Modeling Spatial Layout with Fisher Vectors for Image Categorization

Josip Krapac[†] Jakob Verbeek[†] Frédéric Jurie[††]
[†] LEAR team, INRIA Grenoble Rhône-Alpes, France [††]GREYC, Université de Caen, France
firstname.lastname@inria.fr firstname.lastname@unicaen.fr

Abstract

We introduce an extension of bag-of-words image representations to encode spatial layout. Using the Fisher kernel framework we derive a representation that encodes the spatial mean and the variance of image regions associated with visual words. We extend this representation by using a Gaussian mixture model to encode spatial layout, and show that this model is related to a soft-assign version of the spatial pyramid representation. We also combine our representation of spatial layout with the use of Fisher kernels to encode the appearance of local features. Through an extensive experimental evaluation, we show that our representation yields state-of-the-art image categorization results, while being more compact than spatial pyramid representations. In particular, using Fisher kernels to encode both appearance and spatial layout results in an image representation that is computationally efficient, compact, and yields excellent performance while using linear classifiers.

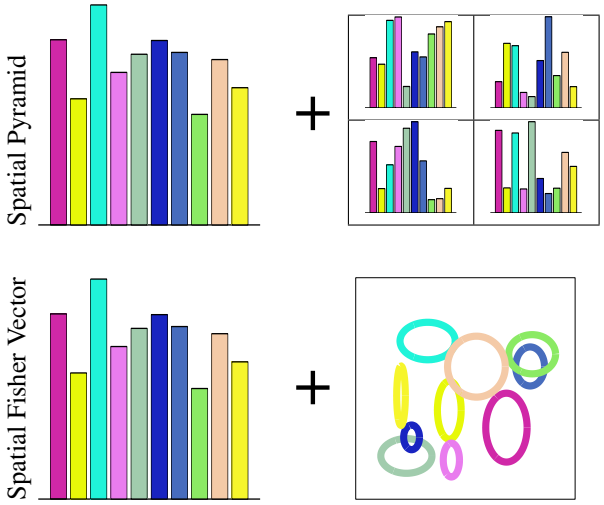

Figure 1. The spatial pyramid image representation concatenates visual word histograms of the complete image and spatial cells. Our spatial Fisher vector representation models spatial layout by the mean and variance of the occurrences of each visual word.

1. Introduction

Image categorization aims to determine the presence of objects in images, or to recognize them as particular scene types such as *city*, *mountain*, or *beach*. Current state-of-the-art image categorization systems use bag-of-word image representations. This approach represents the image content by global statistics of the appearance of local image regions. First, image regions are sampled from the image, either using a regular grid, in a randomized manner , or using interest point detectors. Each region is then described using a feature vector, *e.g.* SIFT or color histograms. A visual vocabulary is then learned using k-means or a mixture of Gaussians (MoG). The visual vocabulary quantizes the feature space into different cells, and region features are assigned to these cells: either using hard-assignment for k-means, or using soft-assigment for a MoG model. The assignments are then aggregated over whole image to obtain an image representation: a histogram with as many bins as visual words, where each bin gives the number of regions assigned to a visual word. In this way the image represented by a set of regions is embedded into vector space in which an image classification model is learned.

Several extensions to the basic bag-of-words image representation have been proposed; we will discuss the most relevant ones in detail in the next section. One recent extension to the bag-of-words model is the Fisher kernel image representation [17]. Instead of only storing the average (soft-)assign of patches to visual words, the first and second order moments of the patches assigned to each visual word are also stored. This means that, for a descriptor of size D and K visual words, the image representation is of size $K(1 + 2D)$. Since more information is stored per visual word, a smaller number of visual words can be used for a given level of categorization performance, which is computationally more efficient.

Another extension is the spatial pyramid representation of [10] which captures the information about the spatial lay-

out of the image by computing bag-of-word histograms over different regions of the image, and concatenating these to form the final representation. Using K visual words and C spatial cells results in image representation of size KC. The same idea applied to the Fisher kernel image representation [19], leads to a representation of size $KC(1+2D)$. This representation has been proven to be effective, in particular when the image categories exhibit characteristic layouts, as in the case of the scene recognition. For object categorization this idea is also effective because even though the objects may appear anywhere in the image, the scenes in which they appear may still have strong layout patterns.

We propose an alternative encoding of spatial layout information, based on the Fisher kernel principle [8], which was previously only used to encode the appearance information [17]. We model the spatial location of the image regions assigned to visual words using MoG models, and compute the Fisher kernel representation for these models. See Figure 1 for a schematic comparison of our approach to spatial pyramids. We explore variants of our image representation experimentally, using the 15-Scenes and PASCAL VOC 2007 data sets. Compared to using spatial pyramids, we obtain representations that are smaller, while not degrading performance. Using bag-of-word for appearance, our representations are smaller and achieve better performance using linear classifiers, as compared to using the spatial pyramid representation with the non-linear intersection kernel. Using Fisher kernels for appearance, our representation achieves similar performance as spatial pyramids, but have the advantage that it is significantly more compact.

