This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE

Local Pyramidal Descriptors for Image Recognition

Lorenzo Seidenari, Giuseppe Serra,
Andrew D. Bagdanov, Alberto Del Bimbo

Abstract—In this paper we present a novel method to improve the flexibility of descriptor matching for image recognition by using local multiresolution pyramids in feature space. We propose that image patches be represented at multiple levels of descriptor detail and that these levels be defined in terms of local spatial pooling resolution. Preserving multiple levels of detail in local descriptors is a way of hedging one's bets on which levels will most relevant for matching during learning and recognition. We introduce the Pyramid SIFT (P-SIFT) descriptor and show that its use in four state-of-the-art image recognition pipelines improves accuracy and yields state-of-the-art results. Our technique is applicable independently of spatial pyramid matching and we show that spatial pyramids can be combined with local pyramids to obtain further improvement. We achieve state-of-the-art results on Caltech-101 (80.1%) and Caltech-256 (52.6%) when compared to other approaches based on SIFT features over intensity images. Our technique is efficient and is extremely easy to integrate into image recognition pipelines.

Index Terms—Object categorization, local features, kernel methods.

1 Introduction

Object class recognition in images has been steadily gaining importance in the computer vision research community. Among the many image representation strategies, models based on local features that capture the most distinctive and dominant structures in the image have been widely used and demonstrate excellent performance. Feature-based representations of images typically consist of a set of local features extracted from patches around salient interest points or over regular grids [1], [2]. The Bag-of-Words (BOW) pipeline and its variants appeal to the analogy of text representation and retrieval [1] through use of frequency statistics of visual word occurrence as an image descriptor. Visual words are usually determined using k-means clustering on a sample of local features. Once local image features are mapped to dictionary words, a pooling stage accumulates local visual word frequency statistics into a global, histogram-based representation of the image suitable for recognition with classifiers such as support vector machines. A plethora of techniques have been proposed to improve the spatial pooling, feature quantization, and kernel classification stages of the BOW pipeline.

In this paper we propose a strategy for building local feature descriptors that capture local information at multiple levels of resolution. Our key idea, illustrated in figure 1 for SIFT features, is to define a local feature that, instead of being composed of a single resolution descriptor, is a multiresolution set of descriptors. This allows us to capture the appearance of a local patch at multiple levels of detail and to maintain distinctiveness, all while preserving invariance at each level of resolution. Our approach can be applied to any descriptor that can be naturally structured as a multiresolution set. Local image descriptors are typically computed through a common pipeline starting from raw image patches and applying a sequence of transformations that ends in a local spatial pooling of image measurements [3]. The spatial pooling stage is parameterized by the number, location and size of pooling regions. We can pyramidize any descriptor with such a stage simply by varying the size and density of pooling regions. Apart from SIFT [4], descriptors that incorporate a spatial pooling stage are HOGs [5], GLOH [6], DAISY-like descriptors [3] and SIFT-like color descriptors [7], among others. Our approach is complementary to both spatial pyramids and multi-scale local descriptor sampling. We demonstrate how our local pyramidal descriptors improve image classification results for the standard BOW approach, as well as for three successful and more recent encoding techniques: the Efficient Match Kernel [8], Locality-constrained Linear Coding [9] and Fisher vector models of image classification [10].

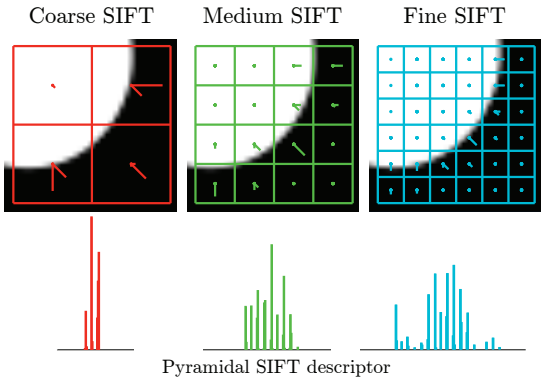


Fig. 1: The pyramidal SIFT descriptor is a set of three SIFT descriptors that describe the patch at different levels of detail.

In the next section we review work from the literature related to our approach and in section 3 we define a multiresolution pyramidal SIFT descriptor (the P-SIFT descriptor) which we use in our general image categorization framework based on the Sum Match Kernel. This framework is used in section 4 where we show how to incorporate pyramidal features into four state-of-the-art image recognition pipelines that can be thought of as approximations of the Sum Match Kernel approach. We show that each of these models lends itself well to incorporation of multiresolution descriptors and in section 5 that use of P-SIFT descriptors results in state-of-the-art performance on the Caltech-101 and Caltech-256 datasets.

2 Related work

A natural way to compute similarity between two images represented as sets of local features is the The Sum Match Kernel [11]. The intuition behind it is to incorporate information about all pairs of feature descriptors between the two sets. The Sum Match Kernel is interesting from a theoretical perspective, but in practice is computationally onerous as its calculation is quadratic in the number of features per image. Especially given the current trend towards large scale problems in image retrieval, it is important to adopt image representations and to use kernels that scale well in the number of images. Many state-of-the-art image recognition approaches, including the BOW model itself, are based on direct, efficient approximations of the Sum Match Kernel. Parsana et al. [12] proposed the neighborhood kernel that integrates feature co-occurrence and spatial

- *L. Seidenari, A. D. Bagdanov and A. Del Bimbo are with the Media Integration and Communication Center of the University of Florence, Viale Morgagni 65, 50139 Firenze, Italy. Email: {seidenari, bagdanov, delbimbo}@dsi.unifi.it*
- *G. Serra is with the University of Modena and Reggio Emilia, Via Vignolese 905, 41100 Modena, Italy. Email: giuseppe.serra@unimore.it*

Digital Object Indentifier 10.1109/TPAMI.2013.232

information of local features. Although these approaches yield state-of-the-art results, they have space and time complexity that is quadratic in the number of images and neighborhood size. To make the computation of such kernels more efficient, Bo et al. [8] recently proposed the Efficient Match Kernel (EMK) that maps local features to a low dimensional feature space and then constructs set-level features by averaging the resulting feature vectors.

Improvements to feature coding have focused primarily on better representations and/or reconstructions of local features, often using more than a single vocabulary descriptor. Zhang et al. [13] proposed an image classification framework that leverages non-negative sparse coding and sparse matrix decomposition. Similarly, Wang et al. [9] presented the Locality Constrained Linear Coding (LLC) technique that substitutes vector quantization. LLC utilizes a locality constraint to project each descriptor onto a local coordinate system and has been shown to improve over the BOW model when used in conjunction with max-pooling. Approaches like LLC are of particular interest because the representation yields state-of-the-art recognition results using linear SVMs, which is important for efficiency and scalability. Liu et al. [14] performed an in depth analysis of soft-assignment of local features to visual words. They show that soft-assignment, considering only the k-nearest words for coding, can be comparable to more complex LLC and sparse-coding techniques. Perronin et al. [10] proposed Fisher vectors as a global image representation based on the pooled gradients of local feature log-likelihoods with respect to the parameters of a generative model.

