Supervised Kernel Descriptors for Visual Recognition

Peng Wang[1] Jingdong Wang[2] Gang Zeng[1] Weiwei Xu[3] Hongbin Zha[1] Shipeng Li[2]
[1]Peking University [2]Microsoft Research Asia [3]Hangzhou Normal University

Abstract

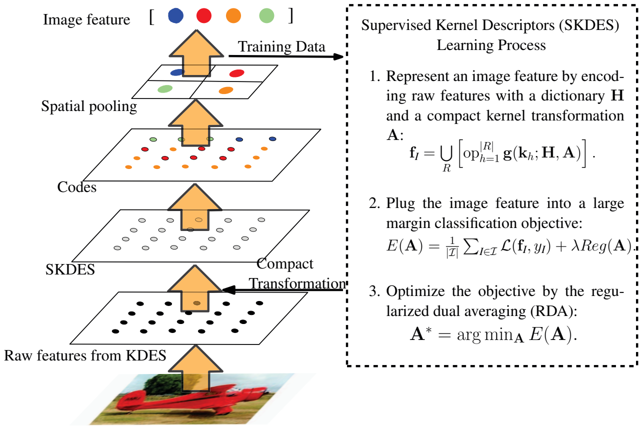
In visual recognition tasks, the design of low level image feature representation is fundamental. The advent of local patch features from pixel attributes such as SIFT and LBP, has precipitated dramatic progresses. Recently, a kernel view of these features, called kernel descriptors (KDES) [1], generalizes the feature design in an unsupervised fashion and yields impressive results.

In this paper, we present a supervised framework to embed the image level label information into the design of patch level kernel descriptors, which we call supervised kernel descriptors (SKDES). Specifically, we adopt the broadly applied bag-of-words (BOW) image classification pipeline and a large margin criterion to learn the low-level patch representation, which makes the patch features much more compact and achieve better discriminative ability than KDES. With this method, we achieve competitive results over several public datasets comparing with state-of-the-art methods.

1. Introduction

For many visual recognition tasks, one of the critical problems is to discover robust image representations (features). The feature design is very challenging because on one hand, image features should be invariant to the inner-class variation, such as object translation, lighting changes, shape changes of non-rigid objects; on the other hand, image features also need to be discriminative regarding the inter-class differences for separating confusing classes.

To handle these challenges, current state-of-the-art image classification algorithms adopt the bag-of-words pipeline that firstly extracts low-level patch based descriptors, then encodes them into a middle level representation through an over-complete dictionary, and finally obtains image features by a spatial pooling strategy [2, 8, 10, 18, 42]. Within such a pipeline (showed in Fig. 1), much emphasis has been directed at coding the patch descriptors such as soft coding [24], sparse coding [5, 10, 37], building a discriminative dictionary [13, 26, 43] and discovering good pooling strategies [8] and receptive fields [3, 12].



Supervised Kernel Descriptors (SKDES)
Learning Process

1. Represent an image feature by encoding raw features with a dictionary $\mathbf{H}$ and a compact kernel transformation $\mathbf{A}$:
$$\mathbf{f}_I = \bigcup_R \left[\mathrm{op}_{h=1}^{|R|}\, \mathbf{g}(\mathbf{k}_h; \mathbf{H}, \mathbf{A})\right].$$
2. Plug the image feature into a large margin classification objective:
$E(\mathbf{A}) = \frac{1}{|\mathcal{I}|}\sum_{I\in\mathcal{I}} \mathcal{L}(\mathbf{f}_I, y_I) + \lambda Reg(\mathbf{A})$.
3. Optimize the objective by the regularized dual averaging (RDA):
$$\mathbf{A}^* = \arg\min_{\mathbf{A}} E(\mathbf{A}).$$

Figure 1. Left: The flowchart of the image classification pipeline with our supervised kernel descriptors (SKDES). Right: The learning process of SKDES, detailed in Sec. 3.

Nevertheless, most work keeps the low level descriptors as hand-crafted features, such as HOG [6] or SIFT [25]. As elaborated by [27, 36], the selection of raw descriptors is also an essential factor for achieving good performance in recognition tasks as the error at the beginning may propagate to latter stages. In this work, we focus on learning discriminative patch descriptors by exploiting image label information for improving racognition accuracy.

1.1. Related work

In recent years, the research over designing local descriptors has attracted much attention since the success of SIFT and HOG in retrieval, detection and recognition. In terms of the orientation histogram, a bunch of data-driven descriptors are proposed. PCA-SIFT [15] performed principal component analysis (PCA) on the gradient patches, which helps to reduce the noise in the descriptors. Dikmen *et al.* [7] used spherical k-means to get dense data-driven filters for local descriptors and showed better recognition accuracy than HOG on the PASCAL challenge. Nevertheless, these works did not obtain a discriminative subspace which might potentially improve the results.

Additionally, some works tried to adopt supervised techniques such as the linear discriminant analysis (LDA), the local discriminant embedding (LDE) to pursue a set of projections that best separate descriptors of different classes. Hua *et al.* [4] proposed to learn a linear transformation for SIFT using LDA and showed better results than SIFT on local feature matching problems. Winder [39] broke the design of local patch descriptors into filtering, pooling and normalizing to produce multiple settings of the descriptors and used Powell's multidimensional direction set method to learn the adjustable parameters in the descriptors by maximizing the receiver operating characteristic (ROC) area. Philbin *et al.* [31] tried to learn a non-linear transformation with deep networks by minimizing a margin-based cost function and presented impressive results on object retrieval tasks. Simonyan *et al.* [35] provided a supervised compact patch descriptor by solving convex max-margin problems and utilized a sparse and low-rank regularization to conduct pooling region selection and supervised dimension reduction.

However, these supervised approaches must have the label information associated with each pair of image patches, i.e., a matched pair or a mismatched pair. While in image recognition tasks, the label is connected with images, not patches. How to learn discriminative low level descriptors for image recognition is rarely addressed.

Another way to learn an image representation from low-level to image-level is through deep learning, which currently needs to build a unified hierarchical model for training. Convolutional neural networks [19] constructed a feed-forward network to learn multiple layers of nonlinear features. The parameters of the entire network, including a final layer for recognition, are jointly optimized using the back-propagation algorithm. Most recently, Krizhevsky *et al.* [16] applied a large, deep convolutional neural network with the hidden unit dropout regularization over the ImageNet recognition challenges[1] and obtained the best performance. This work induced a large amount of hidden units, thus is time consuming and requires huge computational power currently. Recently, to avoid labeling training data, deep learning methods utilized unsupervised fashion such as convolutional deep belief networks [20], convolutional sparse coding [14] and deconvolutional networks [44]. While in most cases like deep belief networks, the results can be further improved by exploiting supervised label information.

