Supervised Learning of Gaussian Mixture Models for Visual Vocabulary Generation

Basura Fernando, Elisa Fromont, Damien Muselet, Marc Sebban

Université de Lyon, F-42023, Saint-Étienne, France
CNRS, UMR 5516, Laboratoire Hubert Curien, F-42000, Saint-Étienne, France
Université de Saint-Étienne, Jean-Monnet, F-42000, Saint-Étienne, France

Abstract

The creation of semantically relevant clusters is vital in bag-of-visual words models which are known to be very successful to achieve image classification tasks. Generally, unsupervised clustering algorithms, such as K-means, are employed to create such clusters from which visual dictionaries are deduced. K-means achieves a *hard assignment* by associating each image descriptor to the cluster with the nearest mean. By this way, the within-cluster sum of squares of distances is minimized. A limitation of this approach in the context of image classification is that it usually does not use any supervision that limits the discriminative power of the resulting visual words (typically the centroids of the clusters). More recently, some supervised dictionary creation methods based on both supervised information and data fitting were proposed leading to more discriminative visual words. But, none of them consider the *uncertainty* present at both image descriptor and cluster levels. In this paper, we propose a supervised learning algorithm based on a Gaussian Mixture model which not only generalizes the K-means algorithm by allowing *soft assignments*, but also exploits supervised information to improve the discriminative power of the clusters. Technically, our algorithm aims at optimizing, using an EM-based approach, a convex combination of two criteria: the first one is unsupervised and based on the likelihood of the training data; the second is supervised and takes into account the purity of the clusters. We show on two well known datasets that our method is able to create more relevant clusters by comparing its behavior with the state of the art dictionary creation methods.

Keywords: Bags of visual words, Supervised Gaussian Mixture Model, Dictionary Generation, Expectation-Maximization algorithm.

Email addresses: palamandadige.basura.fernando@etu.univ-st-etienne.fr (Basura Fernando), elisa.fromont@univ-st-etienne.fr (Elisa Fromont), damien.muselet@univ-st-etienne.fr (Damien Muselet), marc.sebban@univ-st-etienne.fr (Marc Sebban)

This work is part of the ongoing ANR SATTIC 07-1_184534 research project.

1. Introduction

Many of the object recognition and scene classification algorithms first aggregate local statistics computed from images to induce object models before classifying images using supervised techniques such as SVM [1]. Bag of visual words (BoW) approaches have indisputably become a reference in the image processing community [1, 2, 3]. In this context, visual dictionary creation constitutes the first crucial step in BoW methods usually known as *vector quantization*. Generally, clustering algorithms are used to achieve this task in the descriptor space. Each cluster representative (typically the centroid) is considered as a visual word of the visual dictionary. The K-means clustering algorithm [1, 4] is the most common method to create such visual dictionaries even though other unsupervised methods such as K-median clustering [5], mean-shift clustering [6], hierarchical K-means [7], agglomerative clustering [8], radius based-clustering [6, 9], or regular lattice-based strategies [10] have also been used. One of the common features of these unsupervised methods is that they only optimize an objective function fitting to the data but ignoring their class information. Therefore, this reduces the discriminative power of the resulting visual dictionaries. For example, the K-means algorithm minimizes the within-cluster sum of squares of distances without considering the class of the data (*i.e.* the label of the image the descriptor has been extracted from). Without any supervision, only one dictionary can thus be created for all the categories in the dataset, usually called *universal dictionary* or *universal vocabulary*.

To create more discriminative visual words, one solution consists in using supervised approaches. In this context, some methods have been proposed to create class specific or concept specific multiple dictionaries. For instance, in [11, 12], an image is characterized by a set of histograms - one per class - where each histogram describes whether the image content is well modeled by the dictionary of that specific class. But one limitation of these methods is that they ignore the correlation in the D-dimensional space representing the descriptors. For instance, descriptors of a cat's eye and a dog's eye are highly correlated and are likely complementary to learn the generic concept of *eye*. In [13], a dictionary creation method based on the learning of a set of classifiers - one per category - is presented, but they are learned independently. In [14], class specific dictionaries are first created and then merged. Despite the fact that, once again, this is achieved without considering the correlations, visual words are redundant because they can occur in different class specific dictionaries. In [15], Zhang et al. solve this problem of redundancy using a boosting procedure and learn multiple dictionaries with complementary discriminative power. A disadvantage of this approach is that the number of visual dictionaries corresponds to the number of boosting iterations. Too many iterations (dictionaries) obviously leads to overfitting and the authors do not provide any theoretical result to determine the optimum number of visual dictionaries.

Recently, some methods have been proposed to create a unique universal dictionary while using supervised information. In [16], the authors use a Gaussian mixture model to take advantage of a soft assignment (unlike K-means)

and try to maximize the discriminative ability of visual words using image labels. They use a supervised logistic regression model to modify the parameters of the Gaussian mixture. In [17], the authors optimize jointly a single sparse dictionary (using the L1 norm) and a classification model in a mixed generative and discriminative formulation. In both methods [16, 17], each image descriptor is assumed to have been generated from a single object category or a single class. We claim that this is a too strong assumption that limits the efficiency of the resulting vocabularies. We will show experimental evidences of this limitation in this paper. In [18], the authors present a general approach to vector quantization that tries to minimize the information loss under the assumption that each descriptor approximates a sufficient statistic for its class label. Once again, this is a strong assumption notably for images described by local descriptors (*e.g.* SIFT descriptors [19]). In [20], randomized clustering forests are used to build visual dictionaries. This approach is first based on the supervised learning of small ensembles of trees which contain a lot of valuable information about locality in descriptor space. Then, ignoring the class labels, these trees are used as simple spatial *partitioners* that assign a distinct region label to each leaf. The disadvantage of this method is that the resulting clusters suffer from a lack of generalization ability since only normalized mutual information is used as splitting criterion during the construction process. Moreover, the method totally ignores the likelihood of the data which is important in BoW image representation. In [21], the authors present an incremental gradient descent-based clustering algorithm which optimizes the visual word creation by the use of the class label of training examples. This method also assumes that each descriptor is generated from a single class and ignores the correlation in the D-dimensional space representing the descriptors. Even though all the previous supervised methods allow us to significantly outperform traditional dictionary creation methods, they often assume that each local descriptor belongs to a single object category. Moreover, they do not try to optimize at the same time the *likelihood* of the training data and the *purity* of the clusters.

By integrating both criteria in the objective function to optimize, we claim that it is possible to jointly manage the two kinds of uncertainty the descriptors are usually subject to: the *cluster uncertainty* and the *class uncertainty*. The *cluster uncertainty* expresses the fact that it is something of an oversimplification to achieve a hard assignment (like K-means) during the construction of the clusters. For instance, a wheel can contribute to the construction of a visual word representing either a wheel of a bicycle or a wheel of a stroller, with different probabilities of membership. Taking into account this uncertainty during the creation of the visual dictionary can be realized using soft clustering such as Gaussian Mixture (GM) models, which have already been shown to outperform hard assignment-based approaches [9]. The *class uncertainty* can be illustrated by the following example: a brown patch descriptor may have been generated from both dog and cow classes. So given a brown patch descriptor, it would be short-sighted to label it by only one of these two classes. This type of uncertainty is usually ignored at the image descriptor level in most of the supervised dictionary creation algorithms. To overcome this limitation, we

propose to exploit the probability for each descriptor to belong to each class. The estimation of these probabilities can be achieved by resorting to learned classifiers and approximating the Bayesian rule.

It is important to note that the discriminative power of a cluster in the context of image classification is a function of its purity which depends on both the *cluster uncertainty* and the *class uncertainty*. Although several supervised or semi-supervised GM models have been proposed in various domains [22, 23, 24, 25] and in visual dictionary creation [12, 16, 26, 27], none of them addresses the problem of this two-fold uncertainty and none jointly optimizes generalization and discriminative abilities of clusters. To solve these limitations, we present in this paper a new dictionary learning method which optimizes a convex combination of the likelihood of the (labeled and unlabeled) training data and the purity of the clusters. We show that our GM-based method has the ability to quantify both uncertainties during the dictionary creation process and leads to semantically relevant clusters.

The rest of this paper is organized as follows: after having introduced some notations and definitions in Section 2, we present in Section 3 our GM model and the corresponding objective function which will be optimized using an Expectation-Maximization algorithm. Section 4 is devoted to an experimental analysis. We evaluate our method using the *PASCAL VOC-2006* dataset [28] and the *Caltech-101* dataset [29] and show that it significantly outperforms the state of the art dictionary creation approaches. We conclude this paper in Section 5 by outlining promising lines of research on dictionary learning.