In the classic BOW histogram of visual word occurrences the relationships between local features are completely lost. It cannot account for the proximity of one word to another, the spatial configuration in which they appear, or their global coordinates in the image. To embed spatial information into the BOW representation, Lazebnik et al. [15] introduced the Spatial Pyramid Matching (SPM) kernel. It works by partitioning the image into increasingly finer sub-regions, computing the BOW histograms of local features in each sub-region, and concatenating the histograms to form the final representation of the image. Yang et al. [16] proposed an extension of the SPM approach which, instead of traditional k-means quantization, computes a spatial pyramid image representation based on sparse codes of SIFT features.

Rather than quantize sets of image features down to a histogram representation, some researchers have investigated alternative ways to compare differently-sized sets of local features. Grauman and Darrel [17] proposed the Pyramid Matching Kernel (PMK) that finds an approximate correspondence between two sets of feature points. Informally, their method takes a weighted sum of the number of matches that occur at each level of resolution, which are defined by placing a sequence of increasingly coarser grids over the feature space. At any resolution, two feature points match if they fall into the same cell of the grid. Matches at finer resolutions are weighted more than those at coarser ones. Boiman et al. [18] proposed a trivial nearest neighbor-based approach, the Naive-Bayes Nearest-Neighbor classifier (NBNN), which employs nearest neighbor distances in feature space. NBNN computes direct image-to-class distances without descriptor quantization. Removing the quantization step yielded a significant improvement in classification accuracy. This approach was later extended by Tuytelaars et al. [19] who introduced a kernelized version of NBNN. Duchenne et al. [20] proposed a graph-based image representation whose nodes and edges represent the regions associated with a coarse image grid and their adjacency relationships, respectively. The problem of matching two images is formulated as an energy minimization problem in a multi-label Markov Random Field.

3 Pyramidal SIFT descriptors for recognition

In this section we describe how we represent an image using local descriptor pyramids. We also describe a general framework for Bag Of Features (BOF) image representation and classification in terms of the Sum Match Kernel framework.

3.1 The P-SIFT descriptor

We consider SIFT descriptors [4] in an image I sampled on a regular grid. For a patch of size S per side we define the relative centers of the N^2 pooling region centers (e.g. in Fig. 1 the medium SIFT corresponds to $N = 4$) as the Cartesian product $R = C \times C$, where

$$C = \left\{ \left(i - \frac{1}{2}\right)\left(\frac{S}{N}\right) - \frac{S}{2} \mid i = 1, \ldots, N \right\}. \quad (1)$$

For a feature site $\mathbf{s}$ on the regular grid, the local pooling centers $R_{\mathbf{s}} = \{\mathbf{s} + \mathbf{c} \mid \mathbf{c} \in R\}$ are thus defined by the feature location $\mathbf{s}$ and the offsets defined by Eq. (1).

We define $I_\theta = \arctan\left(\frac{I_{y,\sigma}}{I_{x,\sigma}}\right)$ where $I_{x,\sigma}$ and $I_{y,\sigma}$ are Gaussian derivatives of image I at scale σ in the x and y directions, respectively. I_θ is quantized to 8 angles and for each pooling region (identified by its center $\mathbf{r} \in R_{\mathbf{s}}$), an orientation histogram is computed. When binning each angle, the contribution of pixel $\mathbf{p}$ in the patch centered at site $\mathbf{s}$ is weighted by its gradient magnitude at scale σ and a truncated triangular window:

$$w(\mathbf{p}, \mathbf{r}, \mathbf{s}) = ||\nabla_\sigma I(\mathbf{p})|| \cdot \max\left(0, 1 - \frac{||\mathbf{p} - \mathbf{r} - \mathbf{s}||}{S/2}\right). \quad (2)$$

The pyramidal SIFT (P-SIFT)[1] descriptor is constructed by varying the pooling resolution N that controls the number and size of each subregion used to compute each histogram. A P-SIFT consists of multiple SIFT descriptors that describe the patch at different levels of detail. We set the derivative scale σ according to the patch scale and number of pooling regions N^2 similarly to [21]. Figure 1 illustrates the construction of a P-SIFT descriptor consisting of three levels of resolution. The image feature (a circular edge) is captured at three levels of detail: for $N = 2$ (referred as coarse SIFT) practically indistinguishable from a corner, at $N = 4$ (medium SIFT) the circular structure begins to appear, and at $N = 6$ (fine SIFT) the circular structure is evident.

From now on we assume that an image I is represented as a set of local features X:

$$X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}, \quad (3)$$

where each local feature descriptor is a multiresolution P-SIFT descriptor consisting of L SIFT descriptors extracted at pooling resolutions $N_l \in \{N_1, \ldots, N_L\}$ for $l = 1, \ldots, L$:

$$\mathbf{x}_i = \left(\mathbf{x}_i^1, \mathbf{x}_i^2, \ldots, \mathbf{x}_i^L\right), \text{ for } i \in \{1, \ldots, n\}. \quad (4)$$

Each primitive descriptor $\mathbf{x}_i^l$ is a SIFT descriptor computed at the l-th pooling resolution N_l.

1. Source at: http://www.micc.unifi.it/seidenari/projects/p-sift/

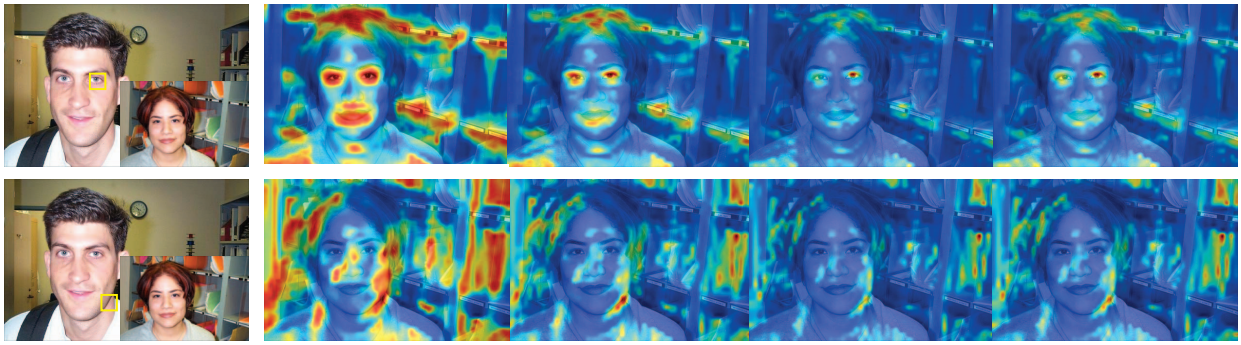


Fig. 2: Example of multi-resolution matching of patches (yellow rectangles) from face images of two subjects. (a) query patches and test image. (b-d) coarse, medium and fine P-SIFT responses on test image. (e) pyramidal kernel over P-SIFT responses.