In the following section we discuss the most relevant related work, and then present our image representations in Section 3. We present extensive experimental results in Section 4, comparing different variants of our image representations to alternatives from the literature. Finally, we present our conclusions in Section 5.

2. Related work

Because of its effectiveness, the bag-of-words (BOW) model has become one of the most popular representations for image categorization since its introduction in the seminal papers [6, 25]. Subsequent research has focused on overcoming its two intrinsic limitations, namely (a) the computational cost of the assignment of local features to visual words, and (b) the lack of information on the spatial layout of the local features.

Quantization issues and codebook compactness. Performance of the BOW model is often reported to increase with the size of the dictionary [6, 25, 26] and the number of regions sampled from images [16]. Typically, vocabularies of several thousands codewords are used, and thousands of regions are densely sampled from the images. Assigning local features to their nearest visual word is computationally expensive, as it scales as the product of the number of visual words, the number of regions, and the local feature dimensionality. These issues have been addressed by different authors, *e.g*. [15] proposed a hierarchical k-means framework scaling logarithmically with the number of codewords, while [20] introduced an approximate k-means algorithm better suited to the use of large vocabularies. Random forests, because of their hierarchical structure, are also good candidates for handling large visual vocabularies [4, 13].

Nevertheless, the simplest way to reduce the time spent in assigning features to visual words is certainly to make the vocabulary smaller, of course without losing performance. Different authors have tried to build compact discriminative vocabularies [12, 13, 32], *i.e*. vocabularies that are specialized in representing the differences between categories. One of the most convincing approaches is the one by Perronnin et al. [18]. However, these vocabularies are not universal since they have to be rebuilt each time a new category is added, which is a severe drawback.

Additionally, when the vocabularies are more compact, the information lost in the quantization process becomes more important, in particular when using hard assigment [26]. The amount of discriminative information is considerably reduced due to the rough quantization of the feature space, as clearly shown by [3] who propose to compute direct image-to-class distances without descriptor quantization. The loss of information can be compensated by assigning descriptors to multiple visual words, as suggested by [21, 26, 27]. The assignment can also be guided by sparsity constraints [30] or locality constraints [28]. However, these approaches again require large codebooks, *e.g*. 2048 visual words in [28].

Regarding the production of compact vocabularies, an appealing approach is the one proposed in [17]. They have suggested to use the Fisher kernel framework [8], whose high dimensional gradient representation contains more information than a simple histogram representation, resulting in informative representations using compact vocabularies.

Spatial information. The BOW representation is a frequency histogram of quantized local appearances, and the spatial layout of the appearances is completely ignored. Clearly, the spatial information may convey useful cues for image categorization, and at least two different ways to encode spatial information have been explored: based on pairwise positions of features, and using absolute positions.

Considering pairs of spatially close image regions is probably the most intuitive way to incorporate spatial information. Visual word "bigrams" are considered in [23], by forming a bag-of-word representation over spatially neighboring image regions. Others have proposed a more effi-

cient feature selection method based on boosting which progressively mines higher-order spatial features [11], and [14] proposes joint feature space clustering to build a compact local pairwise codebook. Distinctive spatial configurations of visual words can also be discovered by data mining techniques, such as frequent itemsets [22].

In addition to pairwise relationships, images often have global spatial biases: the composition of the pictures of particular object or scene category typically share common layout properties. Therefore, exploiting the global positions of the features in the image is effective in many cases. Spatial Pyramid Matching (SPM) [10] exploits this property by partitioning the image into increasingly finer cells and concatenating the BOW histograms of the cells. This strategy is used in most of the state-of-the-art approaches, see *e.g.* [19, 31]. In [5] SPM is further improved by learning a weighting of the levels of the SPM representation on a validation set. The idea of implicitly representing spatial information by weighting image cells based on their discriminative information was explored earlier in the context of facial expression recognition in [24], where linear discriminant analysis was used to find a weighting of the spatial cells. In addition to global spatial information, they also used local auto-correlation measures to include local spatial information. Recently, a similar strategy was applied to address image categorization in [7], which yielded results comparable to the state-of-the-art on the 15-Scenes data set.

More closely related to our work, [33] models regions appearances with a mixture of Gaussian (MoG) density, and uses the posterior over visual words for the image regions to form so called "Gaussian maps". Then then apply SPM to encode the spatial occurrence of visual words in the image. Our approach is similar, as we also use a MoG to model the region appearances and also incorporate spatial layout based on coding the region locations of each visual word. However, different from their approach, we use the more efficient Fisher kernel [8, 17] approach to jointly code appearance and spatial layout, giving efficient, compact, and discriminative image representations. Our work is also related to [1] which employed first and second order spatial moments associated with bins of color histograms to derive an improved representation for mean-shift tracking.

