Front. Comput. Sci., 2015, 9(3): 383–391
DOI 10.1007/s11704-015-4182-7

RESEARCH ARTICLE

Efficient image representation for object recognition via pivots selection

Bojun XIE[1,2], Yi LIU (✉)[1], Hui ZHANG[1,2], Jian YU[1]

1 Beijing Key Lab of Traffic Data Analysis and Mining, School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China

2 Key Lab of Machine Learning and Computational Intelligence, College of Mathematics and Computer Science, Hebei University, Baoding 071000, China

Abstract Patch-level features are essential for achieving good performance in computer vision tasks. Besides well-known pre-defined patch-level descriptors such as scaleinvariant feature transform (SIFT) and histogram of oriented gradient (HOG), the kernel descriptor (KD) method [1] offers a new way to "grow-up" features from a match-kernel defined over image patch pairs using kernel principal component analysis (KPCA) and yields impressive results.

In this paper, we present efficient kernel descriptor (EKD) and efficient hierarchical kernel descriptor (EHKD), which are built upon incomplete Cholesky decomposition. EKD automatically selects a small number of pivot features for generating patch-level features to achieve better computational efficiency. EHKD recursively applies EKD to form image-level features layer-by-layer. Perhaps due to parsimony, we find surprisingly that the EKD and EHKD approaches achieved competitive results on several public datasets compared with other state-of-the-art methods, at an improved efficiency over KD.

Keywords efficient kernel descriptor, efficient hierarchical kernel descriptor, incomplete Cholesky decomposition, patch-level features, image-level features

1 Introduction

Designing good image features is a fundamental problem in computer vision. As a key component in image classification, indexing and retrieval systems, feature engineering is a challenging task since there are ubiquitous sources of image variations, e.g., scale/illumination changes and occlusion. Good image features should be robust to these changes and being as discriminative as possible.

As is noted in [2], researchers have proposed very powerful low-level image features based on gradient orientation histograms, e.g., scale-invariant feature transform (SIFT) [3] or histogram of oriented gradient (HOG) [4]. Although such features have achieved great successes in many tasks, it is shown that intermediate feature representations [5], such as Bag of Words (BoW) [6], is necessary for obtaining excellent performance for classifying a large number of scene categories. Standard BoW pipeline firstly extracts low-level features from images, then encoding them into a middle-level representation through an over-complete dictionary, finally an image is represented by the histogram of codewords occurrence frequencies. However, as BoW ignores information about the spatial organization of local features, the descriptive ability of this representation is not maximized. To overcome this drawback, the spatial pyramid matching (SPM) approach [7] adds spatial information to the model by pooling features over image sub-regions. However, the histogram in-

Received April 18, 2014; accepted June 24, 2014

E-mail: yiliu@bjtu.edu.cn

tersection kernel used in SPM does not naturally correspond to a low-dimensional image-level feature representation for linear SVM classification. As such, the computational complexity of this method is quadratic to the number of images since it is necessary to calculate the Gram matrix explicitly. To resolve this problem, the efficient match kernel (EMK) approach [8] constructs a low-dimensional feature space from the pairwise codewords similarities, and averages the codewords feature maps in this space for an image to obtain an image-level feature representation for classification. The kernel descriptor approach [1] extends the idea of EMK by deriving a patch-level feature representation from the similarities computed from pixel attributes in different ways. It employs kernel principal component analysis (KPCA) [9] to construct the feature space from the pairwise similarities between large numbers of joint basis vectors sampled from the support regions of gradient/shape/color + position attributes. Then, for each image patch, its feature representation can be obtained by quantifying its similarities with all the joint basis vectors. The key problem for the kernel descriptor approach is its high computational complexity. In fact, during the offline kernel principal component analysis (KPCA) step of [1], all the pairwise similarities between the joint basis vectors should be computed, which is very expensive. More importantly, in the online computation of the feature map for an image patch, its similarities with all joint basis vectors should be evaluated, which is not necessary. In the preliminary version of this work [10], we successfully addressed this problem via incomplete Cholesky decomposition instead of using KPCA. However, both this work [10] and the original kernel descriptor [1] used EMK algorithm [8] to generate the image-level feature representation. Because spatial information is lost in the match kernels [8], it limits the capacity of the image-level feature representations in [1, 10]. Recently, Wang et al. [11] propose supervised kernel descriptors. They employ image-level label information to guide the design of low-level features within the kernel descriptor approach. Although they achieved competitive results, the image-level features are learned from image labels based on the large margin criterion with low-rank regularization, which requires large numbers of labeled images and is not computationally efficient.

Another way to build an image-level representation from low-level features is deep learning, which constructs high-level feature representation hierarchically. For example, convolutional neural networks [12] learned multiple layers of nonlinear features using feed-forward architecture. Parameters in the network are jointly optimized using the back propagation algorithm. Similarly, deep belief nets (DBNs) [13,14] also learn a hierarchy of features one layer at a time, where features learned by the current layer become input for training features in the next layer. Most recently, Krizhevsky et al. [15] constructs a large, deep convolutional neural network for the ImageNet recognition challenge and obtains the best performance. This network contains a large number of hidden units, which is computationally demanding and requires a powerful multi-core architecture to run. Bo et al. [16] applies kernel descriptors recursively to aggregate low-level features into high-level features to obtain the hierarchical kernel descriptor representation. But like the kernel descriptor approach, KPCA is again used repeatedly in this work to reduce the dimensionality of feature representations, so it also suffers the problems of the kernel descriptor approach [1].