3.2 The Sum Match Kernel over pyramidal descriptors

Here we show how the P-SIFT descriptor described in the previous section can be integrated into the Sum Match Kernel framework. Let X and Y be two images represented as Bags of Features. The normalized Sum Match Kernel is defined as:

$$K_S(X,Y) = \frac{1}{|X|}\frac{1}{|Y|}\sum_{\mathbf{x}\in X}\sum_{\mathbf{y}\in Y} k(\mathbf{x},\mathbf{y}), \tag{5}$$

where $|\cdot|$ is the cardinality of a set and $k(\mathbf{x},\mathbf{y})$ is a kernel expressing the similarity between two local descriptors.

When $\mathbf{x}$ and $\mathbf{y}$ are P-SIFT descriptors, where each descriptor is an ordered tuple of L SIFT descriptors as described in Section 3, our local kernel over P-SIFT descriptors is defined as a weighted sum of the similarities of the descriptors at each level of the local pyramid:

$$k(\mathbf{x},\mathbf{y}) = \sum_{l=1}^{L} w_l k_l(\mathbf{x},\mathbf{y}), \tag{6}$$

where w_l is the weight corresponding to local pyramid level l and $k_l(\mathbf{x},\mathbf{y})$ is a resolution-local kernel expressing the similarity between the primitive descriptors $\mathbf{x}$ and $\mathbf{y}$ at the l-th level of resolution. The similarity at each level in the local pyramid is weighted according to the description resolution at the corresponding level. If the L descriptors are arranged in ascending order of resolution, we define the weight at level l as $w_l = 2^{l-L}$. This weighting scheme, inspired by [17], [15], proved effective in preliminary experiments and is devised so that similarities at finer resolutions where features are most distinct are weighted more than those at coarser ones. Uniform and reversed weighting resulted in lower accuracy.

The final form of the normalized Sum Match Kernel over pyramidal features then becomes:

$$K_S(X,Y) = \frac{1}{|X|}\frac{1}{|Y|}\sum_{\mathbf{x}\in X}\sum_{\mathbf{y}\in Y}\sum_{l=1}^{L} w_l k_l(\mathbf{x},\mathbf{y}). \tag{7}$$

To give some intuition about the behavior of our pyramidal kernel, in Figure 2 we show an example of multi-resolution matching using the local kernel described in equation (6) over local descriptors from two face images taken from Caltech-101. In this example we use the local kernel $k_l(\mathbf{x},\mathbf{y}) = \exp(-\gamma||\mathbf{x}^l - \mathbf{y}^l||^2)$ to measure similarity at each level of the local pyramid. The first column shows two patches selected from a face image that are used as queries and, enclosed, a test image. The remaining columns show the similarity between the query patches and the dense, local patches from a test image at various levels of descriptor resolution. Observe that the two selected patches have different degrees of distinctiveness. In fact, while the eye patch has a strong distinctive character, the other patch can be approximated simply as an oblique edge. This difference in distinctiveness is also confirmed by the matching results using coarse, medium and fine descriptors. Indeed, for the eye patch we obtain a precisely localized response for patches around the same eye with the finest descriptor. The other query patch only matches with the same part of the face at the medium level.

It is also interesting to note that for the eye query patch in Figure 2, the medium level descriptor matches the left eye of the query image with both eyes in the test face image, which is a desirable property for image classification. The coarser descriptor instead matches patches with more translation (see again the eyes). This invariance comes at the cost of additional correspondences even with objects in the background that are completely unrelated to the query patch. The local pyramidal kernel is able to integrate information across multiple levels of resolution. The left eye is matched with both eyes in the test image, though it matches the left eye more strongly than the right.

The use of the normalized Sum Match Kernel defined in Eq. (5) comes at a high computational cost. Kernel evaluation is quadratic in the number of local features per image and linear in the number of resolution levels per local feature.

4 Image Recognition with P-SIFT Descriptors

In this section we show how to incorporate pyramidal features into four image recognition approaches that use different, efficient approximations of the normalized Sum Match Kernel to compare images. P-SIFT can be integrated in each of these frameworks at a cost that only adds complexity that is linear in the number of resolution levels.

4.1 Pyramid codebooks for BOW models

Pyramidal descriptors can be directly applied in the Bag of Words framework. Let $V = \{\mathbf{v}_1,\ldots,\mathbf{v}_D\}$ be a set of visual words. In the BOW approach each local feature is quantized into a $|D|$ dimensional binary vector $\mu(\mathbf{x}) = [\mu_1(\mathbf{x}),\ldots,\mu_D(\mathbf{x})]^\top$. In this embedding, $\mu_i(\mathbf{x})$ is equal to 1 if the $\mathbf{x}$ is associated to the visual word $\mathbf{v}_i$ and 0 otherwise. Descriptor $\mathbf{x}$ is associated to the nearest visual word $\mathbf{v}_i$. For a

linear classifier, the kernel function is:

$$K_{\text{BOW}}(X,Y) = \frac{1}{|X|}\frac{1}{|Y|}\sum_{\mathbf{x}\in X}\sum_{\mathbf{y}\in Y}\mu(\mathbf{x})^\top\mu(\mathbf{y}) = \frac{1}{|X|}\frac{1}{|Y|}\sum_{\mathbf{x}\in X}\sum_{\mathbf{y}\in Y}\delta(\mathbf{x},\mathbf{y}) \quad (8)$$

where $\delta(\mathbf{x},\mathbf{y}) = 1$ when $\mathbf{x}$ and $\mathbf{y}$ are associated to the same visual word and 0 otherwise.

Using a pyramidal descriptor we can define a dictionary at each resolution level and we obtain the following kernel:

$$K_{\text{BOW}}(X,Y) = \frac{1}{|X|}\frac{1}{|Y|}\sum_{\mathbf{x}\in X}\sum_{\mathbf{y}\in Y}\sum_{l=1}^{L} w_l\delta_l(\mathbf{x},\mathbf{y}) \quad (9)$$

where $\delta_l(\mathbf{x},\mathbf{y})$ is equal to 1 when the feature vectors $\mathbf{x}$ and $\mathbf{y}$ at resolution l are associated to the same visual word from the vocabulary of resolution level l, and 0 otherwise. This formulation allows us to inject the idea of pyramidal descriptors into a standard bag of words framework. The BOW approach is computationally cheap, compared to the normalized Sum Match Kernel, although the patch representation is based on a coarse approximation to the Sum Match Kernel and therefore linear embeddings retain less information with respect to more sophisticated reconstruction approaches [13], [9], [8].

