Systems Performance Analysis: techniques for experimental design, measurement, simulation, and modeling, 1991.
[40] A. Vedaldi, A. Zisserman, Efficient Additive Kernels via Explicit Feature Maps, PAMI 34 (3).
[41] C. Williams, M. Seeger, Using the Nyström Method to Speed Up Kernel Machines, in: Advances in Neural Information Processing Systems 13, 2001.
[42] W. Kelly, A. Donnellan, D. Molloy, Screening for Objectionable Images: A Review of Skin Detection Techniques, in: IMVIP, 2008.
[43] T. Deselaers, L. Pimenidis, H. Ney, Bag-of-visual-words models for adult image classification and filtering, in: ICPR, 2008.
[44] K. van de Sande, T. Gevers, C. Snoek, Evaluating Color Descriptors for Object and Scene Recognition, PAMI 32 (9).