2. Notations and Definitions

Let $X = \{x_k | k = 1 \ldots n \ , x_k \in \mathbb{R}^D\}$ be the set of training examples, *i.e.* descriptors extracted from images and living in a D-dimensional space (*e.g.* SIFT descriptors usually live in a 128-dimensional space). Let C $= \{c_j | j = 1 \ldots$ R $\}$ be the set of classes (*i.e.* the labels of the original images). Since labeling data can be very expensive, we assume that X may contain both labeled and non-labeled data. Let $S = \{s_i | i = 1 \ldots$ I$\}$ be the set of clusters, where $I > 1$.

A Gaussian Mixture (GM) model is a generative model where it is assumed that data are *i.i.d* from an unknown probability density function [30]. In our approach, the distribution over the set of clusters is modeled using a GM model $\Theta = \{\theta_i, i = 1 \ldots I\}$ where $\theta_i = \{\mu_i, \Sigma_i, w_i\}$ are the model parameters of the i^{th} Gaussian (corresponding to the cluster s_i). Here, μ_i is the mean, Σ_i is the covariance matrix and w_i is the weight of the i^{th} Gaussian. Given the GM model defined by its parameters Θ, the probability of the descriptor $x_k \in X$ is computed as follows:

$$p(x_k|\Theta) = \sum_{i=1}^{I} w_i \times N_{\mu_i,\Sigma_i}(x_k), \tag{1}$$

where $N_{\mu,\Sigma}(x)$ is the multivariate Gaussian distribution, such that

$$N_{\mu,\Sigma}(x) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{\frac{1}{2}}} exp(-\frac{1}{2}(x-\mu)^{'}\Sigma^{-1}(x-\mu)). \quad (2)$$

In a GM model, the posterior probability $p(s_i|x_k,\Theta)$ is calculated as follows:

$$p(s_i|x_k,\Theta) = \frac{w_i \times p(x_k|s_i,\theta_i)}{\sum_t w_t \times p(x_k|s_t,\theta_t)} \quad (3)$$

subject to

$$\Sigma_i w_i = 1, \quad (4)$$

where $p(x_k|s_i,\theta_i)$ is the probability for x_k to belong to the i^{th} Gaussian and that is exactly equal to $N_{\mu_i,\Sigma_i}(x_k)$ given by Eq.2.

Usually, GM models are trained using the Expectation Maximization (EM) algorithm to find maximum likelihood parameters [31]. This is achieved by maximizing the log likelihood $\mathcal{L}(X)$ of the training set X defined as follows:

$$\mathcal{L}(X) = log(\prod_{i=1}^{n} p(x_i|\Theta)) = \sum_{i=1}^{n} log(p(x_i|\Theta)). \quad (5)$$

Since $p(s_i|x_k,\Theta)$ and $p(x_k|\Theta)$ are unknown, they will be estimated by the GM and will be denoted by $\hat{p}(s_i|x_k,\Theta)$ and $\hat{p}(x_k|\Theta)$ respectively in the rest of the paper. Note that we will use the same notation for all the other unknown probabilities.

Unlike K-means algorithm which builds clusters in which each descriptor belongs to the cluster with the nearest mean, a GM model has the ability to allow soft assignments by providing a probability for a given instance to belong to each cluster (thanks to Eq.3). It can be very useful to exploit these probabilities not only during the visual dictionary construction (as we will see in the next section) but also in the recognition step as shown in Figure 1.

3. Supervised GM-based Dictionary Learning

3.1. Intuitive idea

We claim that a relevant visual dictionary must be composed of visual words which are not only *specific* enough to be sufficiently discriminative, but also *general* enough to avoid overfitting phenomena. Note that to fulfill these two required conditions, one has to find a good compromise between the *cluster purity* and the *likelihood* of the data. If only the purity of the clusters is optimized using supervised information and without considering the likelihood of the data, the resulting clusters will be very discriminative but the generalization behavior of the BoW model will be likely subject to an overfitting phenomenon. On the other hand, when no class information is considered (like in a standard

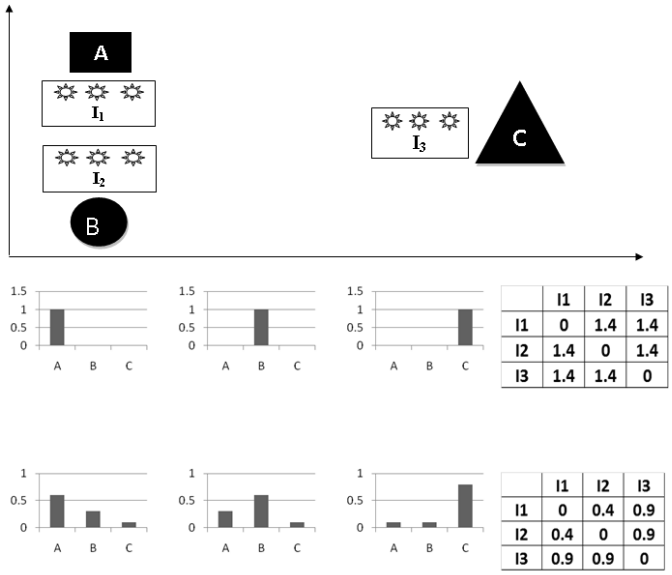

Figure 1: Soft assignment vs. hard assignment. Let us assume that the visual dictionary is made of three visual words A, B and C and that three descriptors have been extracted from three images I_1, I_2 and I_3. With a hard assignment, the label A is assigned to the descriptors of I_1, B to those of I_2 and C to those of I_3. By representing each image by a (normalized) histogram, the (Euclidean) distances between I_1, I_2 and I_3 are all the same while I_1 and I_2 are closer in the D-dimensional space. Applying a soft assignment (at the bottom) allows us to better reflect the realities of the situation.

K-means algorithm), the resulting clusters will tend to be too general and each cluster (visual word) might represent (too) many classes, reducing the discriminative ability of the visual dictionary. Our objective is to optimize not only the likelihood of the data but also the cluster purity by resorting to a convex combination of both criteria optimized by a standard EM-based approach. The aim is to find a good trade-off allowing us to generate visual words that are discriminative enough to classify instances of various concepts and general enough to represent an object model. The motivation behind our method is graphically presented in Fig.2. The next sections are devoted to the presentation of the technical aspects to reach this goal.

3.2. Joint optimization of the likelihood and the purity

In our method, we use a GM model where each Gaussian models a visual word. Contrary to standard GM-based approaches, our supervised GM algorithm not only takes advantage of the soft assignment allowed by a GM model, but also integrates in the objective function a term estimating the purity of the clusters. This purity can be defined in different ways from the entropy of the clusters. In this paper, as it is also usually done to change a distance into a similarity measure, we have used the negative logarithm to define the purity

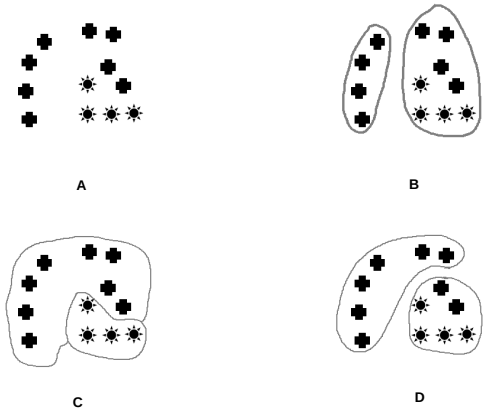

Figure 2: Rough illustration of the idea behind our approach. (A) Original data consisting of two classes. (B) Totally unsupervised clustering (*e.g.* K-Means). The purity of the cluster on the right is very low but the global likelihood (w.r.t. the centroids of the clusters) is optimized. (C) Totally supervised approach. The purity is optimal but the resulting clusters are too specific to allow some generalization ability. (D) Combination of both likelihood and purity criteria leading to a "good" trade-off.

$F(s_i|\theta_i)$ from the entropy of the cluster s_i:

$$F(s_i|\theta_i) = -log(-\sum_j \hat{p}(c_j|s_i,\theta_i) \times log(\hat{p}(c_j|s_i,\theta_i))) + \phi, \quad (6)$$

where $\hat{p}(c_j|s_i,\theta_i)$ is the estimated probability of class c_j given cluster s_i which depends on the GM parameters θ_i of cluster s_i[2]. ϕ is a constant equal to $log(log(R))$ if $R > 2$, otherwise $\phi = 0$. This constant makes sure that $F(s_i)$ is a positive function. The higher the value of $F(s_i)$, the purer the cluster.