In this work, we focus on using image level label information to guide the design of low level features by taking advantage of kernel descriptors (KDES) [1]. In KDES, Bo *et al.* showed the matching of orientation histograms, such as SIFT, is equivalent to a linear kernel. Based on such a view, it provides a unified way to generate low-level patch features from multiple pixel attributes (gradient, color etc.). This descriptor achieves state-of-the-art results over many tasks such as scene labeling [32] and video classification [28].

While kernel descriptors are good at representing raw pixel features for recognition, it generates very high dimensional features because of the kernel trick for explicit representation and Kronecker production for combining multiple features. Though the original work applied the kernel principal component analysis (KPCA) to learn a compact representation, it loses the discriminative ability when dropping to a relatively low dimensional space. In our work, a supervised model, the large margin nearest neighbor (LMNN) criterion [38], is investigated to learn low dimensional patch-level kernel descriptors which we call supervised kernel descriptors (SKDES). Based on SKDES, we show the performance in many recognition tasks is boosted, especially at low dimensional cases.

2. Preliminaries

Kernel descriptors [1] highlight the kernel view of orientation histograms and show that descriptors like SIFT and HOG are a particular type of match kernels over patches. Specifically, let $\theta(z)$ and $m(z)$ be the orientation and magnitude of the image gradient at a pixel z. The gradient orientation of each pixel is discretized into a d-dimensional vector $\mathbf{h}(z) = [h_1(z)\ h_2(z) \cdots h_d(z)]^T$. A patch is then represented by a histogram of oriented gradients, aggregated over the pixels,

$$F_h(P) = \sum\nolimits_{z \in P} \tilde{m}(z)\mathbf{h}(z), \tag{1}$$

where $\tilde{m}(z) = \frac{m(z)}{\sqrt{\sum_{z \in P}(m(z))^2 + \epsilon_g}}$ is the normalized gradient magnitude, with ϵ_g a small constant to avoid zero value of the denominator.

In image recognition, the similarity of image patches P, Q in object recognition can be computed using an inner product in the feature map $F_h(P)$,

$$\begin{aligned} K_h(P, Q) &= F_h^T(P)F_h(Q) \\ &= \sum_{z \in P} \sum_{z' \in Q} \tilde{m}(z)\tilde{m}(z')\mathbf{h}(z)^T\mathbf{h}(z'). \end{aligned} \tag{2}$$

2.1. Kernel descriptors

In KDES, the linear kernel of orientation histogram in Eqn. (2) is generalized. Let $\tilde{m}(z)\tilde{m}(z') = k_{\tilde{m}}(z, z')$ and $\mathbf{h}(z)^T\mathbf{h}(z') = k_h(z, z')$, then the similarity between oriented gradients can be viewed in a kernel form,

$$K_h(P, Q) = \sum_{z \in P} \sum_{z' \in Q} k_{\tilde{m}}(z, z')k_h(z, z'). \tag{3}$$

Furthermore, the kernel can be generalized to capture the positions of pixels,

$$K_{grad}(P, Q) = \sum_{z \in P} \sum_{z' \in Q} k_{\tilde{m}}(z, z')k_h(z, z')k_p(z, z').$$

Rewriting the kernels as inner products, $k_h(z, z') = \phi_o(\theta(z))^T\phi_o(\theta(z'))$ and $k_p(z, z') = \phi_p(z)^T\phi_p(z')$. Then,

[1] http://www.image-net.org/challenges/LSVRC/2012

the patch feature can be viewed as a kernel descriptor, $F_{grad}(P) = \sum_{z \in P} \tilde{m}(z)\phi_o(\theta(z)) \otimes \phi_p(z)$, where $\otimes$ is the Kronecker product.

2.2. Compact kernel descriptors

To make the computation efficient and the representation compact, a kernel dimension reduction approach is presented in [1]. A straightforward way is to sample sufficient image patches from the training images and perform KPCA, which is however quite time-consuming and becomes computationally infeasible when the number of patches is very large. Instead, the dimension reduction is performed on the combinations of a set of uniformly and densely sampled sufficient basis vectors.

Given a set of densely sampled basis vectors, such as the sampled orientations $\{\phi_o(x_i)\}_{i=1}^{d_o}$ and the sampled position vectors $\{\phi_p(y_j)\}_{j=1}^{d_p}$, the gradient kernel descriptors are approximated in the form where the *t-th* component is written as follows,

$$\bar{F}^t_{grad}(P) = \sum_{i=1}^{d_o} \sum_{j=1}^{d_p} \alpha^t_{ij} \{ \sum_{z \in P} \tilde{m}(z) k_o(\theta(z), x_i) k_p(z, y_j) \}.$$

Here $\{\alpha^t_{ij}\}$ forms the t_{th} kernel principal component that is learned over the joint basis vectors, $\{\phi_o(x_1) \otimes \phi_p(y_1), \cdots, \phi_o(x_{d_o}) \otimes \phi_p(y_{d_p})\}$. Precisely, in KDES $\{\alpha^t_{ij}\}$ are learnt through KPCA: $\mathbf{K}_c \boldsymbol{\alpha}^t = \lambda^t \boldsymbol{\alpha}^t$, where $\mathbf{K}_c$ is a centralized kernel matrix which is computed from the joint basis vectors.

In this process, the kernel descriptors are explicitly represented through the kernel trick and Kronecker product, and finally compressed by using KPCA. This largely reduces the quantization error brought by some hand-crafted descriptors, such as SIFT with 8 orientation bins.

3. Supervised kernel descriptors

Learning strategy. We aim to learn compact kernel descriptors by exploiting the image labels, i.e., learning $\{\alpha^t_{ij}\}$ from the training images. Let us rewrite $\bar{F}^t_{grad}(P)$ in a vector form, $\bar{F}^t_{grad}(P) = (\boldsymbol{\alpha}^t)^T \mathbf{k}$, where $\mathbf{k} = \sum_z \tilde{m}(z)[k_o(\theta(z), x_1)k_p(z, y_1), .., k_o(\theta(z), x_{d_o})k_p(z, y_{d_p})]$. Denote $\mathbf{A} = [\boldsymbol{\alpha}^1, \cdots, \boldsymbol{\alpha}^D]$. Then generally, the patch feature can be written as $F(P) = \mathbf{A}^T \mathbf{k}$, where $\mathbf{A}$ is called a kernel transform which is a $D \times D$ matrix and D is the dimensionality of a kernel descriptor $\mathbf{k}$.

Pooling the patch features in region R_s together yields the feature over a region,

$$\mathbf{f}_{R_s} = \text{op}_{h=1}^{|R_s|}\, \mathbf{g}(\mathbf{H}, \mathbf{A}^T \mathbf{k}_h), \tag{4}$$

where $|R_s|$ is the patch feature number inside the region R_s, op is a pooling operator, *e.g.*, max or average, $\mathbf{g}(\cdot)$ is an encoding vector-valued operator, and $\mathbf{H} = [\mathbf{h}_1, \cdots, \mathbf{h}_n]$ is a dictionary. Concatenating the region features together forms the feature at image level, $\mathbf{f}_I = \bigcup_{s=1}^{RN} [\mathbf{f}_{R_s}] \triangleq \varphi(\mathbf{H}, \mathbf{A}; I)$, where RN is the region number and I is an image that can be represented from the kernel features $\mathbf{k}$.