3. Fisher kernels to encode spatial layout

In this section we present our models to encode both the spatial layout of local image features and their visual appearance. In Section 3.1 we start by reinterpreting the bag-of-words (BOW) image representation as a Fisher vector representation for a simple multinomial probabilistic model. We then extend this model in Section 3.2 by including a Gaussian location model, and further extend the spatial model to a mixture of Gaussians (MoG) in Section 3.3. We integrate our spatial models with MoG appearance models in Section 3.4, combining Fisher vector representations for both appearance and spatial layout. Finally, we consider normalization of the Fisher vectors in Section 3.5, and we compare the models we introduced to spatial pyramid image representations in Section 3.6. The derivations of equations can be found in appendix of [9].

3.1. A generative model view on bag-of-words

The BOW image representation uses k-means to quantize the space of patch appearances, for each patch $\boldsymbol{x}_n$ we use $w_n \in \{1, \ldots, K\}$ to denote the index of the k-means center that is closest to $\boldsymbol{x}_n$ among the K centers. The trivial probabilistic model over the quantization indices is just a multinomial $\boldsymbol{\pi}$, and the likelihood of observing the k-th quantization index is given by $p(w_n = k) = \pi_k$. The parameters of this multinomial are fitted from the data used to learn the k-means quantizer, and are simply given by the fraction of the patches assigned to each visual word.

To apply the Fisher kernel framework [8], we consider the average log-likelihood of the N patches in an image, given by

$$\mathcal{L} = \frac{1}{N} \sum_{n=1}^{N} \ln p(w_n), \tag{1}$$

where the average is taken to achieve invariance w.r.t. the number of patches N in the image. We parameterize the multinomial using a softmax by defining $\pi_k = \exp \alpha_k / \sum_j \exp \alpha_j$, which by construction satisfies the constraints $\pi_k \geq 0$, and $\sum \pi_k = 1$ for any setting of the α_k. The gradient is then given by

$$\frac{\partial \mathcal{L}}{\partial \alpha_k} = h_k - \pi_k, \tag{2}$$

where h_k is the frequency of the k-th visual word in the image, *i.e.* its count divided by N.

We recognize the gradient of dimension K as the standard bag-of-word historgam minus the multinomial learned from the vocabulary training data.

3.2. A simple Gaussian spatial model

We extend the appearance-only bag-of-words model by introducing a Gaussian location model per visual word. Each image patch is represented as the tuple $\boldsymbol{f} = (w, \boldsymbol{l})$, where w is the quantization index and $\boldsymbol{l}$ gives the spatial location of the patch in the image. We define a generative model over appearance-location tuples as

$$p(\boldsymbol{f}) = p(w)p(\boldsymbol{l}|w), \tag{3}$$

$$p(w = k) = \pi_k, \tag{4}$$

$$p(\boldsymbol{l}|w = k) = \mathcal{N}(\boldsymbol{l}; \boldsymbol{m}_k, \boldsymbol{S}_k), \tag{5}$$

where $\mathcal{N}(\cdot; \boldsymbol{m}_k, \boldsymbol{S}_k)$ denotes the Gaussian location model with mean $\boldsymbol{m}_k$ and covariance matrix $\boldsymbol{S}_k$ associated with

the k-th visual word. The location models can be learned trivially by computing the mean and variance of the spatial coordinates of image patches assigned to the k-th visual word in the vocabulary training data.

Using diagonal covariance matrices, the gradient of the log-likelihood of a patch $\boldsymbol{f}_n$ is

$$\frac{\partial \ln p(\boldsymbol{f}_n)}{\partial \alpha_k} = q_{nk} - \pi_k, \tag{6}$$

$$\frac{\partial \ln p(\boldsymbol{f}_n)}{\partial \boldsymbol{m}_k} = q_{nk} \boldsymbol{S}_k^{-1} \boldsymbol{l}_{nk}, \tag{7}$$

$$\frac{\partial \ln p(\boldsymbol{f}_n)}{\partial \boldsymbol{S}_k^{-1}} = q_{nk} \left(\boldsymbol{S}_k - \boldsymbol{l}_{nk}^2 \right) / 2, \tag{8}$$

where $q_{nk} = 1$ if $w_n = k$ and $q_{nk} = 0$ otherwise, $\boldsymbol{l}_{nk} = \boldsymbol{l}_n - \boldsymbol{m}_k$, and $\boldsymbol{l}_{nk}^2$ denotes the element-wise square. With slight abuse of notation, the last equation gives the gradient w.r.t. the inverse of the diagonal covariance matrix.

By averaging the gradients over all patches in an image, this yields an image descriptor of size $K(1 + 2d)$, where $d = 2$ is the dimension of the location $\boldsymbol{l}$. For each visual word we have 1 element for the gradient w.r.t. the α_k, and 4 for the gradient w.r.t. the spatial mean $\boldsymbol{m}_k$ and variance $\boldsymbol{S}_k$.

3.3. A spatial mixture of Gaussian model

We extend the spatial model by using an MoG distribution over the patch locations instead of a single Gaussian, *i.e*. we replace Eq. (5) with

$$p(\boldsymbol{l}|w = k) = \sum_{c=1}^{C} \theta_{kc} \mathcal{N}(\boldsymbol{l}; \boldsymbol{m}_{kc}, \boldsymbol{S}_{kc}), \tag{9}$$

using a mixture of C Gaussians to model the spatial locations of the patches per visual word. We define the mixing weights again using the softmax as $\theta_{kc} = \exp \beta_{kc} / \sum_j \exp \beta_{kj}$. The spatial model of each visual word can be learned using the EM algorithm [2] from the patch locations associated with each visual word.