4.2 Fisher vectors over P-SIFT descriptors

The Fisher vector technique uses a probability density function u_λ that models the generative process behind the descriptors appearing in an image X [10]. The Fisher kernel between X and Y is defined as:

$$K_{\text{FV}} = {G_\lambda^X}^\top F_\lambda^{-1} G_\lambda^Y, \quad (10)$$

where F_λ is the Fisher information matrix of u_λ and G_λ^X is the gradient of the log-likelihood of the data X with respect to the parameters λ of the generative model:

$$G_\lambda^X = \nabla_\lambda \log u_\lambda(X). \quad (11)$$

Using the Cholesky factorization of $F_\lambda^{-1} = L_\lambda^\top L_\lambda$ and defining $\mathcal{G}_\lambda^X = L_\lambda G_\lambda^X$ we can rewrite (10) as an inner product:

$$K_{\text{FV}}(X,Y) = {\mathcal{G}_\lambda^X}^\top \mathcal{G}_\lambda^Y. \quad (12)$$

Assuming that descriptors in X are independent, and thus $u_\lambda(X) = \prod_{\mathbf{x}\in X} u_\lambda(\mathbf{x})$, the Fisher vector of image X is a normalized sum of gradients at each point $\mathbf{x}\in X$ with respect to the model parameters λ:

$$\mathcal{G}_\lambda^X = \sum_{\mathbf{x}\in X} L_\lambda \nabla_\lambda \log u_\lambda(\mathbf{x}), \quad (13)$$

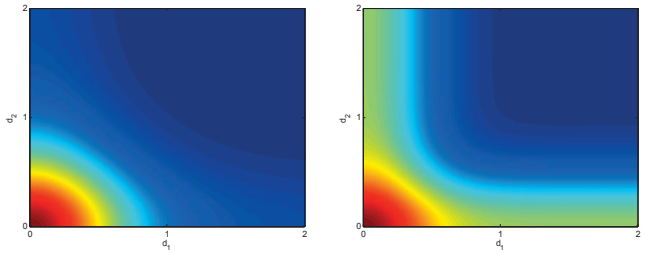
The Fisher vector approach works well because it embeds the original descriptors in a high-dimensional space amenable to linear classification. We can interpret the Fisher kernel in equation (10) as a Sum Match Kernel over P-SIFT descriptors:

$$K_{\text{FV}}(X,Y) = \frac{1}{|X|}\frac{1}{|Y|}\sum_{\mathbf{x}\in X}\sum_{\mathbf{y}\in Y}\sum_{l=1}^{L} k_l(\mathbf{x}^l,\mathbf{y}^l), \quad (14)$$

where the local kernel at pooling resolution level l is defined as:

$$k_l(\mathbf{x},\mathbf{y}) = (L_{\lambda_l}\nabla_{\lambda_l}\log u_{\lambda_l}(\mathbf{x}))^\top (L_{\lambda_l}\nabla_{\lambda_l}\log u_{\lambda_l}(\mathbf{y})), \quad (15)$$

where λ_l are the parameters of the generative model at resolution level l. We use a mixture of Gaussians for each u_{λ_l} and take gradients with respect to the means and diagonal covariance of the mixtures at each resolution level l.



(a) $\left(e^{d_1}+e^{d_2}\right)^q$ (b) $e^{qd_1}+e^{qd_2}$

Fig. 3: The difference between power normalization of the entire sum (a) and power normalization of terms (b).

4.3 Efficient Match Kernels over P-SIFT descriptors

Plugging in a radial-basis kernel as the local kernel used for comparing descriptors of corresponding resolutions into Eq. (7), we obtain the following Sum Match Kernel:

$$K(X,Y) = \frac{1}{|X|}\frac{1}{|Y|}\sum_{\mathbf{x}\in X}\sum_{\mathbf{y}\in Y}\sum_{l=1}^{L} 2^{l-L} e^{(-\gamma||\mathbf{r}^l-\mathbf{s}^l||^2)}. \quad (16)$$

We can define an efficient kernel between sets based on our pyramidal descriptors that approximates Eq. (16). Our approximation is achieved by generalizing the Efficient Match Kernel [8] to multiresolution local features.

Let $\phi(\cdot)$ represent the infinite dimensional feature map corresponding to the kernel $k(\mathbf{x},\mathbf{y})$ from Eq. (16). That is:

$$\begin{aligned} k(\mathbf{x},\mathbf{y}) &= \sum_{l=1}^{L} w_l k_l(\mathbf{x},\mathbf{y}) \\ &= \sum_{l=1}^{L} 2^{l-L} e^{(-\gamma||\mathbf{r}^l-\mathbf{s}^l||^2)} \\ &= [\phi_1(\mathbf{x}),\ldots,\phi_L(\mathbf{x})]^\top [\phi_1(\mathbf{y}),\ldots,\phi_L(\mathbf{y})] \end{aligned} \quad (17)$$

The feature maps $\phi_l(\cdot)$ are infinite dimensional due to the use of the exponential kernel. We proceed estimating a finite-dimensional approximation to these embeddings by reconstructing them as linear combinations of learned basis vectors.

We approximate the embedding for each resolution level $\phi_l(\mathbf{x})$ by solving the following minimization problem:

$$\bar{\mathbf{v}}_l(\mathbf{x}) = \arg\min_{\mathbf{v}} ||\phi_l(\mathbf{x}) - \mathtt{H}_l\mathbf{v}||^2 \quad (18)$$

where $\mathtt{H}_l = [\phi_l(\mathbf{z}_1^l)\ldots\phi_l(\mathbf{z}_{D_l}^l)]$ is a basis of vectors in the infinite dimensional feature space induced by the feature map ϕ_l. The vectors $\mathbf{z}_i^l$ constitute a visual vocabulary of D_l SIFT descriptors for resolution level l.

Solving (18) and replacing $\phi_l(\cdot)$ with $\mathtt{H}_l\bar{\mathbf{v}}_l(\cdot)$ we have

$$k_l(\mathbf{x},\mathbf{y}) = \phi_l(\mathbf{x})^\top\phi_l(\mathbf{y}) \approx k_{\mathbf{z}^l}(\mathbf{x})\mathtt{K}_{\mathbf{z}^l}^{-1}k_{\mathbf{z}^l}(\mathbf{y}), \quad (19)$$

where $\mathtt{K}_{\mathbf{z}^l}$ is the Gramian of $k_l(\cdot,\cdot)$ on the D_l visual words at level l and $k_{\mathbf{z}^l}(\mathbf{x})$ is a vector of kernel evaluations between a feature $\mathbf{x}$ at level l and the basis elements for the same level $\mathbf{z}_i^l$ for $i\in\{1,\ldots D_l\}$.