The goal we are interested in is to find the compact kernel transform $\mathbf{A}$ through a supervised technique, the large margin nearest neighbor criterion [38] specifically. A sample and its target neighbors (one of its k-nearest neighbors that share the same label) are set to be close while other samples with different labels are pushed farther than the k_{th} target neighbor by a large margin. Two constraints are introduced, the first term is a large margin penalty that penalizes small distances between each input and all other inputs that do not share the same label, while the second term penalizes large distances between each input and its target neighbors. In addition, to avoid over-fitting issues and make the descriptor compact, we induce a rank regularization term to control the complexity of the model. In sum, the objective is formulated as,

$$\begin{aligned} \min_{\mathbf{A}} E(\mathbf{A}) = & \sum\nolimits_i \sum\nolimits_{jl} \eta_{ij}(1 - \delta_{il})[1 + d_{ij} - d_{il}]_+ \\ & + \lambda \sum\nolimits_{ij} \eta_{ij} d_{ij} + \lambda_* \|\mathbf{A}\|_*. \end{aligned} \tag{5}$$

Here the $[x]_+ = \max\{x, 0\}$ indicates the hinge loss. $\eta_{ij} \in \{0, 1\}$ and $\eta_{ij} = 1$ indicates that $\mathbf{f}_j$ is the target neighbor of $\mathbf{f}_i$. $\delta_{il} \in \{0, 1\}$ and $\delta_{il} = 0$ means $\mathbf{f}_l$ has a different label from $\mathbf{f}_i$. $d_{ij} = \|\mathbf{f}_i - \mathbf{f}_j\|_2^2 = \|\varphi(\mathbf{H}, \mathbf{A}; I_i) - \varphi(\mathbf{H}, \mathbf{A}; I_j)\|_2^2$. The nuclear norm, $\|\mathbf{A}\|_* = \text{Tr}(\sqrt{\mathbf{A}^T \mathbf{A}}) = \sum_{i=1}^{D} \sigma_i$, where σ_i is the *i-th* singular value of the matrix, performs a convex surrogate of $rank(\mathbf{A})$ which encourages low rank of the transformation. λ_* controls the trade-off between the model complexity and empirical training loss, the larger λ_* is, the smaller the intrinsic dimensionality of feature vectors will be.

Image features by reconstruction. We assume the encoding function $\mathbf{g}$ in Eqn. (4) encodes a patch feature $\mathbf{A}^T \mathbf{k}_h$ with a pre-computed dictionary through ridge regression which is defined as follows,

$$\mathbf{c}_h^* = \arg\min\nolimits_{\mathbf{c}_h} \|\mathbf{A}^T \mathbf{k}_h - \mathbf{H} \mathbf{c}_h\|_2^2 + \mu \|\mathbf{c}_h\|_2^2. \tag{6}$$

Then the code $\mathbf{c}_h$ has a closed-form solution respecting the patch feature and the dictionary as follows,

$$\mathbf{c}_h^* = (\mathbf{H}^T \mathbf{H} + \mu \mathbf{I})^{-1} \mathbf{H}^T (\mathbf{A}^T \mathbf{k}_h), \tag{7}$$

where $\mathbf{I}$ is the identity matrix. We let $\mu > 0$, which makes the matrix $\mathbf{H}^T \mathbf{H} + \mu \mathbf{I}$ positive definite. The dictionary $\mathbf{H}$ is defined in the kernel space, *i.e.*, formed after the kernel transform. Generally, it is formed by linear clustering the features. Specifically, given the set of $\tilde{p}$ patch level kernel features $\mathbf{F}$ (one column is a kernel descriptor $\mathbf{k}$), the dictionary of size n can be represented as $\mathbf{H} = (\mathbf{A}^T \mathbf{F}) \mathbf{Z}^T$, where $\mathbf{Z}$ is a combination matrix of size $n \times \tilde{p}$ which clusters n words out of $\tilde{p}$ patches.

With the encoded patch features, supposing the spatial pooling operation op in Eqn. (4) is an average pooling operation, and the image feature is concatenated from different regions, an image feature can be represented as: $\mathbf{f}_I = \bigcup_{s=1}^{RN}\left[\frac{1}{|R_s|}\sum_{h=1}^{|R_s|}\mathbf{c}_h\right]$. Plugging the image feature $\mathbf{f}_I$ into the objective (Eqn. (5)), we get a formula for the distance between two images i and j,

$$\begin{aligned} d_{ij} &= \sum_{s=1}^{RN}(\mathbf{k}_{si}-\mathbf{k}_{sj})^T\mathbf{L}\mathbf{L}^T(\mathbf{k}_{si}-\mathbf{k}_{sj}), \\ &\text{where } \mathbf{L} = \mathbf{A}\mathbf{A}^T\mathbf{F}\mathbf{Z}^T(\mathbf{Z}\mathbf{F}^T\mathbf{A}\mathbf{A}^T\mathbf{F}\mathbf{Z}^T+\mu\mathbf{I})^{-1}, \\ \mathbf{k}_{si} &= \frac{1}{|R_s|}\sum_{h=1}^{|R_s|}\mathbf{k}_{hi}, \end{aligned} \tag{8}$$

where $\mathbf{k}_{hi}$ is the *h-th* patch inside region R_s of the *i-th* image. Therefore, the optimization problem is turned to computing the gradient of Eqn. (5) regarding $\mathbf{A}$ and $\mathbf{Z}$, which can be reduced to computing the gradient of d_{ij} w.r.t. $\mathbf{A}$ and $\mathbf{Z}$.

4. Optimization

Directly applying the gradient decent algorithm regarding the optimizing parameters is difficult, because of the complexity from computing the gradient of high order matrix and the inverse in Eqn. (8). Additionally, the objective is highly non-convex which leads to local optimal solutions. Later, we overcome the difficulties by simplifying the problem into a convex one.

4.1. Convex reformulation

From Eqn. (8), the image features are computed through a linear mapping $\mathbf{L}$ from the pooled kernel descriptors, and the distance can be written in the following form, $d_{ij} = \sum_{s=1}^{RN}(\mathbf{k}_{si}-\mathbf{k}_{sj})^T\mathbf{M}(\mathbf{k}_{si}-\mathbf{k}_{sj})$ by setting $\mathbf{M} = \mathbf{L}\mathbf{L}^T$ which is known as the Mahalanobis matrix.