The gradient w.r.t. the α_k remains as in Eq. (6), but for the location model parameters we obtain

$$\frac{\partial \ln p(\boldsymbol{f}_n)}{\partial \beta_{kc}} = q_{nk} \left(r_{nkc} - \theta_{kc} \right), \tag{10}$$

$$\frac{\partial \ln p(\boldsymbol{f}_n)}{\partial \boldsymbol{m}_{kc}} = q_{nk} r_{nkc} \boldsymbol{S}_{kc}^{-1} \boldsymbol{l}_{nkc}, \tag{11}$$

$$\frac{\partial \ln p(\boldsymbol{f}_n)}{\partial \boldsymbol{S}_{kc}^{-1}} = q_{nk} r_{nkc} \left(\boldsymbol{S}_{kc} - \boldsymbol{l}_{nkc}^2 \right) / 2, \tag{12}$$

where $\boldsymbol{l}_{nkc} = \boldsymbol{l}_n - \boldsymbol{m}_{kc}$ and $r_{nkc} = p(c|\boldsymbol{l}_n, w_n = k) = \theta_{kc} \mathcal{N}(\boldsymbol{l}_n; \boldsymbol{m}_{kc}, \boldsymbol{S}_{kc}) / p(\boldsymbol{l}_n | w_n = k)$. The r_{nkc} can be interpreted as a "spatial soft-assign" of patches of visual word k to the spatial mixture components. The image representation has size $K + KC(1 + 2d)$, K dimensions for the appearance part, and $KC(1 + 2d)$ for the spatial layout.

3.4. Mixture of Gaussians appearance models

We now combine the ideas from the previous section with a mixture of Gaussians (MoG) model for the patch appearances, and use Fisher vectors to obtain the image representations. The parameters of the models defined in this section can all be learned using the EM algorithm.

Appearance-only Fisher vector image representation. First, we define the appearance-only model as in [17]; the patch appearances $\boldsymbol{x} \in \mathbb{R}^D$ are modeled as

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k p(\boldsymbol{x}|w = k) \tag{13}$$

$$p(\boldsymbol{x}|w = k) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{14}$$

where π_k denotes the mixing weight of the kth Gaussian in the mixture, defined using the softmax as above. Similarly to the spatial models, for the appearance models we also use diagonal covariance matrices, therefore the appearance representation has size $K(1 + 2D)$.

Redefining q_{nk} to denote the posterior $p(w_n = k|\boldsymbol{x}_n)$, or responsibility, and $\boldsymbol{x}_{nk}$ to denote $\boldsymbol{x}_n - \boldsymbol{\mu}_k$, the gradients of the log-likelihood for a single patch are

$$\frac{\partial \ln p(\boldsymbol{x}_n)}{\partial \alpha_k} = q_{nk} - \pi_k, \tag{15}$$

$$\frac{\partial \ln p(\boldsymbol{x}_n)}{\partial \boldsymbol{\mu}_k} = q_{nk} \boldsymbol{\Sigma}_k^{-1} \boldsymbol{x}_{nk}, \tag{16}$$

$$\frac{\partial \ln p(\boldsymbol{x}_n)}{\partial \boldsymbol{\Sigma}_k^{-1}} = q_{nk} \left(\boldsymbol{\Sigma}_k - \boldsymbol{x}_{nk}^2 \right) / 2. \tag{17}$$

The image representation is obtained by averaging these gradients over all patches in the image. This representation has the computational advantage that we can use smaller number of visual words, since the appearance information per visual word is coded more precisely [17].

Gaussian spatial models with MoG for appearance. When we include a single Gaussian spatial model, the appearance-location tuple $\boldsymbol{f} = (\boldsymbol{x}, \boldsymbol{l})$ is modeled as

$$p(\boldsymbol{f}) = \sum_k \pi_k p(\boldsymbol{x}|w = k) p(\boldsymbol{l}|w = k), \tag{18}$$

where $p(\boldsymbol{l}|w = k)$ is defined as in Eq. (5), and $p(\boldsymbol{x}|w = k)$ as in Eq. (14).

If we redefine $q_{nk} = p(w_n = k|\boldsymbol{x}_n, \boldsymbol{l}_n)$, the gradients with respect to the $\alpha_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$ are the same as in Eq. (15)–Eq. (17), and those for the $\boldsymbol{m}_k, \boldsymbol{S}_k$ are the same as in Eq. (7)–Eq. (8), albeit using the current definition of q_{nk}. The image representation has size $K(1 + 2D + 2d)$ in this case. Note that since the patch descriptor $\boldsymbol{x}$ is generally high dimensional, *e.g*. 128 for SIFT, the additional $2d = 4$ dimensions increase the representation size only slightly as compared to the MoG appearance-only model.