$\hat{p}(c_j|s_i)$ is estimated using its marginal distribution expansion w.r.t. all possible samples $x \in X$ (see Appendix A for more details):

$$\hat{p}(c_j|s_i) = \frac{\sum_k \hat{p}(x_k|c_j) \times \hat{p}(x_k|s_i)}{\sum_t \sum_k \hat{p}(x_k|c_t) \times \hat{p}(x_k|s_i)}. \quad (7)$$

The above equation is nothing more than a generalization of the proportion of examples which belong to a class c_j given a cluster s_i. But since we are using a GM, probabilities are used to estimate $\hat{p}(c_j|s_i)$ rather than counting examples. To do that, we first need to estimate $\hat{p}(x_k|s_i)$. This can be done using the posterior probability $\hat{p}(s_i|x_k,\Theta)$ (*i.e.* the so-called *cluster uncertainty*) given by Eq.3. Second, we have to estimate $\hat{p}(x_k|c_j)$. To achieve this task, we suggest to learn a classifier, use it to compute the posterior probability $\hat{p}(c_j|x_k)$ (*i.e.*

[2]To simplify the notations, θ_i will be omitted when it is explicitly related in a formula to cluster s_i. This is the case *e.g.* for $F(s_i|\theta_i)$ or $\hat{p}(c_j|s_i,\theta_i)$.

the so-called *class uncertainty*), and apply the Bayesian rule to get the estimate $\hat{p}(x_k|c_j)$. Note that this way to proceed allows the computation of the purity $F(s_i)$ even in situations where the training set is composed of both *labeled* and *unlabeled* examples. Indeed, by taking advantage of the learned classifier to estimate $\hat{p}(c_j|x_k)$, it is possible to deal with a set X containing unlabeled instances x_k and therefore reduce the risk of errors and the expensive cost of manually and hardly labeling a large amount of data.

As mentioned before, the GM parameters are generally estimated by only optimizing the log likelihood of the data using the expectation maximization (EM) algorithm. Here, our objective is to find the parameters Θ by optimizing not only the likelihood of Eq.5 but also the cluster purity of Eq.6. By this way, the two-fold uncertainty $\hat{p}(c_j|x_k)$ and $\hat{p}(s_i|x_k)$ are taken into consideration in the estimation. The objective function we aim at maximizing is defined as follows:

$$J(\Theta) = (1-\alpha) \times \sum_{x_k \in X} log(\hat{p}(x_k|\Theta)) + \alpha \times \sum_{i}^{I} log(F(s_i)), \tag{8}$$

where α ($0 \leq \alpha \leq 1$) is a control parameter of the algorithm that determines the level of supervision authorized in the GM. If $\alpha = 0$, the algorithm is totally unsupervised and so only optimizes the likelihood. In this case, it boils down to learning a standard GM model with the limitations already mentioned in the context of image classification. If $\alpha = 1$, the optimization process only aims at building pure clusters with the obvious risk to lead to overfitting phenomena. Therefore, α plays an important role in our method and deserves a special attention in order to find a good compromise between these two extreme situations (see Section 4.5 for an experimental study).

Our objective is to find the optimal model parameters Θ^* that maximize the above objective function such that:

$$\Theta^* = argmax_{\Theta} J(\Theta). \tag{9}$$

To do this, we compute the derivatives with respect to each model parameter $\mu_i, \sum_i, w_i$ for $i = 1 \dots I$. Since we are constrained by Eq.4, we use a Lagrange multiplier λ as follows:

$$\widetilde{J}(\Theta) = J(\Theta) + \lambda(1 - \sum_{i} w_i). \tag{10}$$

Computing the partial derivatives of Eq.10 w.r.t. $\mu_i, \sum_i, w_i$ respectively and equating them to zero allows us to find the optimal parameters. The general formula of the derivative of Eq.10 w.r.t. μ_i **or** $\sum_i$ (noted $(\mu_i, \sum_i)$) is given by (see Appendix B for more details):

$$\frac{\partial \widetilde{J}}{\partial_{(\mu_i,\sum_i)}} = \sum_k \{(1-\alpha)\hat{p}(s_i|x_k) + \alpha B_i \sum_j a_i^j \times \hat{p}(x_k|c_j) \times \hat{p}(x_k|s_i)\} \times \frac{\partial}{\partial(\mu_i,\sum_i)} log(\hat{p}(x_k|s_i)) \quad (11)$$

where B_i, the normalization parameter of the i^{th} Gaussian, is given by

$$B_i = \frac{-1}{F(s_i)[\sum_j \hat{p}(c_j|s_i) \times log(\hat{p}(c_j|s_i))]} \times \frac{1}{\sum_t \sum_k \hat{p}(x_k|c_t) \times \hat{p}(x_k|s_i)}, \quad (12)$$

and where

$$a_i^j = 1 + log(\hat{p}(c_j|s_i)). \quad (13)$$

Using either μ_i or Σ_i in Eq.11, and equating to zero we find the optimal parameters of each Gaussian such that:

$$\mu_i = \frac{\sum_k \{(1-\alpha)\hat{p}(s_i|x_k) + \alpha B_i \sum_j a_i^j \hat{p}(x_k|s_i)\hat{p}(x_k|c_j)\} x_k}{\sum_k \{(1-\alpha)\hat{p}(s_i|x_k) + \alpha B_i \sum_j a_i^j \hat{p}(x_k|s_i)\hat{p}(x_k|c_j)\}}, \quad (14)$$

$$\Sigma_i = \frac{\sum_k \{(1-\alpha)\hat{p}(s_i|x_k) + \alpha B_i \sum_j a_i^j \hat{p}(x_k|s_i)\hat{p}(x_k|c_j)\} A_i^k}{\sum_k \{(1-\alpha)\hat{p}(s_i|x_k) + \alpha B_i \sum_j a_i^j \hat{p}(x_k|s_i)\hat{p}(x_k|c_j)\}}, \quad (15)$$

where $A_i^k = (\mu_i - x_k)(\mu_i - x_k)^t$.

Computing the derivatives of Eq.10 with respect to parameter w_i (see Appendix B for more details), we get

$$\frac{\partial \widetilde{J}(\Theta)}{\partial w_i} = \Sigma_k \frac{\hat{p}(x_k|s_i)}{\Sigma_t w_t \times \hat{p}(x_k|s_t)} - \lambda, \quad (16)$$

and equating to zero, we get

$$w_i = \frac{\sum_k \hat{p}(s_i|x_k)}{n}. \quad (17)$$

In the next section, we will use Equations 14, 15 and 17 to update the parameters of the GM model using an EM-based iterative learning algorithm.

3.3. EM-based learning algorithm

After an initialization step, our EM-based algorithm iteratively performs (as usually) an expectation (E) step and a maximization (M) step. The E-step consists of estimating from the training set the expected value of the parameters, which are then used in the M-step (i) to maximize the expected value of our objective function J and (ii) to estimate the new model parameters.

To initialize our GM model, we have to predetermine the number of clusters I (*i.e.* the number of visual words). We also have to provide to the EM-algorithm a first series of estimates for the parameters μ_i, Σ_i and w_i, $i = 1 \dots I$. To do this, we simply run K-means algorithm on the training set X. The mean μ_i corresponds to the centroid of the i^{th} cluster s_i and Σ_i to the corresponding covariance matrix. Finally, the initial weight w_i is calculated by counting the number of training examples located in the cluster s_i and by normalizing it to satisfy Condition 4.

The pseudo-code of our algorithm (called *GEMP*) is presented in Algorithm 1. The convergence of *GEMP* is reached if the objective function J does not increase sufficiently between two iterations. This condition can be verified w.r.t. a given threshold. But note that since J is a weighted average computed from more than 30 examples, we can apply the central limit theorem stating that J asymptotically convergences towards a normal distribution and check the convergence by resorting to a statistical test of average equivalence.

Data: $X = \{x_1 ... x_n\}$, a number of clusters I, and a learned classifier providing $\hat{p}(c_j|x_k)$ $\forall j, k$
Result: Final GM parameters
$t \leftarrow 0$;
Initialization-Step: use of K-means to initialize parameters Θ_0;
while *No convergence* **do**
  E-Step;
  Estimate expected value for $\hat{p}(x_k|s_i)$, $\hat{p}(s_i|x_k)$, $\hat{p}(x_k|c_j)$, B_i and a_i^j, $\forall i, j, k$;
  M-Step;
  Update parameters Θ_{t+1} using Equations 14,15 and 17;
  Evaluate objective function $J(\Theta_{t+1})$;
  $t \leftarrow t + 1$;
Return parameters Θ_t;

Algorithm 1: *GEMP* Algorithm.