It can be easily prove that the first two penalizations in the objective (Eqn. (5)) is convex regarding $\mathbf{M}$, which is irrelevant to the selection of $\mathbf{F}$, and we will later show in Sec. 4.3 that our descriptors can be represented using the learned matrix $\mathbf{M}$.

Moreover, the low-rank regularization on $\mathbf{A}$ can be performed on $\mathbf{M}$ through the following demonstration. It is known that $rank(\mathbf{M}) = rank(\mathbf{L}\mathbf{L}^T) = rank(\mathbf{L})$. Let $\mathbf{L} = \mathbf{A}\mathbf{A}^\mathbf{T}\mathbf{B}$ (from Eqn. (8)). As $\mathbf{B}$ is a $D \times n$ matrix, we further know $rank(\mathbf{B}) = D$ by assuming that the dictionary $\mathbf{H}$ for coding is over-complete, which is always the case for the BOW pipeline. Hence $rank(\mathbf{L}) = rank(\mathbf{A}\mathbf{A}^\mathbf{T}\mathbf{B}) = rank(\mathbf{A}\mathbf{A}^\mathbf{T}) = rank(\mathbf{A})$, which leads to $rank(\mathbf{M}) = rank(\mathbf{A})$. Thus the rank constraint on $\mathbf{A}$ can be equivalently transformed into a rank constraint on $\mathbf{M}$. While for a semi-definite matrix, $\|\mathbf{M}\|_* = \text{Tr}(\mathbf{M})$, thus the trace of $\mathbf{M}$ can be used for the rank constraint of $\mathbf{M}$. Therefore, our optimization can be performed over the convex cone of positive semi-definite matrices $\mathbf{M}$. Thus, the convex version of our objective is formulated as follows,

$$\begin{aligned} \min_{\mathbf{M}} \quad & E(\mathbf{M}) = \sum_i \sum_{jl} \eta_{ij}(1-\delta_{il})[1+d_{ij}-d_{il}]_+ \\ & \qquad + \lambda \sum_{ij} \eta_{ij} d_{ij} + \lambda_* \text{Tr}(\mathbf{M}), \\ \text{s.t.} \quad & \mathbf{M} \succeq 0. \end{aligned} \tag{9}$$

where $d_{ij} = \sum_{s=1}^{RN}(\mathbf{k}_{si}-\mathbf{k}_{sj})^T\mathbf{M}(\mathbf{k}_{si}-\mathbf{k}_{sj})$ and $\mathbf{M} \succeq 0$ indicates that $\mathbf{M}$ is a semi-definite matrix.

4.2. Regularized stochastic learning

The corresponding objective yields a large problem as the complexity of each iteration increases in an order of $O(ND_I k_p k_n)$, where N is the training samples, D_I is the dimensionality of image features, k_p is the defined target neighbor number and k_n is the average number of activated negative examples (examples having different labels but smaller distances than the target neighbors) which is normally equal to k_p. Typically, this is in an order of $10^8 \sim 10^9$ in a dataset with thousands of images (Sec. 5), which makes general batch optimization strategies failed for a common PC. As indicated by [35], this problem can be handled by using online optimization methods. We adopt the recent developed regularized dual averaging (RDA) from Xiao [41], which is generic and applicable to non-smooth losses like the hinge loss in our case.

RDA is a stochastic proximal gradient method effective for problems of the form,

$$\min_{\mathbf{w}} \frac{1}{T}\sum_{t=1}^{T} f(\mathbf{w},\mathbf{z}_t) + R(\mathbf{w}), \tag{10}$$

where $\mathbf{w}$ is the weight vector to be learned, $\mathbf{z}_t$ is the t_{th} available training sample (pair), and $f(\mathbf{w},\mathbf{z}_t)$ is a convex loss, and $R(\mathbf{w})$ is a convex regularization term. Compared to the statistical gradient decent (SGD), it uses aggressive thresholding, thus produces solutions that are stronger regularized (in the L_1 case, with higher sparsity). A detailed description of RDA can be found in [41], and here we provide a brief overview.

At iteration t, RDA uses the loss subgradient $\mathbf{g}_t \in \partial f(\mathbf{w}_t,\mathbf{z}_t)$ to perform the update,

$$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w}}\left\{\langle \overline{\mathbf{g}}_t, \mathbf{w}\rangle + R(\mathbf{w}) + \frac{\beta_t}{t}h(\mathbf{w})\right\}, \tag{11}$$

where $\overline{\mathbf{g}}_t = \frac{1}{t}\sum_i^t \mathbf{g}_i$ is the average subgradient. $h(\mathbf{w})$ is a strongly convex function (like L_2 norm) such that $\arg\min_{\mathbf{w}} h(\mathbf{w})$ also minimizes $R(\mathbf{w})$, and β_t is *t-th* number in a specially chosen non-negative non-decreasing sequence. If the regularization $R(\mathbf{w})$ is not strongly convex (as the trace in our case), one can set $h(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2, \beta_t = \gamma\sqrt{t}, \gamma > 0$ to obtain a solution. In our case specifically, we define the $f(\mathbf{w}_t,\mathbf{z}_t)$ at iteration t as the loss (Eqn. (9)) brought by the added *t-th* example using $\mathbf{M}_t$, and the derived specific form of the RDA update step for $\mathbf{M}$ is,

$$\mathbf{M}_{t+1} = \prod \left(-\frac{\sqrt{t}}{\gamma} (\overline{\mathbf{G}}_t + \lambda_* \mathbf{I}) \right), \qquad (12)$$

where $\prod$ is the projection onto the cone of positive semi-definite matrix computed by cropping negative eigenvalues in the eigen-decomposition. $\overline{\mathbf{G}}_t = \frac{1}{t} \sum_{i=1}^{t} \mathbf{G}_i$ is the average subgradient of the corresponding loss for inner-class distances and inter-class distances, which can be inferenced as follows,

$$\begin{aligned} \mathbf{G}_i = & \sum\nolimits_{ij} \eta_{ij} \sum\nolimits_{s=1}^{RN} (\mathbf{k}_{si} - \mathbf{k}_{sj})(\mathbf{k}_{si} - \mathbf{k}_{sj})^T \qquad (13) \\ & + \sum\nolimits_{ijl} \eta_{ij} (1 - \delta_{il}) [\sum\nolimits_{s=1}^{RN} ((\mathbf{k}_{si} - \mathbf{k}_{sj})(\mathbf{k}_{si} - \mathbf{k}_{sj})^T \\ & - (\mathbf{k}_{si} - \mathbf{k}_{sl})(\mathbf{k}_{si} - \mathbf{k}_{sl})^T)] h'(1 + d_{ij} - d_{il})), \end{aligned}$$

where we use the smooth hinge loss, $h(x) = \frac{1}{b} \log(1 + e^{bx})$, for computing the subgradient of hinge loss at 0, and we set $b = 10$. In our experiments, the online optimization method is sufficient for a large dataset, while for a smaller dataset that contains hundreds of training images, we simply run RDA for several rounds to get a converged solution.