Using MoG spatial models with MoG for appearance. In this case we use the model of Eq. (18), with the MoG spatial model $p(\boldsymbol{l}|w = k)$ of Eq. (9). The model now has $K(1 + 2D)$ parameters for the appearance model, and $KC(1 + 2d)$ for the spatial models. So in total we have $K(1 + 2D) + KC(1 + 2d)$ parameters.

The gradients with respect to the appearance parameters $\alpha_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$ remain as in Eqs. (15)—(17). For the spatial parameters $\beta_{kc}, \boldsymbol{m}_{kc}, \boldsymbol{S}_{kc}$ the gradients are the same as in Eqs. (10)—(12) using the current definition of the q_{nk}.

3.5. Fisher score vector normalization

In order to obtain invariance of the kernel w.r.t. reparametrization of the model, the Fisher kernel framework of [8] requires multiplication of the gradient vectors with $\boldsymbol{F}^{-1/2}$ where $\boldsymbol{F} = \mathbb{E}_{\boldsymbol{x}}\left[\boldsymbol{g}(\boldsymbol{x})\boldsymbol{g}(\boldsymbol{x})^\top\right]$ is the Fisher information matrix and $\boldsymbol{g}(\boldsymbol{x})$ denotes the gradient vector. Since $\boldsymbol{F}$ may be large, computation of $\boldsymbol{F}^{-1/2}$ can be costly, *e.g.* using $K = 500$ visual words, descriptors of size $D = 64$, and $C = 5$ spatial cell in SPM, $\boldsymbol{F}$ is a $322,500 \times 322,500$ matrix. Therefore it is common to use a diagonal approximation of $\boldsymbol{F}$; where [17] uses an analytical approximation, we follow [2] (section 6.2) and use the empirical approximation of the diagonal. Based on the patches used for vocabulary construction, we compute the mean and variance on each dimension of the gradient vectors to obtain an additive and multiplicative normalizer, so that the dimensions of the normalized gradient vectors have zero-mean and unit-variance.

3.6. Discussion and comparison to SPM

We summarize the models we have presented in this section in Table 1, giving the representation size for each of them, and comparing them to the sizes obtained using spatial pyramids (SPM) [10, 19] that concatenate appearance representations obtained over C spatial cells. We use C to denote either the number of components in our spatial MoG, or the total number of cells in the SPM representation.

Comparing SPM to our MoG spatial model in combination with k-means for appearance, we see that our representation adds $2d = 4$ numbers for each visual word (K) and spatial Gaussian (C). The size of the single Gaussian location model with equals that of the SPM model with $C = 5$.

Comparing SPM to our MoG spatial model with MoG appearance models, we see that our model yields a much more compact representation. Where SPM concatenates C appearance Fisher vectors of size $K(1 + 2D)$, our representation uses a single appearance Fisher vector of size $K(1 + 2D)$ and adds KC spatial Fisher vectors of size $(1 + 2d) = 5$. For a typical setting of $K = 200, D = 64, C = 5$, the SPM representation is $129,000$, while our MoG spatial-appearance model yields a descriptor of size $30,800$: more than four times smaller. When using $C = 21$ the sizes would be $541,800$ and $46,800$ respectively, and our descriptor would be more than 11 times smaller.

spatial	appearance k-means	appearance MoG
None	K	$K(1 + 2D)$
Gauss.	$K(1 + 2d)$	$K(1 + 2D + 2d)$
MoG	$K + KC(1 + 2d)$	$K(1 + 2D) + KC(1 + 2d)$
SPM	KC	$KC(1 + 2D)$

Table 1. Comparison of representation size for different models, using k-means or a MoG for appearance, and no spatial model, a single Gaussian, a C-component MoG, or C spatial pyramid cells.

To compute our representation we have to compute the appearance soft-assign, and the spatial soft-assign per visual word. So, the only additional cost is to compute a spatial soft-assign per visual word, which costs $O(KC)$. Since the appearance soft assign has cost $O(KD)$ per patch (regardless of k-means or MoG model), and since the descriptor dimension is typically much larger then the number of spatial cells, *i.e.* $D \gg C$, we can state that in general the computational cost is less than doubled.

4. Experimental evaluation

Feature extraction and vocabulary learning. In all experiments we follow the same feature extraction process. We sample image patches on a regular spatial grid, with step-size half of the patch-size, over 8 scales separated by a factor 1.2. At the finest scale we use a patch-size of 20 and 16 pixels for the 15-Scenes and PASCAL VOC data sets, respectively. We compute SIFT descriptors and reduce the dimensionality from 128 to 64 when using Fisher vectors to code appearance, in order to reduce the image representation, as it is done in [17, 33]. Because of the spatial binning in the SIFT descriptor we expect that the local features are highly correlated, which we decorrelate globally by using PCA, therefore better fitting our modeling assumption that the features are uncorrelated locally, which is assumed by diagonal form of covariance matrices for appearance model components. The k-means and MoG appearance models, as well as PCA subspace, are learned using a random sample of $500,000$ patches from the training images.