Figure 3: Sample images drawn from the 10 classes of *PASCAL VOC-2006* dataset.

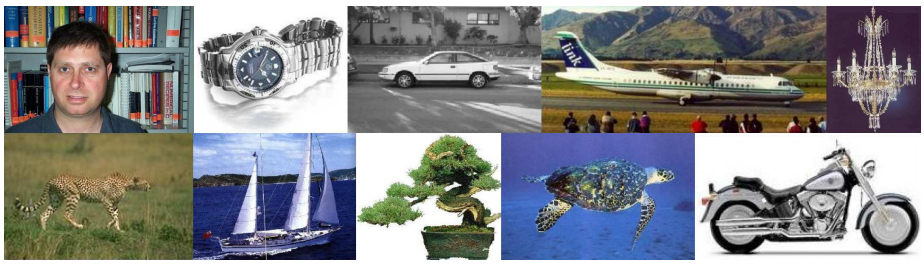

Figure 4: Sample images drawn from the 10 classes of *Caltech-10* dataset.

4. Experiments

4.1. Database

In order to assess the efficiency of our new dictionary creation method, we carry out experiments from the well-known challenging *PASCAL-VOC-2006* dataset [28] and from the *Caltech-10* [29] dataset, as also used in [16]. The first one contains 5,304 images, 2,618 are used as training examples and 2,686 as testing data including 9,507 annotated objects. The second one contains 400 images for training and 2644 for testing. In both cases, ten annotated object classes are provided which are equally distributed between the training and the test sets.

4.2. Data preparation and compared methods

Before learning a visual dictionary, one needs to represent each image by a set of descriptors. To achieve this task, we first apply a Harris-Laplace [32] key point detector using the implementation provided in [33]. Then, SIFT descriptors [19] are generated from these key points. As pointed out in [26], dimensionality reduction is an important step in GM-based BoW image representation. This makes sure that the most relevant directions of the input feature space are identified and that the remaining noisy directions discarded. Moreover, this allows us to substantially reduce the computational complexity of the algorithm. Therefore, we perform a principal component analysis to reduce the dimension

D of the descriptors from 128 to 32.

We compare our algorithm *GEMP* with three other approaches:

- The *K-means* clustering algorithm which is used as a baseline.
- The *SDLM* model introduced in [16] which combines an unsupervised model (a GM) and a supervised model (a logistic regression model) in a probabilistic framework. To avoid having to implement this method and to allow us to simply report the results presented in [16], we use the same experimental setup.
- The incremental gradient descent-based clustering algorithm (*SA* for short) presented in [21] which optimizes the visual word creation by the use of the class label of training examples. Unlike *SDLM*, note that *SA* is a supervised hard-assignment method.

Note that the choice of these methods is driven by our desire to compare *GEMP* with the state of the art dictionary methods, that is with either (i) unsupervised methods (*K-means*) or (ii) supervised methods with hard-assignment (*SA*) or (iii) supervised methods with soft-assignment (*SDLM*).

For both datasets, we build the visual dictionary using 20,000 SIFT descriptors per class leading to a whole training set X of 200,000 examples. Among these data, we only use 40,000 labeled descriptors (*i.e.* 20% of X) to create the dictionary. These labeled examples are obtained from training images using a bounding box surrounding the considered object for the *PASCAL-VOC-2006* dataset (see Figure 3) and using the boundaries of the object provided by the authors for the *Caltech-10* dataset (see Figure 4). The 80% remaining descriptors are generated from the whole image without bounding boxes. Moreover, as explained before, we need in our algorithm to estimate the posterior probability $\hat{p}(c_j|x_k)$ of the descriptors. To do this, we train several classifiers (Random Forests, Decision Trees, Bayesian model, etc.) and keep the most performing one (using a validation set) to estimate $\hat{p}(c_j|x_k)$. For both datasets, the best learner is the Random Forest algorithm [34] which is run with 20 trees and 8 random features are used during the induction process.

4.3. Image classification task

To compare the different visual dictionaries learned by the four approaches and assess their respective discriminative power, we perform an image classification task. For both datasets, we learn 10 classifiers (one for each class against the others) using a Support Vector Machine algorithm and a chi-square kernel as used in [16]. The binary classification performance for each object class is quantitatively estimated by the *average precision*, which is a popular measure that takes into account both *recall* and *precision* [35]. One can express precision as a function of recall, denoted by $p(r)$.

Definition 1. *The average value of $p(r)$ over the entire interval from $r = 0$ to $r = 1$, $\int_0^1 p(r)dr$, is called the average precision.*

Note that to be able to achieve this classification task, an image P has to be represented in the form of a feature vector H. For the hard assignment-based methods (*i.e. K-means* and *SA*), we use a standard normalized term frequency histogram approach. In this case, a component H_i of H corresponds to the proportion of times the i^{th} visual word (representing cluster s_i) is assigned to the descriptors extracted from the image P. More formally,

$$H_i = \frac{1}{|P|} \sum_{x_k \in P} \mathbb{1}_{[s_i = NN(x_k)]}, \tag{18}$$

where $\mathbb{1}_{[s_i=NN(x_k)]}$ is an indicator function which takes the value of 1 if the center of s_i is the nearest neighbor $NN(x_k)$ (using the Euclidean distance) of the descriptor x_k and 0 otherwise, and where $|P|$ is the number of descriptors extracted from the image P.

For the two soft-assignment methods based on a GM model (*i.e. SDLM* and our method *GEMP*), each component is obtained by summing (and normalizing) the conditional probabilities $\hat{p}(s_i|x_k)$ for a descriptor x_k to belong to cluster s_i, that exactly corresponds to the observed frequency of descriptors in cluster s_i. More formally,

$$H_i = \frac{1}{|P|} \times \sum_{x_k \in P} \hat{p}(s_i|x_k). \tag{19}$$

4.4. Experimental results

The mean average precisions (Mean AP) are reported in Tables 1 and 2 for each binary classification problem. In each table, we indicate the number of visual words (parameter I) required to reach the best performance on average. Note that we use a value $\alpha = 0.5$ for *GEMP* (an experimental study on the effect of α is presented in Section 4.5). Several remarks can be made:

- First, for both datasets, our *GEMP* algorithm allows us to outperform all the other supervised and unsupervised methods. Using a Student paired-t test with a Type I error of 5%, we can show that the difference between the mean average precisions is significant in favor of our method. We obtain a precision of 0.8257 for *GEMP* versus 0.6425, 0.6715 and 0.7516 for *K-means*, *SA* and *SDLM* respectively for the *PASCAL-VOC-2006* dataset and 0.9293 for *GEMP* versus 0.8373, 0.8748 and 0.8928 for *K-means*, *SA* and *SDLM* respectively for the *Caltech-10* dataset.

- Second, on the *PASCAL-VOC-2006* dataset, for all the binary classification problems, *GEMP* is better than *SDLM*. Better still, for 9 binary

Object class	K-Means I=2,000	SA I=1,000	SDLM I=400	GEMP I=400
Bicycle	0.7295	0.7081	0.8198	0.8267
Bus	0.4994	0.6457	0.8538	**0.8959**
Car	0.6822	0.6538	0.8122	**0.9038**
Cat	0.7131	0.7119	0.7537	**0.8639**
Cow	0.6358	0.6469	0.7581	**0.8582**
Dog	0.5917	0.6635	0.7234	**0.7617**
Horse	0.6431	0.6966	0.6322	**0.7604**
Motorbike	0.6890	0.6216	0.7946	**0.8332**
Person	0.5773	0.6686	0.6307	**0.6818**
Sheep	0.6905	0.6982	0.7376	**0.8714**
Mean AP	0.6425	0.6715	0.7516	**0.8257**

Table 1: Mean average precision evaluated on *PASCAL VOC-2006* dataset. An average precision in bold font means that *GEMP* significantly outperforms all the other methods using a Student paired-t test with a Type I error of 5%.

problems out of 10, *GEMP* significantly outperforms *SDLM* using a Student paired-t test (the difference is not significant only for the class *Bicycle*). For the *Caltech-10* dataset, *GEMP* is better than the other methods on 9 out of 10 classes and significantly better on 5. This constitutes an experimental evidence of the efficiency of *GEMP* since *SDLM* has already been proven to be very competitive in comparison with state of the art approaches [16]. The main reason of this improvement comes from the fact that we do not assume in our approach that each image descriptor is generated from a single object category. By taking into consideration the two-fold uncertainty in our GM model, *GEMP* is able to improve the discriminative power of the created visual words.