Using the Cholesky decomposition of $\mathtt{K}_{\mathbf{z}^l}^{-1} = \mathtt{G}_l^\top\mathtt{G}_l$ and substituting the approximations of Eq. (19) into Eq. (17) we obtain the final approximate pyramidal kernel:

$$\begin{aligned} \hat{k}(\mathbf{x},\mathbf{y}) =& [\sqrt{w_1}\mathtt{G}_1 k_{\mathbf{z}^1}(\mathbf{x})\ldots\sqrt{w_L}\mathtt{G}_L k_{\mathbf{z}^L}(\mathbf{x})]^\top \\ & [\sqrt{w_1}\mathtt{G}_1 k_{\mathbf{z}^1}(\mathbf{y})\ldots\sqrt{w_L}\mathtt{G}_L k_{\mathbf{z}^L}(\mathbf{y})]. \end{aligned} \quad (20)$$

The Sum Match Kernel and its approximations perform well in terms of recognition, but has the drawback that every

similarity between pairs of features $k_l(\mathbf{x}, \mathbf{y})$ contributes equally to the overall feature set similarity of Eq. (7). The result can be that many weakly similar feature pairs drown out the relatively few strongly similar ones. To address this, we perform a power normalization on scale-local similarity comparisons in order to accentuate highly-similar pairs, while minimizing the influence of weakly similar ones. For some $q > 1$, the Sum Match Kernel becomes:

$$K_{\text{EMK}}(X, Y) = \frac{1}{|X|}\frac{1}{|Y|}\sum_{\mathbf{x}\in X}\sum_{\mathbf{y}\in Y}\sum_{l=1}^{L} w_l k_l(\mathbf{x}, \mathbf{y})^q. \tag{21}$$

Using this type of power normalization effectively makes the feature selective in that if any of the levels of resolutions match well between the descriptors $\mathbf{x}$ and $\mathbf{y}$, the overall kernel will reflect this. This property is illustrated in Figure 3 where we show the difference in behavior when the power is taken inside or outside the sum of scale-local kernels. For any positive integer q the power normalized kernel is still Mercer since it can be written as a product of Mercer kernels.

4.4 Locality Constrained Coding of P-SIFT descriptors

Locality-constrained Linear Coding (LLC) is a technique that encodes local feature descriptors using an overcomplete basis or dictionary. Each descriptor is represented by reconstructing it with a sparse combination of words from a visual vocabulary. Coding of feature descriptors using LLC works particularly well when integrating global information into kernel computations through max-pooling of codes over larger regions [9]. It can also be thought of as an approximation of the Sum Match Kernel representation, one that uses local information to code features and that incorporates non-local information through max-pooling.

In the classical sparse coding approach, sparsity is enforced through an ℓ_1 regularization term. In LLC, both sparsity and locality are obtained by constraining the reconstruction for each descriptor to use only its k nearest neighbors. Formally, the code $\mathbf{c}(\mathbf{x}) = [c_1(\mathbf{x}), \ldots, c_{|V|}(\mathbf{x})]$ for a descriptor $\mathbf{x}$ is computed as the solution of the following optimization problem:

$$\begin{aligned} &\mathbf{c}(\mathbf{x}) = \arg\min_{\mathbf{c}} ||\mathbf{x} - \mathtt{B}_{\mathbf{x},k}\mathbf{c}||^2 + \lambda||\mathbf{c}||^2 \\ &\text{s. t. } \mathbf{1}^\top\mathbf{c} = 1, \end{aligned} \tag{22}$$

where $\mathtt{B}_{\mathbf{x},k}$ is the local basis constructed by the k nearest visual words of descriptor $\mathbf{x}$ from dictionary V.

To incorporate max-pooling into the matching between two LLC-encoded images, we can formulate the local kernel as follows. Given two descriptors $\mathbf{x}$ and $\mathbf{y}$ from two images X and Y, we form the max-pooled local kernel:

$$k_l(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{|V|} c_i(\mathbf{x})c_i(\mathbf{y})\mu_i(\mathbf{x})\mu_i(\mathbf{y}), \tag{23}$$

where

$$\mu_i(\mathbf{x}) = \begin{cases} 1 & \text{if } c_i(\mathbf{x}) \geq c_i(\mathbf{x}') \ \forall \mathbf{x}' \in X \\ 0 & \text{otherwise.} \end{cases} \tag{24}$$

Each $c_i(\cdot)$ is a single dimension of an LLC code, while the $\mu_i(\cdot)$ act as selector functions that ensure that the corresponding $c_i(\cdot)$ contributes to the kernel if and only if it is the maximum in dimension i over all local features in the image.

As with the other approaches above, we can extend the local kernel k_l to take into account the different resolutions of each descriptor:

$$k(\mathbf{x}, \mathbf{y}) = \sum_{l=1}^{L} w_l k_l(\mathbf{x}^l, \mathbf{y}^l). \tag{25}$$

For each $l \in \{1 \ldots L\}$ we define the max-pooled LLC codes for resolution level l:

$$\Phi_l(X) = \left[\max_{\mathbf{x}\in X} c_1^l(\mathbf{x}^l), \ldots, \max_{\mathbf{x}\in X} c_{|D_l|}^l(\mathbf{x}^l)\right], \tag{26}$$

where D_l is size of the visual vocabulary for resolution l. Defining the complete linear embedding as the concatenation of all levels:

$$\Phi(X) = [\Phi_1(X), \Phi_2(X), \ldots, \Phi_L(X)], \tag{27}$$

it results from Eq. (25) and Eq. (26) that the similarity between two images X and Y represented by pyramidal descriptors is:

$$K_{\text{LLC}}(X, Y) = \sum_{\mathbf{x}\in X}\sum_{\mathbf{y}\in Y} k(\mathbf{x}, \mathbf{y}) \tag{28}$$

$$= \sum_{\mathbf{x}\in X}\sum_{\mathbf{y}\in Y}\sum_{l=1}^{L} w_l k_l(\mathbf{x}^l, \mathbf{y}^l) \tag{29}$$

$$= \Phi(X)^\top\Phi(Y). \tag{30}$$

From this we see that pyramidal, max-pooled LLC image representations can be compared using the local kernel formulation of Eq. (25), or equivalently as the scalar product between concatenated, pyramidal embeddings as in Eq. (30).

5 Experimental Results

We evaluated the performance of our pyramidal descriptors on Caltech-101 and Caltech-256. For both datasets we resize images so that their longest dimension is 300 pixels [22]. We compute P-SIFT descriptors at three patch sizes (24, 32 and 40 pixels) over a dense regular grid with a spacing of 6 pixels. The extracted pyramidal descriptor for a given patch size consists of a set of SIFT descriptors at three different spatial pooling resolutions: 2×2, 4×4 and 6×6. Spatial pyramids are used to partition the whole image using configurations 1×1, 2×2 and 4×4. In the following SP0 refers to the first pyramid level with no spatial partitioning, SP1 to the concatenation of the first and second, and SP2 for all three. We use linear SVMs for classification [23] and all classification accuracies reported are the average over five independent training and test set splits.