Construction of spatial models. Once appearance models are learned we learn the spatial models, either Gaussian or MoG, from the patches assigned to each visual word. However, in initial experiments we found that without loss of performance we can also use a fixed spatial model shared across all visual words ($\boldsymbol{m}_c = \boldsymbol{m}_{kc}, \boldsymbol{S}_c = \boldsymbol{S}_{kc}$). Therefore there is no additional computational cost in training as compared to training just the visual vocabularies using k-means, or EM for MoG. Importantly, we do compute the gradient w.r.t. the spatial models per visual word. Using one spatial Gaussian per visual word, we set the mean and variance to

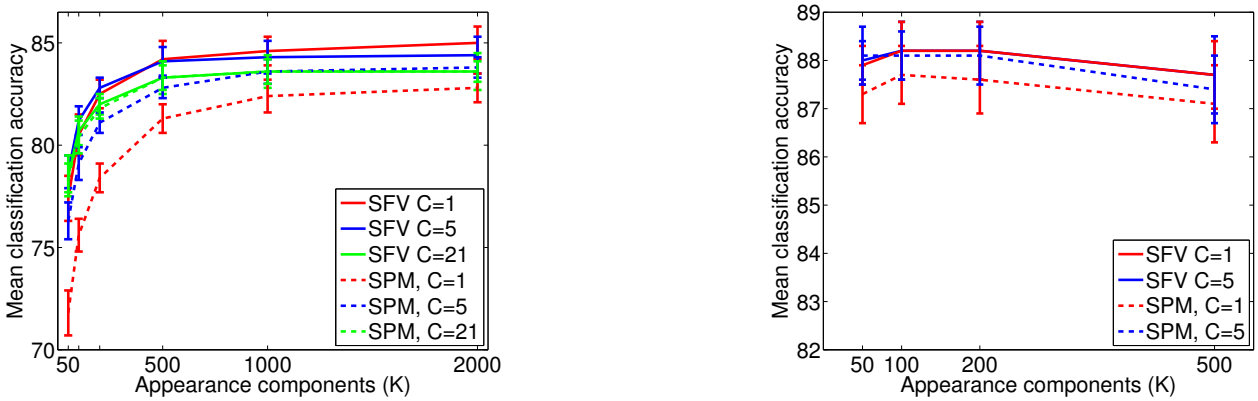

Figure 2. Using 15-Scenes data set to compare Spatial Fisher Vectors (SFV, solid curves) to Spatial Pyramids (SPM, dashed curves) for coding spatial layout, when using bag-of-words for coding appearance (left), and when using Fisher vector for coding appearance (right).

match the first and second order moments of the uniform distribution over the unit square. Using $C = 5$ components we complement the global Gaussian with four Gaussians, each matching the first and second order moments of the four quadrants of the unit square. In a similar manner we add 16 Gaussians when using spatial models with up to $C = 21$ spatial mixture components. Note that the spatial model resembles the structure of the SPM in this case. The main differences are that we also store spatial first and second order moments of the patches assigned to each spatial component, and that we use a spatial soft-assign.

Compared representations. In our experiments we compare the representations summarized in Table 1. We test SPM representations up to three levels; at the first level we have only $C = 1$ global spatial cell which does not encode any spatial information. Using the first two levels we have $C = 5$ spatial cells, and using all three levels we have $C = 21$ spatial cells. When using Fisher vectors for appearance, we do not include $C = 21$ since then the image representation becomes very large, without increasing performance.

Classifier training and evaluation. For all image representations we learn a linear classifier over the Fisher vector representations, and include the L2 and power normalizations of [19]. For a fair comparison, we use the histogram intersection kernel [10] when using BOW+SPM representations, since these seem to be optimal for that representation. For the 15-Scenes data set we learn (kernelized) multi-class logistic discriminant models, and report classification accuracy measured as the fraction of correctly clasified test images. For PASCAL VOC 2007 we learn a binary SVM classifier per class, and report the mean of the per-class average precision (mAP) values.

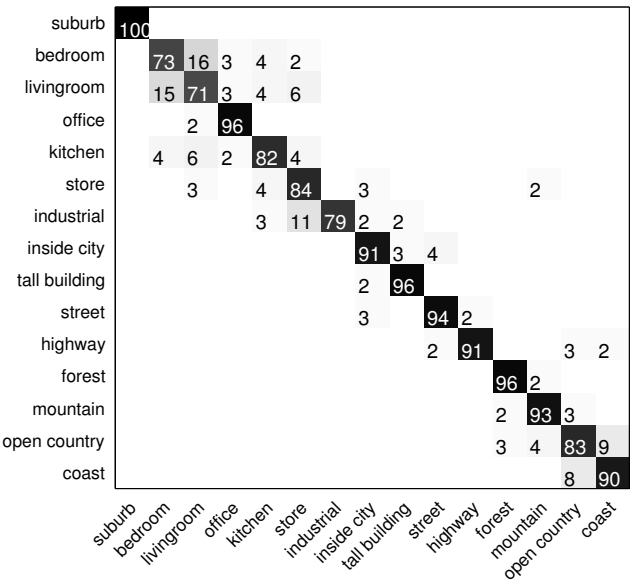

Figure 3. Normalized confusion matrix for 15-Scenes dataset (the rows are the true classes), we only show figures larger than one.