- Finally, we can note that like *SDLM*, *GEMP* is sparse in terms of visual words required to reach the optimal performance. For the *PASCAL-VOC-2006* dataset, only 400 visual words are sufficient for *GEMP* and *SDLM* while 1,000 clusters are necessary for *SA*, and 2,000 for *K-means*. The same remark can be made for the *Caltech-10* dataset for which *SDLM* and *GEMP* need a small number of visual words ($I = 200$) to outperform the other methods. Through this classification in terms of required visual words to reach the optimal behavior, we can confirm that (i) using supervised information is better ($\gg$) than resorting to a standard K-Means clustering (*i.e. SA* $\gg$ *K-means*), (ii) allowing a soft-assignment is better than a hard assignment of a descriptor to the nearest centroid (*i.e. SDLM* $\gg$ *SA*) and finally (iii) taking into consideration in a GM model the two-fold uncertainty provides better results (*i.e. GEMP* $\gg$ *SDLM*).

To show the behavior of our method on various sizes of visual dictionaries, we report on Figure 5 the results obtained with *GEMP* according to an increasing

Object class	K-Means I=400	SA I=400	SDLM I=200	GEMP I=200
Airplanes	0.8253	0.9212	0.8803	**0.9253**
Bonsai	0.8020	0.7050	0.8649	0.8721
Car	0.7981	0.8012	0.8000	**0.8981**
Chandelier	0.8286	0.8393	0.9310	0.9486
Faces	0.8988	0.9191	0.9772	**0.9989**
Hawksbill	0.7816	0.8885	0.7375	**0.8217**
Ketch	0.7948	0.9085	0.9153	0.9248
Leopards	0.8956	0.9937	0.9157	**0.9957**
Motorbikes	0.8781	0.8441	0.9314	0.9381
Watch	0.8697	0.9272	0.9750	0.9697
Mean AP	0.8373	0.8748	0.8928	**0.9293**

Table 2: Mean average precision evaluated on *Caltech-10* dataset. An average precision in bold font means that *GEMP* significantly outperforms all the other methods using a Student paired-t test with a Type I error of 5%.

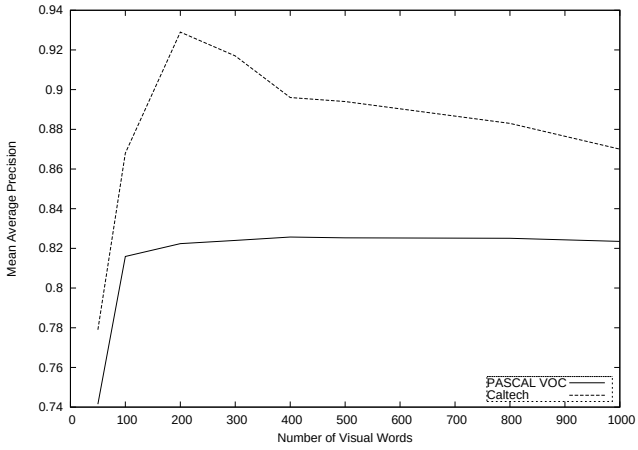

Figure 5: Mean average precision of *GEMP* (using $\alpha = 0.5$) evaluated on *PASCAL VOC-2006* and *Caltech-10* datasets on different dictionary sizes (from 50 to 1,000 visual words).

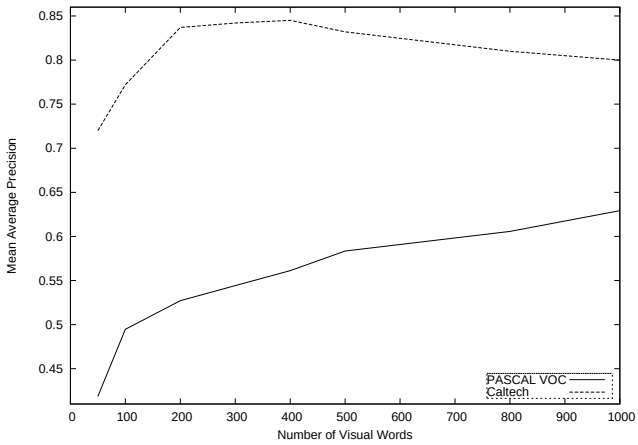

Figure 6: Mean average precision of *K-Means* evaluated on *PASCAL VOC-2006* and *Caltech-10* datasets on different dictionary sizes (from 50 to 1,000 visual words).

number of visual words from 50 to 1,000. We can see that whatever the size (even for very small dictionaries), *GEMP* is very competitive that shows that it is a very interesting sparse method. A comparison with the behavior of *K-Means* (see Figure 6) confirms that the use of supervised information for building dictionaries dramatically improves the quality of the resulting visual words. *K-Means* requires much more visual words (1,000 for *PASCAL-VOC-2006* - in fact it needs 2,000 clusters to reach its optimum- and 400 for *Caltech-10*) to reach rather poor average precisions.

Figures 7 and 8 show for both datasets the mean and the standard deviation of the purity (with $\alpha = 0.5$) w.r.t. an increasing number of visual words (clusters). We can note that the average purity increases showing that clusters generally tend to be specialized for a given class. Interestingly, we can also note that the standard deviation of the purity grows as we increase the number of words. This can be explained by the nature of our objective function which aims at jointly optimizing the likelihood and the entropy. Therefore, this leads to some clusters having a high purity and some others having moderate purity but high likelihood.

In order to give an idea about which kind of words contributes the most to a good performance in image categorization, we estimate the mean average precision with (i) only words of high purity, (ii) only words of low purity and (iii) "intermediate" words representing a compromise between high purity and high likelihood. To achieve this task, we split the optimal set of visual words obtained for *PASCAL VOC-2006* and *Caltech-10* (*i.e.* 400 and 200 words respectively) into three balanced subsets: the most pure words, the less pure words, and the remaining intermediate ones. The results are presented in Figures 9 and 10. *GEMP** is the optimal mean average precision (as already shown in Tables 1 and 2). *GEMP (1/3)* is the mean average precision of our algorithm when only

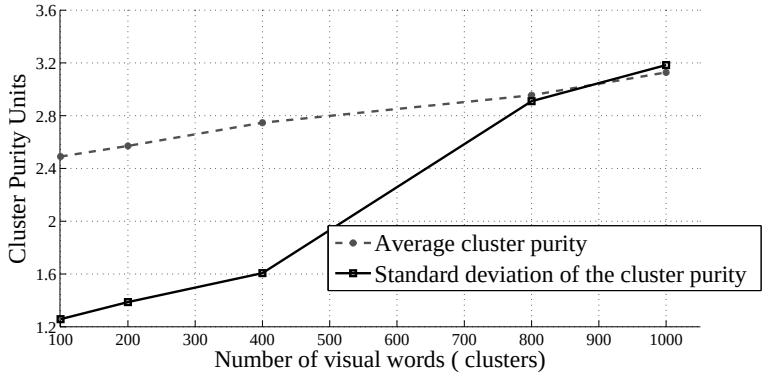

Figure 7: Mean cluster purity and standard deviation of the cluster purity according to an increasing number of visual words for the *PASCAL VOC-2006* dataset.

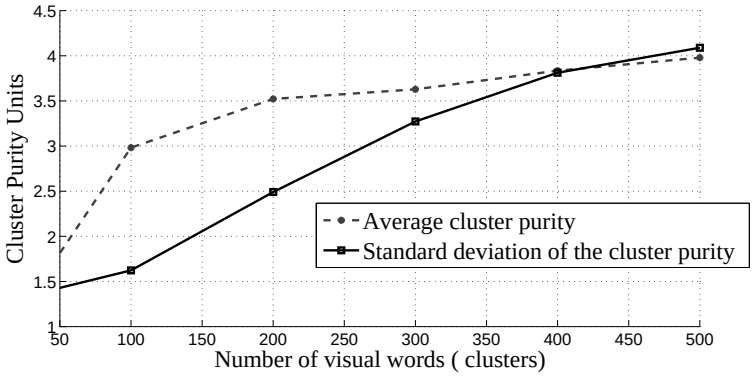

Figure 8: Mean cluster purity and standard deviation of the cluster purity according to an increasing number of visual words for the *Caltech-10* dataset.