To determine the appropriate size for the visual vocabularies of each resolution level (i.e. to balance the trade-off between reconstruction accuracy and memory consumption), we evaluated codebook quality by analyzing errors computed using the EMK approximation in Eq. (18). We used a subsample of 150k SIFT descriptors and ran k-means to learn vocabularies over a range of sizes. In general, as can be seen in Fig. 4, the reconstruction error is high when a limited set of visual words is used, but decreases rapidly with increasing vocabulary size. We also observed that reconstruction error at the coarse level is less than that at finer levels, mainly due to the higher distinctiveness of fine descriptors. The error typically saturates and after a point there is no advantage in increasing vocabulary size. Based on this error analysis, and considering the dimensionality of the final image descriptor, we selected 1,000, 2,000 and 2,500 visual words for coarse, medium and fine levels, respectively, for Caltech-101. For Caltech-256 we found 3,000, 4,000 and 4,500 visual words to be appropriate sizes. We use these vocabulary sizes for all experiments on the

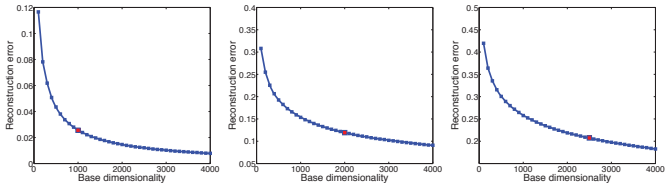


Fig. 4: Reconstruction error as a function of the base dimensionality for three levels of descriptor resolution on Caltech-101. Red dots highlight the dimensionality of the bases selected.

P-EMK, P-BOW and P-LLC approaches. For Fisher vectors we used PCA as recommended in [10] and used 20, 60 and 80 principal components, and 64, 128 and 256 Gaussians for the respective dictionaries.

Caltech-101 [24] consists of 9,144 images from 101 object categories plus one background category. The number of examples per category varies from 31 to 800 images. Object categories exhibit color and shape variation, but objects are all centered and have no viewpoint diversity. We train models on 15 or 30 randomly selected images per category and test on the remaining images. Caltech-256 [25] consists of 30,607 images from 256 object categories plus background. Each class contains at least 80 images. Caltech-256 is challenging due to high variations in object size, location and pose. To evaluate classification performance we follow the standard setup: 30 or 60 images were randomly selected from all the categories for training, and the remaining images were used for testing.

5.1 The contribution of multi-resolution descriptors

To visualize how our multi-resolution representation improves classification accuracy, we generated object-centric relevance maps. We first train individual classifiers at each of the three single levels of resolution. Then, for each test image correctly recognized by all three learned classifiers, we iteratively remove each patch descriptor and compute the variation of the distance from the learned margin:

$$\Delta f(\hat{\mathbf{x}}) = \boldsymbol{\beta}_C^\top \cdot \left(\frac{1}{|X|} \sum_{\mathbf{x} \in X \setminus \hat{\mathbf{x}}} \bar{\phi}(\mathbf{x}) \right) - \boldsymbol{\beta}_C^\top \cdot \left(\frac{1}{|X|} \sum_{\mathbf{x} \in X} \bar{\phi}(\mathbf{x}) \right) \quad (31)$$

where $\hat{\mathbf{x}}$ is the removed patch and $\boldsymbol{\beta}_C$ is the learned hyperplane for the correct class C.

A negative variation is a cue of relevance of that particular patch, while a positive variation indicates that removing the patch improves the confidence for the correct class. Values of $\Delta f(\hat{\mathbf{x}})$ are accumulated at the locations of patches $\hat{\mathbf{x}}$. A final relevance map for a class is obtained by cropping the object using the ground-truth annotations and averaging over all cropped and resized relevance maps for an object category. The final relevance map size is the average size of the annotated examples. In Figure 5 we show the three classes with the best (Figure 5a) and worst (Figure 5b) improvement in accuracy compared to a classifier trained using standard SIFT and EMK. We can observe that, for classes with the highest improvement in accuracy, each resolution has a distinct spatial relevance pattern. For classes with lower improvement the relevance maps are quite similar, which means that all levels concentrate on representing the same parts. It should be noted that classes with lower improvement are easier and exhibit less intra-class variation. The individual relevance maps in Figure 5a show that this intra-class variation is captured by the different resolution levels of our descriptor, each focusing on a different global object layout.

TABLE 1: Accuracy at different descriptor and spatial pyramid resolution levels on Caltech-101.

Coarse	Medium	Fine	SP0	SP1	SP2
✓	✓	✓	**65.45**	**75.10**	**78.31**
✓	✓	-	64.37	74.74	77.20
✓	-	-	59.35	71.55	75.33
-	✓	-	62.97	73.62	76.82
-	-	✓	61.46	73.38	76.36

We performed another set of experiments to quantify how each level of resolution contributes to improving classification accuracy. First, we tested the classification performance of our method obtained by adding each resolution level in turn to the descriptor. Table 1 summarizes classification accuracy on Caltech-101 for three spatial pyramid levels using 30 training images per class. We can observe that, although the coarser level is quite descriptive, the use of more discriminative information considerably increases performance. In fact, without the spatial pyramid the improvement is about six percentage points (from 59.35% to 65.45%).

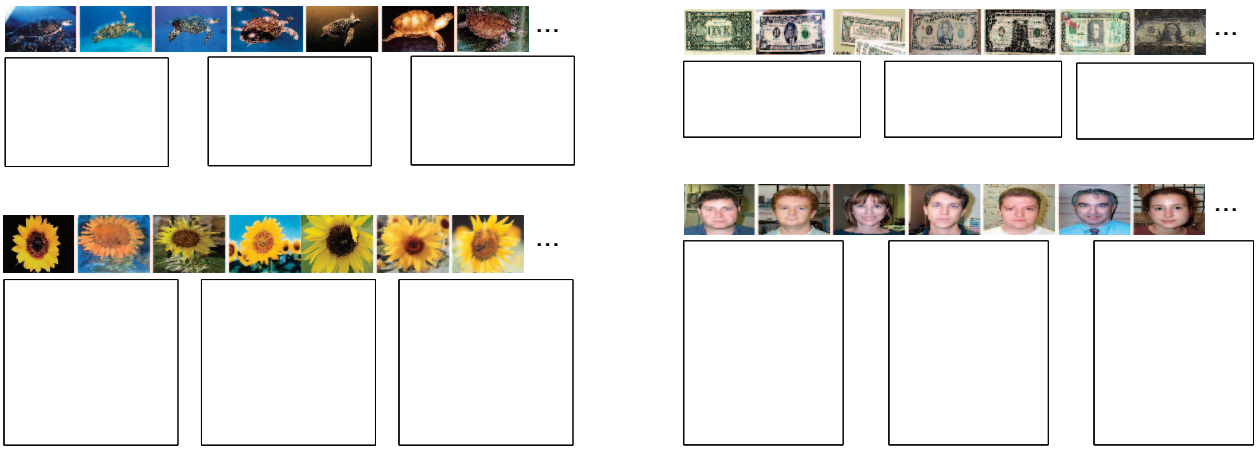
The best single-resolution performance is obtained using the medium resolution descriptor, which corresponds to the standard SIFT descriptor. The coarse and the fine descriptors lose few percentage points because the coarse descriptors are not discriminative enough while the fine ones are too discriminative. However, the accuracy achieved from our pyramidal descriptors is higher. This is due to the fact that the pyramidal descriptor has several levels of distinctiveness that are used adaptively by the pyramidal kernel. From Table 1 we see that both spatial and feature pyramids contribute to improved classification accuracy. Starting from just the coarse resolution descriptor and three levels of spatial pyramid, adding the medium and fine resolutions yields an increase of about three percentage points in classification accuracy.