Experimental results for the 15-Scenes dataset. This data set [10] contains 4485 images of 15 scene categories. We use the standard setup for this data set, using 10 random splits of the data into a train set of 100 images per class, and using the rest as test data. We then average the classification accuracy over the test/train splits.

In Figure 2 we show the classification accuracies as a function of the vocabulary size K. Using k-means to encode appearance (left panel) we see that large vocabularies ($K \geq 1000$) yield the best performance, and that our Spatial Fisher Vector representation with $C = 1$ outperforms all others, achieving 85.0 ± 0.8 accuracy. The size of our representation is in this case $K + K2d = 10,000$, which is the same as the size of the best SPM model with $C = 5$

		SPM			SFV	
	K / C	1	5	21	1	5
BOW	50	29.1	37.0	41.4	35.1	37.9
	100	33.8	40.4	44.1	39.8	41.7
	200	38.1	44.1	47.1	43.5	45.1
	500	42.7	47.7	49.9	47.5	48.9
	1000	45.9	50.1	51.5	50.1	50.8
	2000	48.0	51.1	**52.3**	52.3	**52.9**
Fisher vector	50	54.1	55.8		55.4	50.2
	100	55.0	56.5		56.1	55.6
	200	55.5	**56.7**		56.5	56.1
	500	55.5	56.5		**56.6**	56.3

Table 2. PASCAL VOC 2007: comparison of spatial pyramids (SPM) with with C cells (left) to Spatial Fisher vectors (SFV) with C Gaussian components (right) for coding spatial layout. Using bag-of-words (BOW) for coding appearance (top), and using Fisher vector for coding appearance (bottom).

which uses a non-linear kernel an achieves 83.8 ± 0.5. Our SFV results are remarkably good for a bag-of-word image appearance models in combination with linear classifiers.

When using Fisher vectors for appearance (right panel) performance is generally much higher (note difference in scaling on both axes). In this case our Spatial Fisher Vector representation with $C = 1$ and $K = 100$ achieves best performance at $88.2\% \pm 0.6$, which is comparable to using SPM with $C = 5$ cells ($88.1\% \pm 0.5$). Note that our representation is much smaller, $K(1 + 2D + 2d) = 13,300$ dimensions, than using SPM: $KC(1 + 2D) = 64,500$ dimensions. We also noticed that performance saturates or drops when using vocabularies larger than 100 to 200 visual words. This consistent with the observations made by [17].

Our results with only $K = 200$ visual words are on par with the current state-of-the-art of 88.1% reported in [29]. While we only use SIFT descriptors, [29] combines 14 different low-level image features; when using only SIFT [29] reports 81.2% using a BOW+SPM representation and intersection kernels. In Figure 3 we show the confusion matrix we obtain with our best model.

Experimental results for PASCAL VOC 2007. The PASCAL VOC 2007 data set contains 9963 images, annotated for presence of 20 different object categories. We have used the 5011 images in the train and validation sets to train our models, and evaluate them on the 4952 test images.

In Table 2 we show the mAP scores for different vocabulary sizes. When using bag-of-word appearance models (top), we observe that our Spatial Fisher vector representations with $C = 1$ and a linear classifier yield performance comparable to using $C = 5$ cells with SPM and intersection kernel. The best performance of 52.9% is obtained using spatial Fisher vectors with $C = 5$ components, and 2000 visual words. The best SPM results of 52.3% are obtained using $C = 21$ cells, and $K = 2000$. As for the 15-Scenes data set, using Fisher vectors for appearance (bottom) improves the results, to a maximum of 56.6% using SFV with a single Gaussian, and for SPM the best results are 56.7% using $C = 5$ cells. Again, our representation is much smaller, using $K = 200, C = 1$ the SFV has size $K(1 + 2D + 2d) = 26.600$, while using SPM with $K = 200, C = 5$ yields a $KC(1 + 2D) = 129.000$ dimensional image representation.

$K = 50$ $K = 2000$

Figure 4. Ellipsoidal display of the spatial distributions of patches assigned to the five most frequent visual words in an image. As the vocabulary size grows, the ellipses become smaller since fewer patches are assigned to each visual word. In that case, the spatial distribution is succinctly captured by a Gaussian distribution, while many SPM cells are needed to attain the same precision.

Our results are comparable to those in [19], which reports 55.3% using SPM with $C = 1, K = 256$, our results with SPM and $C = 1, K = 200$ are 55.5%. They reported 58.3% using SPM with $C = 8$ cells, which uses the complete image, the four quadrants, and 3 horizontal strips; a configuration which we did not explore here.

Discussion of experimental results. In Figure 4 we visualize the spatial distributions of patches assigned to visual words in a particular image for two vocabulary sizes. As the number of visual words grows, less patches are assigned per visual word, and our spatial Fisher vectors — even with a single spatial component— are able to accurately describe the positions of patches assigned to each visual word. To represent spatial layout with the same accuracy, spatial pyramids would need many spatial cells. However, this would result in very large image representations, that are more prone to overfitting.