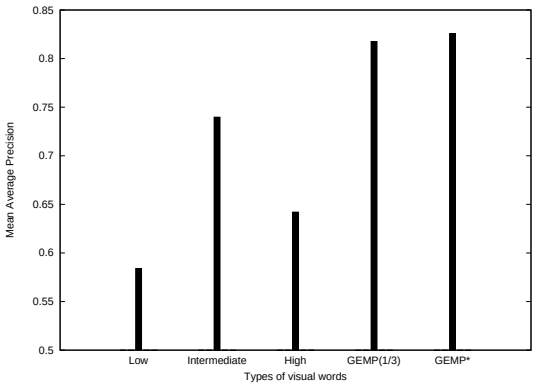

Figure 9: Mean average precision on not pure (Low), moderately pure (Intermediate) and pure (High) visual words for the *PASCAL VOC-2006*. *GEMP (1/3)* and *GEMP** are the results obtained with our algorithm with 133 and 400 visual words respectively.

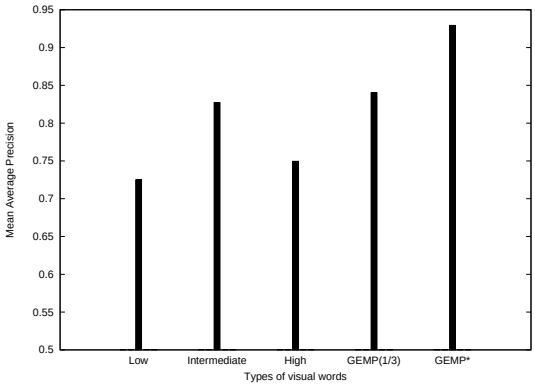

Figure 10: Mean average precision on not pure (Low), moderately pure (Intermediate) and pure (High) visual words for the *Caltech-10* dataset. *GEMP (1/3)* and *GEMP** are the results obtained with our algorithm with 66 and 200 visual words respectively.

a third of the words are used (to be fair with the other results). As expected, using only words of high purity or of low purity is not sufficient to obtain a good precision. Though a good behavior is obtained with the intermediate words, we can see that one needs the three categories to reach the best performance (with the same number of words) that provides an experimental evidence of the interest of our algorithm.

4.5. Effect of the parameter α in GEMP

As we said before, *GEMP* depends on the parameter α which controls the level of supervision in the objective function $J(\Theta)$. If $\alpha = 0$, the algorithm is totally unsupervised and so only optimizes the likelihood. If $\alpha = 1$, the optimization process only aims at building pure clusters. To assess the impact of α

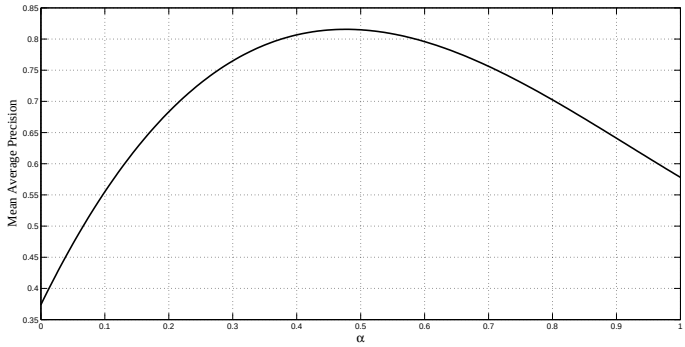

Figure 11: Impact of α on the average precision. Fifteen points were computed and the curve has been interpolated using a 3rd order function.

on the average precision and on the average purity, we perform an experimental study by varying α from 0 to 1 with a constant number of visual words $I = 400$. The following remarks can be made from the results obtained on the *PASCAL-VOC-2006* dataset and presented in Figures 11 and 12 (note that since the same behavior is observed on the *Caltech-10* dataset, we do not report the results):

- As expected, for small values ($\alpha < 0.15$), the level of supervision is not sufficient to take advantage of *GEMP* (see Figure 11). We obtain more or less the same results as *K-Means*. On the other hand, for large values ($\alpha > 0.85$), most of the clusters become purer and purer leading to overfitting phenomena. Therefore, even if in both situations the obtained average precisions are almost the same (smaller than 0.65), this is definitely not due to the same reasons. In the first case, the resulting clusters are too general to be discriminative, while in the second case, the visual words are too specific to allow us to generalize.
  The best results are obtained with middle values ($\alpha \approx 0.5$). Even if we are aware that the optimal α obviously depends on the application we deal with, we can note that a good balance between the two criteria (*i.e.* likelihood and purity) of the objective function seems to be a relevant way to obtain good results.

- From Figure 12, we can note that, as expected, the average purity of the clusters monotonically increases as α raises. But as explained before, a too large purity leads to overfitting, compromising the generalization ability of the resulting clusters.

5. Conclusion and future work

In this paper, we have presented a new supervised Gaussian Mixture model to build discriminative visual dictionaries. Our algorithm, called *GEMP*, aims

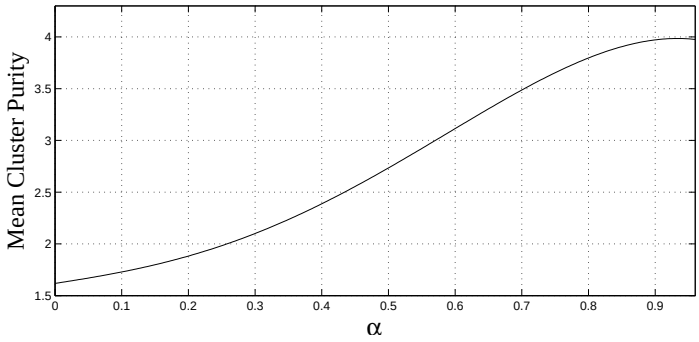

Figure 12: Impact of α on the average purity of the clusters. Nine points were computed and the curve has been interpolated using a 3rd order function.

at maximizing not only the likelihood of a training set of labeled and unlabeled data but also the purity of the clusters. The originality of *GEMP* comes from the fact that it takes into consideration a two-fold uncertainty (cluster and class uncertainty) in the optimization process. We carried out a large experimental study on the *VOC-2006* and *Caltech-101* datasets and compared *GEMP* with three other dictionary creation methods. We provided some experimental evidences that *GEMP* outperforms the state of the art methods, and has the interesting ability to be rather sparse, *i.e.* it only requires a small amount of visual words to reach its optimum in classification.

In our experiments, we have shown that the average precision depends on the parameter α. In this paper, we considered α as a meta-parameter to manually tune. But Figure 11 brings to the fore the need for α to be optimized. This could be achieved by slightly modifying Eq.8 and by generalizing it as follows:

$$J(\Theta, \alpha) = f(\alpha) \times \sum_{x_k \in X} log(p(x_k|\Theta)) + g(\alpha) \times \sum_{i}^{I} log(F(s_i)). \qquad (20)$$

So far, we assumed in our model that a simple relationship is sufficient to model the trade-off between the likelihood of the training data and the purity of the clusters. In this context, we used $f(\alpha) = (1 - \alpha)$ and $g(\alpha) = \alpha$ in Eq.20. While we empirically tried to find a good α, the results suggest that it would be interesting to directly optimize $J(\Theta, \alpha)$ with respect to both θ and α with the EM-algorithm. This will allow us to find not only the optimal Gaussian Mixture parameters but also the best possible parameter α via an iterative learning process. The interesting challenge we plan to deal with is to define relevant functions for $f(\alpha)$ and $g(\alpha)$.

Another interesting point which would deserve further investigation is related to the optimal number of clusters (as suggested by Figure 5). In *GEMP*, we have to predetermine this parameter. While this question is still an open

problem in unsupervised learning, we think that our specific probabilistic model provides a good framework to find a relevant solution. One possible way would consist in using a greedy approach starting at time $t = 1$ from a small number of clusters $I_{t=1} \geq 1$ and splitting step by step during the iterative process the one with the worst contribution to the objective function $J(\Theta)$. But this solution raises another problem related to the complexity cost and requires algorithmic improvements.

A usual way to select key points from the images consists in using a Harris Laplace detector (as we did in this paper) or a dense sampling. We think that our new GM model could be very useful to select the most relevant key points of a new image to classify w.r.t. their probabilities and so reduce the algorithmic complexity of the classification algorithm.

Finally, a current trend in image categorization is to exploit color information to improve the classification accuracy. However, color features are not always relevant and highly depend on the considered application. Our objective is to use our performing dictionary construction method from two different descriptor spaces respectively built from color and shape features, merge the resulting dictionaries and then learn by a Logistic Regression model about the relevance of each visual words. By this way, our aim is to be able to weight color features and estimate their discriminative power to learn some specific classes of objects.