5.2 Pyramidal descriptors for image recognition

In this experiment we compare the extension of the BOW model with pyramidal descriptors (see Section 4.1) and a standard BOW on Caltech-101. We used linear and nonlinear[2] SVM classifiers with 30 training images per category. In Table 2 we report the accuracy for the three spatial pyramid levels SP0, SP1 and SP2 described above. The codebook size for Linear and Hellinger BOW is fixed to 4,000, as we observed that the performance tends to saturate beyond this. For the extended bag-of-words we use 1,000, 2,000 and 2,500 as the codebook size for the coarse, medium and fine levels, respectively. In both the linear and non-linear cases, the P-SIFT descriptor consistently outperforms the corresponding multi-scale SIFT baseline at all pyramid levels. These results show that using pyramidal SIFT descriptors and pyramidal dictionaries can improve the standard BOW model.

In Table 2 we also show a comparison of the baseline methods (BOW, EMK, LLC and FV) and their pyramidized versions with single- and multi-scale sampling. In rows indicated with "3S" we sample standard SIFT at patch sizes 24, 32 and 64 so that the pooling region sizes are comparable to those in single-scale P-SIFT at patch size 32. Similarly, in rows indicated with

2. All experiments with non-linear kernels use the Hellinger kernel $K(x, y) = \sum_i \sqrt{x_i y_i}$ which improves histogram comparison by discounting small contributions to dimensions with large magnitudes [10].



(a) Classes benefitting most from pyramidal descriptor (b) Classes benefitting least from pyramidal descriptor

Fig. 5: Classes benefitting most (a) and least (b) from pyramidal representation. For each class we show sample images and below the classifier responses calculated for each resolution level computed from Eq. (31) and averaged over the entire training set for that class. Classifier responses are ordered from coarse to fine.

TABLE 2: Accuracy of baseline encoding methods and their pyramidized versions (P-*) on Caltech-101.

Features	Kernel SVM	SP0	SP1	SP2
BOW (3S)	Linear	48.91	56.93	60.51
P-BOW	Linear	49.18	57.86	61.34
BOW (7S)	Linear	50.21	57.62	61.74
P-BOW (M)	Linear	**51.82**	**59.77**	**64.46**
BOW (3S)	Hellinger	60.22	70.22	72.53
P-BOW	Hellinger	60.35	70.35	73.06
BOW (7S)	Hellinger	61.39	70.81	73.26
P-BOW (M)	Hellinger	**62.55**	**71.85**	**74.27**
EMK (3S)	Linear	58.56	71.68	74.59
P-EMK	Linear	65.45	75.10	78.31
EMK (7S)	Linear	57.02	71.75	74.55
P-EMK (M)	Linear	**66.11**	**75.69**	**78.36**
LLC (3S)	Linear	60.26	72.24	75.47
P-LLC	Linear	63.31	73.13	75.59
LLC (7S)	Linear	63.83	72.02	75.13
P-LLC (M)	Linear	**66.61**	**76.29**	**78.75**
FV (3S)	Linear	70.44	75.83	76.26
P-FV	Linear	70.82	77.16	78.80
FV (7S)	Linear	70.48	74.99	76.83
P-FV (M)	Linear	**71.02**	**77.83**	**80.13**

"7S" we sample standard SIFT at patch sizes 16, 24, 32, 40, 48, 64 and 80 to obtain pooling regions comparable to multi-scale sampling of P-SIFT descriptors at patch sizes 24, 32 and 40 (indicated with "M" in Table 2).

Sampling multiple patch sizes is beneficial in all cases, and LLC benefits so much from it likely due to the max-pooling stage unique to it among tested methods. Note that single-scale P-EMK, P-LLC and P-FV already outperform both multi-scale BOW baselines for nearly all spatial pyramid configurations. The best results are consistently achieved with multi-scale descriptor sampling and our local, pyramidized descriptor. EMK, LLC and FV better preserve local feature representation and indeed exhibit better results, and the improvement of P-SIFT is even more dramatic for EMK and LLC when the spatial pyramid is not used. This suggests that our technique improves feature matching and that this improvement is less noticeable when spatial pyramids avoid confusion by imposing geometric constraints on local feature matching.

5.3 Comparison to the state-of-the-art

We compare our results with several existing approaches that use comparable image representations (dense sampling of SIFT descriptors) on both Caltech-101 and Caltech-256[3]. In Table 3 we report a comparison between our P-SIFT based approaches and the state-of-the-art. P-EMK, P-LLC and P-FV all perform comparably, with P-FV outperforming all methods. For completeness, at the bottom of Table 3 we include results from more complex approaches that incorporate many cues and learn optimal feature combinations [26], [27], or that use global alignment kernels [20]. Though not strictly comparable with our approach, we do outperform more complex techniques such as [20] and [28] on Caltech-256. Note also that our P-SIFT features can be considered complementary to these approaches and integrating multiple descriptor resolutions into them should yield improved results.

6 Conclusions

In this paper we described an approach to image recognition using multi-resolution, pyramidized local feature descriptors. Our P-SIFT descriptor uses three levels of local pooling resolution to construct a discriminative, local feature representation for image classification. We further showed how our image representation can be used within the BOW, EMK, LLC and Fisher vector techniques to improve classification performance. The P-SIFT feature is simple and easy to implement, and it naturally complements a range of image coding and classification techniques.

The performance of the P-SIFT descriptor for image classification is comparable to the state-of-the-art on Caltech-101 and exceeds the state-of-the-art on Caltech-256. Our approach, using only SIFT descriptors over intensity images, linear classifiers and no global feature alignment, outperforms significantly more complex methods, especially on Caltech-256.

P-SIFT features can be incorporated into a BOW pipeline at marginal cost. The increase in complexity is linear in the number of resolution levels introduced and the size of the vocabularies of each level. On Caltech-101, for example, our

3. More results at http://zybler.blogspot.com/2009/08/ and http://zybler.blogspot.com/2009/10/

TABLE 3: Comparison with the-state-of-the-art for Caltech-101 and Caltech-256.