4.3. Kernel of kernel

Having $\mathbf{M}$ at hand, we avoid the difficulty to recover $\mathbf{A}$ and $\mathbf{Z}$ in Eqn. (8) by utilizing the learned distances to represent kernel similarity of encoded patches descriptors through the efficient match kernel method (EMK) [2]. In practice, we construct the RBF kernel between a pair of encoded patch features in Eqn. (7) with the learned linear mapping as: $k_{\mathbf{M}}(\mathbf{c}_x, \mathbf{c}_y) = \exp(-\gamma_m (\mathbf{c}_x - \mathbf{c}_y)^T (\mathbf{c}_x - \mathbf{c}_y)) = \exp(-\gamma_m (\mathbf{k}_x - \mathbf{k}_y)^T \mathbf{M} (\mathbf{k}_x - \mathbf{k}_y))$. With the kernel between patches features, an image feature can be later represented by the EMK as: $\mathbf{f}_I = \bigcup_{s=1}^{RN} \left[\frac{1}{|R_s|} \mathbf{G} \sum_{m \in R_s} k_{\mathbf{M}}(\mathbf{c}_m, \mathbf{C}) \right]$ where $\mathbf{C}$ is a set of patch level basis vectors (a.k.a. dictionary) generated through singular value decomposition (CKSVD). $\mathbf{G}$ is defined as, $\mathbf{G}^T \mathbf{G} = (k_{\mathbf{M}}(\mathbf{C}, \mathbf{C}))^{-1}$. We refer readers to [2] for more details due to the limitated space.

Through this kernel view, the final supervised kernel descriptors (SKDES) $\mathbf{f}$ can be represented efficiently by decomposing the $\mathbf{M}$ into $\hat{\mathbf{L}} \hat{\mathbf{L}}^T$, *i.e.*, $\mathbf{f} = \hat{\mathbf{L}}^T \mathbf{k}$, where $\hat{\mathbf{L}}$ is injective (a column full rank matrix) with rank of d, $d \ll D$, which largely reduces the dimensionality of the kernel descriptors.

5. Experiments

In this section, we evaluate the SKDES in terms of different experimental settings. To demonstrate the generalization of our approach, we evaluate our approach over datasets from different recognition tasks such as scene and object recognition. Specifically, we conduct experiments on UIUC sport, Scene 15, Caltech 101 and provide extensive comparisons in terms of accuracy reported by state-of-the-art methods.

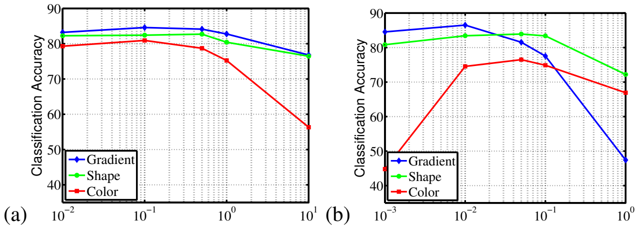


Figure 2. Parameters investigation for different kinds of descriptors. (a) Accuracy w.r.t. the regularization trade-off λ_*. (b) Accuracy w.r.t. the γ_m of the efficient match kernel (EMK).

5.1. Parameter setting

In our experiments, for constructing raw kernel descriptors, we adopt the code provided by Bo *et al.*[2] We use the RBF Kernel and the same parameters as the KDES [1] regarding γ_p and γ_h in Sec. 2.1. We set the patch size to 16×16, the sampling step length to 8 pixels and rescale the maximum length of images to 300 pixels. We first use KPCA to drop the raw KDES into 1000 dimensions. We set the default codebook size (number of words in the dictionary $\mathbf{C}$ of Sec. 4.3) to 2000. We use the spatial pyramid [18] for pooling, set the pyramid level to 1×1, 2×2, 3×3 and 4×4 and weight the cell s at layer l to $\frac{1}{l}$ for measuring the importance of the features in cell s. We use LIBlinear[3] for classification and adopt the proposed three kinds of KDES from [1], *i.e.*, gradient, color and shape descriptors. We evaluate the performances of each descriptor and the combination of the three kernel descriptors by simply concatenating the image-level feature vectors.

For SKDES, we set the number of target neighbors in Eqn. (9) to 4, λ in Eqn. (9) to 0.5 and we investigate two influential parameters which are λ_* for the rank regularization and γ_m for efficient match kernel over a validation image set. We enumerate the parameters over the range from 10^{-3} to 10^2, and the results from the UIUC sport dataset are showed in Fig. 2 for different descriptors. We choose optimal λ_* and γ_m for each dataset. For getting the feature with a desired dimensionality, we propose different parameters and select the model with the best accuracy over a validation set among the ones whose dimensionality are no higher than the request dimensionality.

5.2. Performance w.r.t. dimensionality

To demonstrate that our supervised method can learn more compact low level descriptors, we tested the performance of our supervised descriptor under different dimensionality over UIUC sport. Fig. 3 shows the evaluation results of our supervised dimensionality reduction using the compact mapping $\hat{\mathbf{L}}$ from Sec. 4.3, on the shape descriptors. We trained a dictionary of size 800 and used a single grid for learning. This is for speed consideration and our experi-

[2] http://www.cs.washington.edu/robotics/projects/kdes/

[3] http://www.csie.ntu.edu.tw/ cjlin/liblinear/

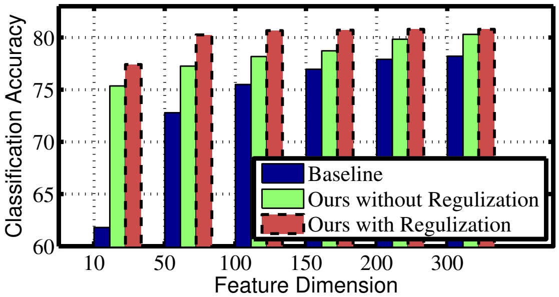


Figure 3. Classification accuracy comparison between the baseline KDES [1] and our approach in dimension reduction of patch descriptors.

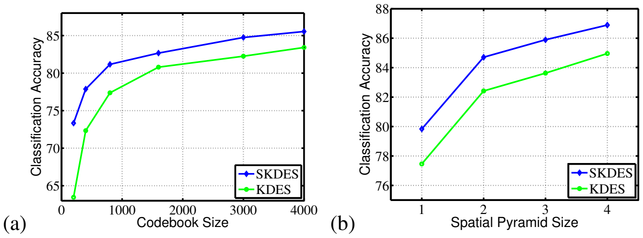


Figure 4. (a) Testing accuracy w.r.t. codebook size. (b) Testing accuracy w.r.t. pooling spatial pyramid size.

ments show the gain from supervised information is consistent between different codebooks and grid sizes in Sec. 5.3. The supervised scheme performs very well, as there is less drop in accuracy going from 300 dimensions to 10. With regularization, the optimal transform performs even better at a low dimensional space. In comparison, KDES [1] loses much accuracy (over 10%) when reducing to 10 dimensions. This shows the supervised strategy greatly improves storage and computational efficiency of the kernel descriptors.