Appendix A: Derivation of $\hat{p}(c_j|s_i)$

Eq.7 shows the way to compute $\hat{p}(c_j|s_i)$. This expression is obtained by using the marginal distribution expansion of $\hat{p}(c_j|s_i)$ w.r.t. all possible samples $x \in X$. The probability $p(c_j|s_i)$ is obtained as follows:

$$p(c_j|s_i) \quad = \quad \frac{p(c_j, s_i)}{p(s_i)}.$$

Using the training set X to estimate $p(c_j|s_i)$, we get:

$$\hat{p}(c_j|s_i) \quad \propto \quad \frac{\sum_k \hat{p}(c_j, s_i, x_k)}{\hat{p}(s_i)}. \tag{21}$$

Assuming c_j is conditionally independent ($\perp$) of s_i given x_k and using the property $c_j \perp s_i | x_k \Leftrightarrow \hat{p}(c_j|s_i, x_k) = \hat{p}(c_j|x_k)$, we get

$$\hat{p}(c_j, s_i, x_k) = \hat{p}(c_j|x_k) \times \hat{p}(s_i, x_k).$$

Since we know that $\hat{p}(c_j|x_k) \propto \hat{p}(x_k|c_j)\hat{p}(c_j)$ and $\hat{p}(x_k|c_j)\hat{p}(c_j) = \hat{p}(x_k, c_j)$, we get $\hat{p}(c_j, s_i, x_k) \propto \hat{p}(x_k, c_j) \times \hat{p}(s_i, x_k)$.

So, we can assume the conditional independence between $\hat{p}(c_j, x_k)$ and $\hat{p}(s_i, x_k)$ and rewrite Eq.(21) as follows:

$$\begin{aligned}
\hat{p}(c_j|s_i) &\propto \frac{\sum_k \hat{p}(c_j, x_k) \times \hat{p}(s_i, x_k)}{\hat{p}(s_i)} \\
\hat{p}(c_j|s_i) &\propto \frac{\sum_k \hat{p}(x_k|c_j) \times \hat{p}(c_j) \times \hat{p}(x_k|s_i) \times \hat{p}(s_i)}{\hat{p}(s_i)} \\
\hat{p}(c_j|s_i) &\propto \sum_k \hat{p}(x_k|c_j) \times \hat{p}(c_j) \times \hat{p}(x_k|s_i).
\end{aligned}$$

In order to satisfy the constraint of a probability distribution, let us normalize the previous expression as follows:

$$\hat{p}(c_j|s_i) = \frac{\sum_k \hat{p}(x_k|c_j) \times \hat{p}(c_j) \times \hat{p}(x_k|s_i)}{\sum_t \sum_k \hat{p}(x_k|c_t) \times \hat{p}(c_t) \times \hat{p}(x_k|s_i)}$$

Assuming that the class prior probability $\hat{p}(c_j) = \frac{1}{R}, \forall j = 1 \dots R$, we get:

$$\hat{p}(c_j|s_i) = \frac{\frac{1}{|C|} \times \sum_k \hat{p}(x_k|c_j) \times \hat{p}(x_k|s_i)}{\frac{1}{|C|} \times \sum_t \sum_k \hat{p}(x_k|c_t) \times \hat{p}(x_k|s_i)}.$$

Therefore, we get:

$$\hat{p}(c_j|s_i) = \frac{\sum_k \hat{p}(x_k|c_j) \times \hat{p}(x_k|s_i)}{\sum_t \sum_k \hat{p}(x_k|c_t) \times \hat{p}(x_k|s_i)},$$

and so Eq.7 holds.

Appendix B: Derivation of $\frac{\partial \widetilde{J}}{\partial \theta_i}$

Derivation of $\frac{\partial \widetilde{J}}{\partial \theta_i}$ can be performed in two steps, first by computing the partial derivative $\frac{\partial \widetilde{J}}{\partial_{(\mu_i, \sum_i)}}$ where the term $\lambda(1 - \sum_i w_i)$ disappears as a constant, and then by calculating $\frac{\partial \widetilde{J}}{\partial_{w_i}}$.

Derivation of $\frac{\partial \widetilde{J}}{\partial_{(\mu_i, \sum_i)}}$

$$\begin{aligned}
\frac{\partial \widetilde{J}}{\partial_{(\mu_i, \sum_i)}} &= \frac{\partial}{\partial_{(\mu_i, \sum_i)}} [(1-\alpha) \sum_k log(\hat{p}(x_k|\Theta)) + \alpha \sum_i^I log(F(s_i))] \\
&= (1-\alpha) \sum_k \frac{w_i \frac{\partial \hat{p}(x_k|s_i)}{\partial(\mu_i, \sum_i)}}{\sum_j^I w_j \hat{p}(x_k|s_j)} + \alpha \frac{F^{'}(s_i)}{F(s_i)}.
\end{aligned}$$

Using the property holding for any function G and stating that $G'(x) = G(x) \times \frac{\partial}{\partial x} log(G(x))$, we get:

$$\frac{\partial \widetilde{J}}{\partial_{(\mu_i, \sum_i)}} = (1-\alpha) \sum_k \hat{p}(s_i|x_k) \frac{\partial}{\partial(\mu_i, \sum_i)} log(\hat{p}(x_k|s_i)) + \alpha \frac{F^{'}(s_i)}{F(s_i)}, \quad (22)$$

where $F^{'}(s_i)$ is the partial derivative of $F(s_i)$ with respect to $(\mu_i, \sum_i)$.

$$F(s_i|\theta_i) = -log(-\sum_j \hat{p}(c_j|s_i) \times log(\hat{p}(c_j|s_i))) + \phi$$
$$F^{'}(s_i) = \frac{-\sum_j (1 + log(\hat{p}(c_j|s_i)) \frac{\partial}{\partial(\mu_i, \sum_i)} \hat{p}(c_j|s_i)}{\sum_j \hat{p}(c_j|s_i) \times log(\hat{p}(c_j|s_i))}$$

Let us simplify the notations by changing $(1 + log(\hat{p}(c_j|s_i))$ by a_i^j. We get,

$$\begin{aligned}
F^{'}(s_i) &= \frac{-\sum_j [a_i^j \times \sum_k \frac{\partial}{\partial(\mu_i, \sum_i)} \hat{p}(x_k|c_j) \times \hat{p}(x_k|s_i)]}{\sum_j \hat{p}(c_j|s_i) \times log(\hat{p}(c_j|s_i)) \times \sum_t \sum_k \hat{p}(x_k|c_t) \hat{p}(x_k|s_i)}, \\
&= \frac{-\sum_j [a_i^j \times \hat{p}(x_k|c_j) \times \sum_k \frac{\partial}{\partial(\mu_i, \sum_i)} \hat{p}(x_k|s_i)]}{\sum_j \hat{p}(c_j|s_i) \times log(\hat{p}(c_j|s_i)) \times \sum_t \sum_k \hat{p}(x_k|c_t) \hat{p}(x_k|s_i)}, \\
&= \frac{-\sum_k \sum_j [a_i^j \times \hat{p}(x_k|c_j) \times \frac{\partial}{\partial(\mu_i, \sum_i)} \hat{p}(x_k|s_i)]}{\sum_j \hat{p}(c_j|s_i) \times log(\hat{p}(c_j|s_i)) \times \sum_t \sum_k \hat{p}(x_k|c_t) \hat{p}(x_k|s_i)}, \\
&= \frac{-\sum_k \sum_j [a_i^j \times \hat{p}(x_k|c_j) \times \hat{p}(x_k|s_i)] \times \frac{\partial}{\partial(\mu_i, \sum_i)} log(\hat{p}(x_k|s_i))}{\sum_j \hat{p}(c_j|s_i) \times log(\hat{p}(c_j|s_i)) \times \sum_t \sum_k \hat{p}(x_k|c_t) \hat{p}(x_k|s_i)}.
\end{aligned}$$

Substituting this expression in Eq.22, we get:

$$\frac{\partial \widetilde{J}}{\partial_{(\mu_i, \sum_i)}} = (1-\alpha) \sum_k [\hat{p}(s_i|x_k)] \times \frac{\partial}{\partial(\mu_i, \sum_i)} log(\hat{p}(x_k|s_i)) +$$
$$\alpha B_i \sum_k \sum_j [a_i^j \times \hat{p}(x_k|c_j) \times \hat{p}(x_k|s_i)] \times \frac{\partial}{\partial(\mu_i, \sum_i)} log(\hat{p}(x_k|s_i))$$

where

$$B_i = \frac{-1}{F(s_i)[\sum_j \hat{p}(c_j|s_i) \times log(\hat{p}(c_j|s_i))] \times \sum_t \sum_k \hat{p}(x_k|c_t) \times \hat{p}(x_k|s_i)}$$

$$\frac{\partial \widetilde{J}}{\partial_{(\mu_i,\sum_i)}} = \sum_k \{(1-\alpha)\hat{p}(s_i|x_k) + \alpha B_i \sum_j a_i^j \times \hat{p}(x_k|c_j) \times \hat{p}(x_k|s_i)\} \times \frac{\partial}{\partial(\mu_i,\sum_i)} log(\hat{p}(x_k|s_i))$$