	Methods	Caltech-101		Caltech-256	
		15 Training	30 Training	30 Training	60 Training
P-FV (M)	P-SIFT + Fisher encoding + SPM + Linear SVM	**71.47**	**80.13**	**44.86**	**52.59**
P-LLC (M)	P-SIFT + LLC encoding + SPM + Linear SVM	68.25	78.75	42.24	48.91
P-EMK (M)	P-SIFT + EMK encoding + SPM + Linear SVM	70.10	78.31	42.08	48.85
Bo et al. [8]	EMK + SPM + Kernel SVM	60.50	73.86	30.50	37.60
Tuytelaars et al. [19]	Kernelized NBNN + Spatial correspondences	69.20	75.20	37.00	-
Fisher Vectors [22], [10]	Fisher encoding + SPM + Linear SVM	-	77.78	40.80	47.90
Carreira et al. [29]	Second Order Pooling + SPM + Linear SVM	-	79.20	-	-
C. Zhang et al. [13]	Non-negative sparse coding + SPM	69.58	75.68	-	-
Wang et al. [9]	LLC + SPM + Linear SVM	65.43	73.40	41.19	47.68
Grauman et al. [17]	Pyramid match kernel	50.00	58.20	-	-
Yang et al. [16]	Sparse codes + SPM + Linear SVM	67.00	73.20	34.00	40.10
Lazebnik et al. [15]	Hard quantization + SPM + Kernel SVM	56.40	64.60	-	-
Duchenne et al. [20]	Graph-Matching + Kernel SVM	75.30	80.30	38.10	-
Cao et al. [28]	Sparse codes + superpixels + attributes + linear SVM	-	-	38.74	45.43
Todorovic et al. [30]	Segmentation tree + Subcategories + Linear SVM	71.60	81.90	49.50	-
Bo et al. [26]	RGB + Sparse codes + Deep Learning + Linear SVM	-	82.50	50.70	58.00
Gehler et al. [27]	Multiple Features + SPM + LP-β	-	77.80	45.80	-

final image descriptor dimensionality is only 5,500 after incorporating multiple levels of resolution. Thus our representation is comparable in size with the typical 4,000 visual words needed to obtain state-of-the-art results using the vanilla BOW approach. This little added complexity and the good performance with linear SVMs are key contributions considering the recent trend toward large scale image recognition.

The pyramidized local descriptors that we propose are complementary to many existing image representation and coding techniques. We demonstrated this with the BOW, EMK, LLC and Fisher vector approaches in this work, but the P-SIFT descriptor could be used in more complex image representation and matching frameworks which perform global alignment of image features before recognition. We expect similar performance gains when combined with more complex image matching techniques.

REFERENCES

[1] J. Sivic, A. Zisserman, Efficient visual search cast as text retrieval, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (4) (2009) 591–606. 1

[2] L. Fei-Fei, P. Perona, A bayesian hierarchical model for learning natural scene categories, in: Proc. of CVPR, 2005. 1

[3] M. Brown, G. Hua, S. Winder, Discriminative learning of local image descriptors, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (1) (2011) 43–57. 1

[4] D. G. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2) (2004) 91–110. 1, 2

[5] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Proc. of CVPR, 2005. 1

[6] K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors, Pattern Analysis and Machine Intelligence, IEEE Transactions on 27 (10) (2005) 1615 –1630. 1

[7] K. E. A. van de Sande, T. Gevers, C. G. M. Snoek, Evaluating color descriptors for object and scene recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (9) (2010) 1582–1596. 1

[8] L. Bo, C. Sminchisescu, Efficient Match Kernel between Sets of Features for Visual Recognition, in: Proc. of NIPS, 2009. 1, 2, 4, 8

[9] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, Y. Gong, Locality-constrained linear coding for image classification, in: Proc. of CVPR, 2010. 1, 2, 4, 5, 8

[10] F. Perronnin, J. Sanchez, T. Mensink, Improving the fisher kernel for large-scale image classification, in: Proc. of ECCV, 2010. 1, 2, 4, 6, 8

[11] S. Lyu, Mercer kernels for object recognition with local features, in: Proc. of CVPR, 2005. 1

[12] M. Parsana, S. Bhattacharya, C. Bhattacharyya, K. R. Ramakrishnan, Kernels on attributed pointsets with applications, in: Proc. of NIPS, 2007. 1

[13] C. Zhang, J. Liu, Q. Tian, C. Xu, H. Lu, S. Ma, Image classification by non-negative sparse coding, low-rank and sparse decomposition, in: Proc. of CVPR, 2011. 2, 4, 8

[14] L. Liu, L. Wang, X. Liu, In defense of soft-assignment coding, in: Proc. of ICCV, 2011. 2

[15] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, in: Proc. of CVPR, 2006. 2, 3, 8

[16] J. Yang, K. Yu, Y. Gong, T. Huang, Linear spatial pyramid matching using sparse coding for image classification, in: Proc. of CVPR, 2009. 2, 8

[17] K. Grauman, T. Darrell, The pyramid match kernel: Efficient learning with sets of features, Journal of Machine Learning Research 8 (2007) 725–760. 2, 3, 8

[18] O. Boiman, E. Shechtman, M. Irani, In defense of nearest-neighbor based image classification, in: Proc. of CVPR, 2008. 2

[19] T. Tuytelaars, M. Fritz, K. Saenko, T. Darrell, The nbnn kernel, in: Proc. of ICCV, 2011. 2, 8

[20] O. Duchenne, A. Joulin, J. Ponce, A graph-matching kernel for object categorization, in: Proc. of ICCV, 2011. 2, 7, 8

[21] A. Vedaldi, B. Fulkerson, Vlfeat – an open and portable library of computer vision algorithms, in: Proc. of ACM-MM, 2010. 2

[22] K. Chatfield, V. Lempitsky, A. Vedaldi, A. Zisserman, The devil is in the details: an evaluation of recent feature encoding methods, in: Proc. of BMVC, 2011. 5, 8

[23] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, C.-J. Lin, Liblinear: A library for large linear classification, Journal of Machine Learning Research 9 (2008) 1871–1874. 5

[24] L. Fei-Fei, R. Fergus, P. Perona, One-shot learning of object categories, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (4) (2006) 594 –611. 6

[25] G. Griffin, A. Holub, P. Perona, Caltech-256 object category dataset., Tech. rep., California Institute of Technology. URL http://authors.library.caltech.edu/7694 6

[26] L. Bo, X. Ren, D. Fox, Multipath sparse coding using hierarchical matching pursuit, in: Proc of CVPR, 2013. 7, 8

[27] P. Gehler, S. Nowozin, On feature combination for multiclass object classification, in: Proc. of ICCV, 2009. 7, 8

[28] L. Cao, R. Ji, Y. Gao, Y. Yang, Q. Tian, Weakly supervised sparse coding with geometric consistency pooling, in: Proc. of CVPR, 2012. 7, 8

[29] J. a. Carreira, R. Caseiro, J. Batista, C. Sminchisescu, Semantic segmentation with second-order pooling, in: Proc. ECCV, 2012, pp. 430–443. 8

[30] S. Todorovic, N. Ahuja, Learning subcategory relevances for category recognition, in: Proc. of CVPR, 2008. 8