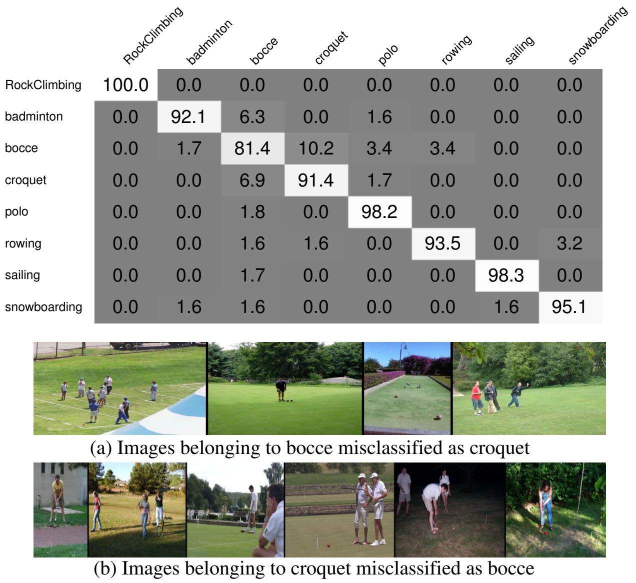
5.3. Performance w.r.t. codebook and pyramid size

To empirically justify the performance gain from the supervised information, we trained dictionaries of multiple codebook sizes, constructed spatial pooling pyramids of different levels and compared the accuracy produced from KDES and SKDES in Fig. 4 with shape descriptors. As can be observed, the gain from SKDES is consistently obtained over different codebook sizes and different spatial pyramid sizes, which means that the supervised information is complementary to other stages of the classification pipeline.

Another observation is that the performance is not saturated at the largest dictionary we had tested, and by combining with word selection strategies, such as the one in [12], a compact dictionary can be selected from orginal large one, which yields further improvements with fewer words.

5.4. Performance over diverse benchmarks

UIUC sport. UIUC sport [22] contains 1579 images with 8 sport event categories. Images are divided into easy and medium according to the human subject judgment. Using the standard setting, for each category, we randomly selected 70 images for training, sampled 60 images for testing and ran 5 rounds for a confident accuracy. To obtain optimal results, we improved the codebook size to 3000. Table 1 shows the comparison results between SKDES with KDES and provides the results of other current works from their papers. We obtained the best results over this dataset. Additionally, we also visualized the confusion matrix and wrongly classified examples in Fig. 5. We found the confusion images in the two classes have similar backgrounds and human postures. The only difference is whether a human is holding a stick or not. For a more robust representation, a semantic object region selection like [33] provided a reasonable extension.



	RockClimbing	badminton	bocce	croquet	polo	rowing	sailing	snowboarding
RockClimbing	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
badminton	0.0	92.1	6.3	0.0	1.6	0.0	0.0	0.0
bocce	0.0	1.7	81.4	10.2	3.4	3.4	0.0	0.0
croquet	0.0	0.0	6.9	91.4	1.7	0.0	0.0	0.0
polo	0.0	0.0	1.8	0.0	98.2	0.0	0.0	0.0
rowing	0.0	0.0	1.6	1.6	0.0	93.5	0.0	3.2
sailing	0.0	0.0	1.7	0.0	0.0	0.0	98.3	0.0
snowboarding	0.0	1.6	1.6	0.0	0.0	0.0	1.6	95.1

(a) Images belonging to bocce misclassified as croquet

(b) Images belonging to croquet misclassified as bocce

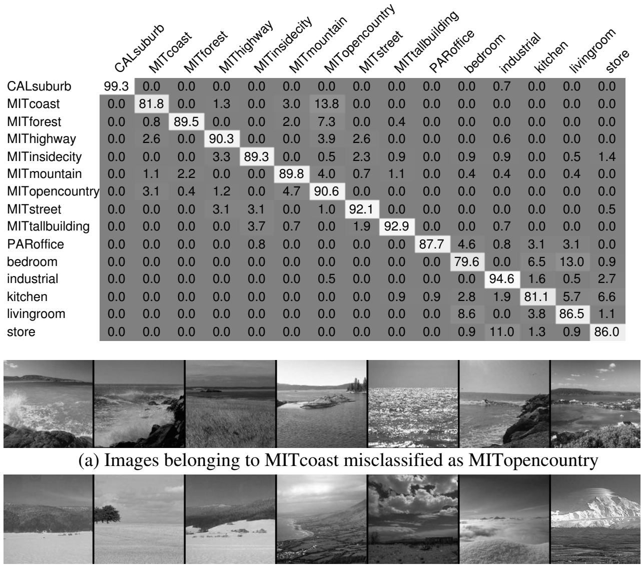
Figure 5. Up: The confusion matrix over UIUC sport dataset (%). Down: Some misclassified instances between two confusing classes.

Scene 15. Scene 15 [18] contains 15 categories and 4485 gray images in all, 200 to 400 images per category. The images contain not only indoor scenes, such as bedroom, kitchen, but also outdoor scenes, such as building and coun-

Method	Grad.	Color	Shape	Comb.
KDES [1]	85.7±0.7	72.9±1.4	83.5±1.2	89±0.4
SKDES	88.6±1.2	78.3±1.3	85.5±1.1	**91.0±0.8**

Method	Average Classification Rate (%)
Semantic Manifold [17]	83
CA-TM [29]	78
Local Soft Assign. [24]	84.56±1.5
LLC [37]	82.73±1.3
KSRSPM [9]	84.92±0.78
LScSPM [10]	85.31±0.51
OB [23]	76.4
HIK+OCSVM [40]	83.54±1.13
ScSPM [42]	82.74±1.46
SIFT+GGM [22]	73.4

Table 1. Performance comparison on UIUC sport dataset.