This analysis could explain the results in Table 2 and Figure 2 when using BOW appearance models. When using small number of visual words the gain by adding more spatial components to our model is significant, while this gain diminishes as we increase the number of visual words.

5. Discussion and conclusion

We introduced Spatial Fisher Vectors (SFV) as a new method to encode spatial layout for image categorization. In SFV, spatial cells are adapted per word to the patch positions, unlike the rigid structure of spatial pyramid cells. The advantages of our representation are (i) its compactness, and (ii) its good performance with linear classifiers. When using bag-of-words appearance models, our representation with linear classifiers gives similar or better results than SPMs with nonlinear intersection kernel classifiers, for comparable size of the representation. When we combine our model with Fisher vector coding of appearance, we obtain similar or better results compared to SPM, but the image descriptors are roughly four times more compact, reducing requirements on disk storage, memory, and classifier training time by the same factor. In future work we want to further explore the Fisher kernel framework using more advanced generative models to capture the correlations between the appearance and spatial layout of local images features.

Acknowledgements. This work was partly funded by the EU project AXES and the OESO and ANR project Quaero.

References

[1] S. Birchfield and S. Rangarajan. Spatiograms versus histograms for region-based tracking. In *CVPR*, 2005.

[2] C. Bishop. *Pattern recognition and machine learning*. Spinger-Verlag, 2006.

[3] O. Boiman, E. Shechtman, and M. Irani. In defense of nearest-neighbor based image classification. In *CVPR*, 2008.

[4] A. Bosch, A. Zisserman, and X. Munoz. Image classification using random forests and ferns. In *ICCV*, 2007.

[5] A. Bosch, A. Zisserman, and X. Munoz. Representing shape with a spatial pyramid kernel. In *CIVR*, 2007.

[6] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *ECCV Int. Workshop on Stat. Learning in Computer Vision*, 2004.

[7] T. Harada, H. Nakayama, and Y. Kuniyoshi. Improving local descriptors by embedding global and local spatial information. In *ECCV*, 2010.

[8] T. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. In *NIPS*, 1999.

[9] K. Krapac, J. Verbeek, and F. Jurie. Spatial Fisher Vectors for Image Categorization. Research Report RR-7680, INRIA, July 2011.

[10] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.

[11] D. Liu, G. Hua, P. Viola, and T. Chen. Integrated feature selection and higher-order spatial feature extraction for object categorization. In *CVPR*, 2008.

[12] R. López-Sastre, T. Tuytelaars, F. Acevedo-Rodríguez, and S. Maldonado-Bascón. Towards a more discriminative and semantic visual vocabulary. *CVIU*, 115(3):415–425, 2011.

[13] F. Moosmann, B. Triggs, and F. Jurie. Fast discriminative visual codebooks using randomized clustering forests. In *NIPS*, 2007.

[14] N. Morioka and S. Satoh. Building compact local pairwise codebook with joint feature space clustering. In *ECCV*, 2010.

[15] D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *CVPR*, 2006.

[16] E. Nowak, F. Jurie, and B. Triggs. Sampling strategies for bag-of-features image classification. In *ECCV*, 2006.

[17] F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. In *CVPR*, 2007.

[18] F. Perronnin, C. Dance, G. Csurka, and M. Bressan. Adapted vocabularies for generic visual categorization. In *ECCV*, 2006.

[19] F. Perronnin, J. Sanchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *ECCV*, 2010.

[20] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007.

[21] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *CVPR*, 2008.

[22] T. Quack, V. Ferrari, B. Leibe, and L. van Gool. Efficient mining of frequent and distinctive feature configurations. In *ICCV*, 2007.

[23] S. Savarese, J. Winn, and A. Criminisi. Discriminative object class models of appearance and shape by correlatons. In *CVPR*, 2006.

[24] Y. Shinohara and N. Otsu. Facial expression recognition using fisher weight maps. *Automatic Face and Gesture Recognition*, 2004.

[25] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, 2003.

[26] J. van Gemert, C. Snoek, C. Veenman, A. Smeulders, and J.-M. Geusebroek. Comparing compact codebooks for visual categorization. *CVIU*, 114(4):450–462, 2010.

[27] J. van Gemert, C. Veenman, A. Smeulders, and J.-M. Geusebroek. Visual word ambiguity. *PAMI*, 32(7):1271–1283, 2010.

[28] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *CVPR*, 2010.

[29] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba. SUN database: Large scale scene recognition from abbey to zoo. In *CVPR*, 2010.

[30] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, 2009.

[31] J. Yang, K. Yu, and T. Huang. Efficient highly over-complete sparse coding using a mixture model. In *ECCV*, 2010.

[32] L. Yang, R. Jin, R. Sukthankar, and F. Jurie. Unifying discriminative visual codebook generation with classifier training for object category recognition. In *CVPR*, 2008.

[33] X. Zhou, N. Cui, Z. Li, F. Liang, and T. Huang. Hierarchical Gaussianization for image classification. In *ICCV*, 2009.