Derivation of $\frac{\partial \widetilde{J}}{\partial_{w_i}}$

$$\frac{\partial \widetilde{J}}{\partial_{w_i}} = (1-\alpha)\sum_k \frac{\hat{p}(x_k|s_i)}{\sum_t \hat{p}(x_k|s_t)w_t} - \lambda$$

By setting $\frac{\partial \widetilde{J}}{\partial_{w_i}} = 0$, we get

$$\begin{aligned}
(1-\alpha)\sum_k \frac{\hat{p}(x_k|s_i)}{\sum_t \hat{p}(x_k|s_t)w_t} - \lambda = 0 \\
\implies (1-\alpha)\sum_k \frac{\hat{p}(x_k|s_i)w_i}{\sum_t \hat{p}(x_k|s_t)w_t} - \lambda \times w_i = 0 \\
\implies (1-\alpha)\sum_k \hat{p}(s_i|x_k) - \lambda \times w_i = 0 \\
\implies (1-\alpha)\sum_i \sum_k \hat{p}(s_i|x_k) - \sum_i \lambda \times w_i = 0 \\
\implies (1-\alpha)n - \sum_i \lambda \times w_i = 0 \\
\implies \lambda = (1-\alpha) \times n \\
\implies w_i = \frac{\sum_k \hat{p}(s_i|x_k)}{n}.
\end{aligned}$$

References

[1] J. Yang, Y.-G. Jiang, A. G. Hauptmann, C.-W. Ngo, Evaluating bag-of-visual-words representations in scene classification, in: Proceedings of the International workshop on multimedia information retrieval, MIR '07, ACM, New York, NY, USA, 2007, pp. 197–206.

[2] L. Fei-Fei, P. Perona, A bayesian hierarchical model for learning natural scene categories, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2005), Vol. 2, 2005, pp. 524–531.

[3] G. Qiu, Indexing chromatic and achromatic patterns for content-based colour image retrieval, Pattern Recognition 35 (8) (2002) 1675 – 1686.

[4] K. E. van de Sande, T. Gevers, C. G. Snoek, Evaluating color descriptors for object and scene recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (2010) 1582–1596.

[5] R. Cavet, S. Volmer, E. Leopold, J. Kindermann, G. Paaß, Revealing the connoted visual code: a new approach to video classification, Computers & Graphics 28 (3) (2004) 361 – 369.

[6] F. Jurie, B. Triggs, Creating efficient codebooks for visual recognition, in: Proceedings of the International Conference on Computer Vision(ICCV 2005), Vol. 1, 2005, pp. 604 – 610.

[7] D. Nister, H. Stewenius, Scalable recognition with a vocabulary tree, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR 2006), Vol. 2, 2006, pp. 2161–2168.

[8] B. Leibe, K. Mikolajczyk, B. Schiele, Efficient clustering and matching for object class recognition, in: British Machine Vision Conference (BMVC'06), 2006.

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

[10] T. Tuytelaars, C. Schmid, Vector quantizing feature space with a regular lattice, in: Proceedings of the International Conference on Computer Vision(ICCV 2007), 2007, pp. 1–8.

[11] F. Perronnin, Universal and adapted vocabularies for generic visual categorization, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (7) (2008) 1243–1256.

[12] F. Perronnin, C. Dance, G. Csurka, M. Bressan, Adapted vocabularies for generic visual categorization, in: A. Leonardis, H. Bischof, A. Pinz (Eds.), Proceedings of the European Conference on Computer Vision (ECCV 2006), Vol. 3954 of Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2006, pp. 464–475.

[13] L. Yang, R. Jin, R. Sukthankar, F. Jurie, Unifying discriminative visual codebook generation with classifier training for object category recognition, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR 2008) (2008) 1–8.

[14] Q. Jianzhao, Y. Nelson, H. C., Scene categorization with multiscale category specific visual words, Proceedings of SPIE 7252, 72520N.

[15] W. Zhang, A. Surve, X. Fern, T. Dietterich, Learning non-redundant codebooks for classifying complex objects, in: Proceedings of the International Conference on Machine Learning (ICML 2009), ACM, New York, NY, USA, 2009, pp. 1241–1248.

[16] X.-C. Lian, Z. Li, C. Wang, B.-L. Lu, L. Zhang, Probabilistic models for supervised dictionary learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR 2010), 2010, pp. 2305 –2312.

[17] J. Mairal, F. Bach, J. Ponce, G. Sapiro, A. Zisserman, Supervised dictionary learning, in: D. Koller, D. Schuurmans, Y. Bengio, L. Bottou (Eds.), Advances in Neural Information Processing Systems (NIPS 2009), 2009, pp. 1033–1040.

[18] S. Lazebnik, M. Raginsky, Supervised learning of quantizer codebooks by information loss minimization, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (7) (2009) 1294 –1309.

[19] D. G. Lowe, Object recognition from local scale-invariant features, in: Proceedings of the International Conference on Computer Vision(ICCV 1999), Vol. 2, IEEE Computer Society, 1999, pp. 1150–1157 vol.2.

[20] F. Moosmann, B. Triggs, F. Jurie, Fast discriminative visual codebooks using randomized clustering forests, in: B. Schölkopf, J. Platt, T. Hoffman (Eds.), Advances in Neural Information Processing Systems (NIPS 2007), MIT Press, Cambridge, MA, 2007, pp. 985–992.

[21] B. Fernando, E. Fromont, D. Muselet, M. Sebban, Accurate Visual Word Construction using a Supervised Approach, in: Proceedings of the International Conference of Image and Vision Computing New Zealand (ICVNZ 2010), New Zealand, 2010.

[22] J. Ma, W. Gao, A fast globally supervised learning algorithm for gaussian mixture models, in: H. Lu, A. Zhou (Eds.), Web-Age Information Management, Vol. 1846 of Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2000, pp. 449–454.

[23] J. Huang, M. Johnson, Semi-supervised training of gaussian mixture models by conditional entropy minimization, in: Proceedings of Interspeech, NSF 0703624, 2010.

[24] M. andnez Uso, F. Pla, J. M. Sotoca, A semi-supervised gaussian mixture model for image segmentation, in: Proceedings of the International Conference on Pattern Recognition (ICPR 2010), 2010, pp. 2941 –2944.

[25] J. Ma, W. Gao, The supervised learning gaussian mixture model, Computer Science and Technology 13 (1998) 4471–474.

[26] J. Farquhar, S. Szedmak, H. Meng, J. Shawe-Taylor, Improving bag-of-keypoints image categorisation: Generative models and pdf-kernels, Technical report, University of Southampton,.

[27] J. Winn, A. Criminisi, T. Minka, Object categorization by learned universal visual dictionary, in: Proceedings of the International Conference on Computer Vision(ICCV 2005), Vol. 2, 2005, pp. 1800–1807.

[28] M. Everingham, A. Zisserman, C. K. I. Williams, L. Van Gool, The PASCAL Visual Object Classes Challenge 2006 (VOC2006) Results, http://www.pascal-network.org/challenges/VOC/voc2006/results.pdf.

[29] R. F. L. Fei-Fei, P. Perona, Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories, in: IEEE. CVPR 2004, Workshop on Generative-Model Based Vision., 2004.

[30] D. Ormoneit, V. Tresp, Averaging, maximum penalized likelihood and bayesian estimation for improving gaussian mixture probability density estimates, IEEE Transactions on Neural Networks 9 (4) (1998) 639 –650.

[31] A. P. Dempster, N. M. Laird, D. B. Rubin, Maximum Likelihood from Incomplete Data via the EM Algorithm, Royal Statistical Society 39 (1) (1977) 1–38.

[32] K. Mikolajczyk, C. Schmid, Scale & affine invariant interest point detectors, Computer Vision 60 (2004) 63–86.

[33] Color descriptor software from university of amsterdam.
URL `http://staff.science.uva.nl/ ksande/research/colordescriptors/`

[34] L. Breiman, Random forests, Machine Learning 45(1) (2001) 5–32.

[35] M. Zhu, Recall, precision and average precision, Technical report, University of Waterloo,.