	CALsuburb	MITcoast	MITforest	MIThighway	MITinsidecity	MITmountain	MITopencountry	MITstreet	MITtallbuilding	PARoffice	bedroom	industrial	kitchen	livingroom	store
CALsuburb	99.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.7	0.0	0.0	0.0
MITcoast	0.0	81.8	0.0	1.3	0.0	3.0	13.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
MITforest	0.0	0.8	89.5	0.0	0.0	2.0	7.3	0.0	0.4	0.0	0.0	0.0	0.0	0.0	0.0
MIThighway	0.0	2.6	0.0	90.3	0.0	0.0	3.9	2.6	0.0	0.0	0.0	0.6	0.0	0.0	0.0
MITinsidecity	0.0	0.0	0.0	3.3	89.3	0.0	0.5	2.3	0.9	0.0	0.9	0.9	0.0	0.5	1.4
MITmountain	0.0	1.1	2.2	0.0	0.0	89.8	4.0	0.7	1.1	0.0	0.4	0.4	0.0	0.4	0.0
MITopencountry	0.0	3.1	0.4	1.2	0.0	4.7	90.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
MITstreet	0.0	0.0	0.0	3.1	3.1	0.0	1.0	92.1	0.0	0.0	0.0	0.0	0.0	0.0	0.5
MITtallbuilding	0.0	0.0	0.0	0.0	3.7	0.7	0.0	1.9	92.9	0.0	0.0	0.7	0.0	0.0	0.0
PARoffice	0.0	0.0	0.0	0.0	0.8	0.0	0.0	0.0	0.0	87.7	4.6	0.8	3.1	3.1	0.0
bedroom	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	79.6	0.0	6.5	13.0	0.9
industrial	0.0	0.0	0.0	0.0	0.0	0.0	0.5	0.0	0.0	0.0	0.0	94.6	1.6	0.5	2.7
kitchen	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.9	0.9	2.8	1.9	81.1	5.7	6.6
livingroom	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	8.6	0.0	3.8	86.5	1.1
store	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.9	11.0	1.3	0.9	86.0

(a) Images belonging to MITcoast misclassified as MITopencountry

(b) Images belonging to MITopencountry misclassified as MITcoast

Figure 6. Up: The confusion matrix on Scene 15 dataset (%). Down: Some misclassified instances between two confusing classes.

try. Following the standard setting, we randomly selected 100 images per class as training data, used the rest as test data and ran 5 rounds. We replicated the gray image into rgb channels to obtain the color descriptors. Table 2 lists our results and the performances published by other strategies. Moreover, the confusion matrix and misclassified images are showed in Fig. 6. As can be seen, the confusing images have a close spatial layout and similar texture information which makes them difficult to be classified. To our best knowledge, we achieved the 2_{nd} best results to date over this dataset. The winner LScSPM [10] uses sparse coding with a Laplacian similarity constraint between sparse codes. In the future, we will try to combine the SKDES with the sparsity and Laplacian constraint for feature encoding, which may possibly further improve the accuracy.

Method	Grad.	Color	Shape	Comb.
KDES [1]	83.5±1.4	76.2±0.8	80.9±0.4	87.7±0.3
SKDES	84.9±0.5	78.2±1.4	83.6±0.9	**88.7±0.7**

Method	Average Classification Rate (%)
DSS [34]	85.5±0.6
Semantic Manifold [17]	82.3
RBoW [30]	78.6±0.7
CA-TM [29]	82.5
Local Soft Assign. [24]	83.76±0.59
Feng *et al.* [8]	84.60
LLC [37]	81.53±0.65
KSRSPM [9]	83.68±0.61
LScSPM [10]	89.75±0.50
HIK+OCSVM [40]	84.00±0.46
ScSPM [42]	80.28±0.93
SIFT+GGM [22]	73.4
KSPM [18]	81.4±0.50

Table 2. Performance comparison on Scene 15 dataset.

Caltech 101. Caltech 101 [21] consists of 9,144 images in 101 object categories and one background category. The number of images per category varies from 31 to 800. Following the standard experimental setting, we trained classifiers on randomly selected 3060 images (30 images per class), tested on no more than 50 images per category and also conducted the experiment 5 times. Table 3 lists the average accuracy from KDES, SKDES and recently published results obtained from the original papers. Our result also ranks the 2_{nd} best results to date within the listed methods. The winner [8] proposed a geometrically discriminative pooling strategy that can also be regarded as a complementary strategy for our work.

5.5. Effectiveness of learned feature distances

We finally empirically show that our learned distances generate more discriminative features. In Fig. 7, we compared the RBF kernel matrix, between testing and training data of UIUC sport, generated from the image features in Sec. 3 with KDES (Fig. 7(a)) and SKDES (Fig. 7(b)) in a low dimensional space (50 dim) based on three kinds of descriptors. As can be seen, SKDES provides much more distinctive kernel similarity (much clearer diagonal structure of the matrix), which effectively pushes the images with different labels farther and pulls the images with the same label closer. In Fig. 7(c), we retrieve different numbers of nearest neighbors based on the two types of distances and show that our learned image distance yields more true neighborhoods, which benefits the classification.

6. Conclusion

In this paper, we proposed supervised kernel descriptors (SKDES) for local image patches, which is learned from image labels based on the large margin criterion with low-rank regularization and the widely-applied BOW image classification pipeline. Experiments over several public benchmarks show our SKDES outperforms the original KDES especially in low dimensional spaces, and it achieves competitive results compared to other state-of-the-art methods.

Method	Grad.	Color	Shape	Comb.
KDES [1]	75.2±0.4	66.2±0.8	70.3±1.4	77.4±0.6
SKDES	77.3±0.7	68.4±1.4	71.6±1.3	**79.2±0.6**

Method	Average Classification Rate (%)
Jia *et al.* [12]	75.3±0.70
SDL [13]	75.3±0.40
Adaptive Deconv. Net [44]	71.0±1
Boureau *et al.* [3]	77.3±0.6
LSAQ [24]	74.21±0.81
Feng *et al.* [8]	82.60
Local Soft Assign. [24]	74.21±0.81
LLC [37]	73.44
Lp-β (Multiple Kernel) [11]	77.7±0.3
ScSPM [42]	73.2±0.54
KSPM [18]	64.6±0.8

Table 3. Performance comparison on Caltech 101 dataset.

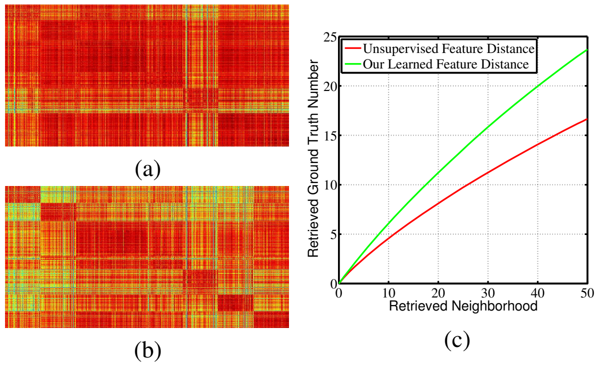


Figure 7. (a) The kernel matrix between training and testing image features from KDES. (The redder, the higher the kernel value is). (b) The kernel matrix from our SKDES, which yields a much clearer diagonal structure indicating the kernel similarity from SKDES is more discriminative. (c) The number of retrieved ground truth neighbors w.r.t. number of retrieved neighbors.

Acknowledgements

The research work of Peng Wang, Gang Zeng and Hongbin Zha is supported by National Nature Science Foundation of China (NSFC Grant) 61005037, 90920304 and 91120004, and National Basic Research Program of China (973 Program) 2011CB302202. Weiwei Xu is partially supported by NSFC 61272392 and the Open Project Program of the State Key Lab of CAD&CG (A1307).

References

[1] L. Bo, X. Ren, and D. Fox. Kernel descriptors for visual recognition. In *NIPS*, pages 244–252, 2010. 1, 2, 3, 5, 6, 7

[2] L. Bo and C. Sminchisescu. Efficient match kernel between sets of features for visual recognition. In *NIPS*, pages 135–143, 2009. 1, 5

[3] Y.-L. Boureau, N. L. Roux, F. Bach, J. Ponce, and Y. LeCun. Ask the locals: Multi-way local pooling for image recognition. In *ICCV*, pages 2651–2658, 2011. 1, 7

[4] M. Brown, G. Hua, and S. A. J. Winder. Discriminative learning of local image descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(1):43–57, 2011. 1

[5] A. Coates and A. Y. Ng. The importance of encoding versus training with sparse coding and vector quantization. In *ICML*, pages 921–928, 2011. 1

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages 886–893, 2005. 1

[7] M. Dikmen, D. Hoiem, and T. S. Huang. A data driven method for feature transformation. In *CVPR*, pages 3314–3321, 2012. 1

[8] J. Feng, B. Ni, Q. Tian, and S. Yan. Geometric p-norm feature pooling for image classification. In *CVPR*, pages 2697–2704, 2011. 1, 7

[9] S. Gao, I. W.-H. Tsang, and L.-T. Chia. Kernel sparse representation for image classification and face recognition. In *ECCV (4)*, pages 1–14, 2010. 6, 7

[10] S. Gao, I. W.-H. Tsang, L.-T. Chia, and P. Zhao. Local features are not lonely - laplacian sparse coding for image classification. In *CVPR*, pages 3555–3561, 2010. 1, 6, 7

[11] P. V. Gehler and S. Nowozin. On feature combination for multiclass object classification. In *ICCV*, pages 221–228, 2009. 7

[12] Y. Jia, C. Huang, and T. Darrell. Beyond spatial pyramids: Receptive field learning for pooled image features. In *CVPR*, pages 3370–3377, 2012. 1, 6, 7

[13] Z. Jiang, G. Zhang, and L. S. Davis. Submodular dictionary learning for sparse coding. In *CVPR*, pages 3418–3425, 2012. 1, 7

[14] K. Kavukcuoglu, P. Sermanet, Y.-L. Boureau, K. Gregor, M. Mathieu, and Y. LeCun. Learning convolutional feature hierarchies for visual recognition. In *NIPS*, pages 1090–1098, 2010. 2

[15] Y. Ke and R. Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. In *CVPR (2)*, pages 506–513, 2004. 1

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1106–1114, 2012. 2

[17] R. Kwitt, N. Vasconcelos, and N. Rasiwasia. Scene recognition on the semantic manifold. In *ECCV (4)*, pages 359–372, 2012. 6, 7

[18] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR (2)*, pages 2169–2178, 2006. 1, 5, 6, 7

[19] Y. LeCun, F. J. Huang, and L. Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *CVPR (2)*, pages 97–104, 2004. 2

[20] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *ICML*, page 77, 2009. 2

[21] F.-F. Li, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(4):594–611, 2006. 7

[22] L.-J. Li and F.-F. Li. What, where and who? classifying events by scene and object recognition. In *ICCV*, pages 1–8, 2007. 6, 7

[23] L.-J. Li, H. Su, E. P. Xing, and F.-F. Li. Object bank: A high-level image representation for scene classification & semantic feature sparsification. In *NIPS*, pages 1378–1386, 2010. 6

[24] L. Liu, L. Wang, and X. Liu. In defense of soft-assignment coding. In *ICCV*, pages 2486–2493, 2011. 1, 6, 7

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

[26] J. Mairal, F. Bach, and J. Ponce. Task-driven dictionary learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(4):791–804, 2012. 1

[27] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(10):1615–1630, 2005. 1

[28] P. Natarajan, S. Wu, S. N. P. Vitaladevuni, X. Zhuang, U. Park, R. Prasad, and P. Natarajan. Multi-channel shape-flow kernel descriptors for robust video event detection and retrieval. In *ECCV (2)*, pages 301–314, 2012. 2

[29] Z. Niu, G. Hua, X. Gao, and Q. Tian. Context aware topic model for scene recognition. In *CVPR*, pages 2743–2750, 2012. 6, 7

[30] S. N. Parizi, J. G. Oberlin, and P. F. Felzenszwalb. Reconfigurable models for scene recognition. In *CVPR*, pages 2775–2782, 2012. 7

[31] J. Philbin, M. Isard, J. Sivic, and A. Zisserman. Descriptor learning for efficient retrieval. In *ECCV (3)*, pages 677–691, 2010. 2

[32] X. Ren, L. Bo, and D. Fox. Rgb-(d) scene labeling: Features and algorithms. In *CVPR*, pages 2759–2766, 2012. 2

[33] O. Russakovsky, Y. Lin, K. Yu, and L. Fei-Fei. Object-centric spatial pooling for image classification. In *ECCV (2)*, pages 1–15, 2012. 6

[34] G. Sharma, F. Jurie, and C. Schmid. Discriminative spatial saliency for image classification. In *CVPR*, pages 3506–3513, 2012. 7

[35] K. Simonyan, A. Vedaldi, and A. Zisserman. Descriptor learning using convex optimisation. In *ECCV (1)*, pages 243–256, 2012. 2, 4

[36] A. Tamrakar, S. Ali, Q. Yu, J. Liu, O. Javed, A. Divakaran, H. Cheng, and H. S. Sawhney. Evaluation of low-level features and their combinations for complex event detection in open source videos. In *CVPR*, pages 3681–3688, 2012. 1

[37] J. Wang, J. Yang, K. Yu, F. Lv, T. S. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *CVPR*, pages 3360–3367, 2010. 1, 6, 7

[38] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10:207–244, 2009. 2, 3

[39] S. A. J. Winder and M. Brown. Learning local image descriptors. In *CVPR*, 2007. 2

[40] J. Wu and J. M. Rehg. Beyond the euclidean distance: Creating effective visual codebooks using the histogram intersection kernel. In *ICCV*, pages 630–637, 2009. 6, 7

[41] L. Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11:2543–2596, 2010. 4

[42] J. Yang, K. Yu, Y. Gong, and T. S. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, pages 1794–1801, 2009. 1, 6, 7

[43] J. Yang, K. Yu, and T. S. Huang. Supervised translation-invariant sparse coding. In *CVPR*, pages 3517–3524, 2010. 1

[44] M. D. Zeiler, G. W. Taylor, and R. Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In *ICCV*, pages 2018–2025, 2011. 2, 7