In this work, to achieve the flexibility of the data-driven kernel expansion approach [1] while alleviating the data-size depend computational complexity to an acceptable level, we propose efficient kernel descriptor (EKD) based on the incomplete Cholesky decomposition, which not only avoids exhaustively computation of all the pair-wise similarities in the offline stage, but also reduces the computational complexity in the online stage of evaluating the similarities of an image patch to a small set of pivot joint basis vectors. Moreover, we can flexibly control the computational complexity by adaptively tuning the rank of the incomplete Cholesky decomposition. Furthermore, inspired by [16], we also propose a new approach for generating the image-level feature representation via iterative incomplete Cholesky decomposition, which is termed as efficient hierarchical kernel descriptor (EHKD). We found that combined with linear SVMs, EHKD outperforms many state-of-the-art algorithms in this vein.

In the remaining part of this paper, we first describe the proposed EKD approach and compare it with the original kernel descriptor (KD) [1] in Section 2. Then, we introduce the EHKD approach and compare it with the hierarchical kernel descriptor (HKD) [16] in Section 3. Detailed experimental results and discussions are presented in Section 4. Finally, we conclude this paper in Section 5.

2 Efficient kernel descriptor

In this section, we introduce the technical details of the proposed efficient kernel descriptor and highlight its connection with the original kernel descriptor approach [1].

2.1 A brief introduction to the kernel descriptor approach

As described above, the kernel descriptor approach [1] employ KPCA to empirically estimate finite-dimensional feature maps based on a set of basis vectors sampled over the support regions of pixel attributes. For simplicity, we only introduce its key idea based on the gradient attribute.

First, define the gradient match kernel based on the gradient attribute of pixels:

$$K_{grad}(P, Q) = \sum_{z\in P}\sum_{z'\in Q} \tilde{m}_z \tilde{m}_{z'} k_o(\tilde{\theta}_z, \tilde{\theta}_{z'}) k_p(z, z'), \qquad (1)$$

where P and Q represent two different image patches, z is the 2D position of a pixel in the image patch, $\tilde{\theta}_z$ and $\tilde{m}_z$ are the orientation and magnitude of the intensity gradient at pixel z, k_o and k_p are the orientation and position RBF kernels, respectively.

Then, the low-dimensional feature map for image patch P can be computed by

$$\bar{F}^t_{grad}(P) = \sum_{i=1}^{d_o}\sum_{j=1}^{d_p} \alpha^t_{ij}\{\sum_{z\in P} \tilde{m}(z) k_o(\tilde{\theta}(z), \theta(x_i)) k_p(z, y_j)\}, \qquad (2)$$

where $\{\theta(x_i)\}_{i=1}^{d_o}$ and $\{y_j\}_{j=1}^{d_p}$ are the gradient orientation and position basis uniformly sampled from their respective support regions, d_o and d_p are the sample sizes of the corresponding basis vectors and α^t_{ij} are projection coefficients computed by applying KPCA to the set of joint basis vectors: $\{\phi_o(\theta(x_1)) \bigotimes \phi_p(y_1), \ldots, \phi_o(\theta(x_{d_o})) \bigotimes \phi_p(y_{d_p})\}$ ($\bigotimes$ is the Kronecker product).

The computational complexity of the feature map construction approach shown in Eq. (2) is high. First, we have to compute all the $(d_o d_p)^2$ pairwise similarities between the joint basis vectors to obtain the Gram matrix. Second, eigendecomposition of the matrix usually takes $O(d_o^3 d_p^3)$ time complexity in KPCA. Third, since the projection coefficients are not sparse, when online computing the feature map for the image patch P, we should perform summation over the kernel products $k_o(\cdot,\cdot)k_p(\cdot,\cdot)$ $d_o d_p$ times and all the joint basis vectors should be stored in memory.

2.2 Efficient kernel descriptor

To avoid this problem, we propose efficient kernel descriptor (EKD), which is based on the incomplete Cholesky decomposition of a Gram matrix [17, 18].

On the high level, this approach consists of two steps: i) compute the incomplete Cholesky decomposition [18] of the pairwise similarity matrix over the joint basis vectors. Denote the rank of this decomposition as M, usually we have: $M \ll N$, where $N = d_o d_p$ is the number of all joint basis vectors. The merits of our approach are two-fold. First, we only need to compute $O(MN)$ elements of the Gram matrix on demand. Second, performing the Cholesky decomposition of the matrix only has a $O(M^2N)$ time complexity rather than the brute-force $O(N^3)$ complexity of KPCA; ii) based on the M pivot joint basis vectors computed by the incomplete Cholesky decomposition, we generate the feature map for image patch P using only $O(M)$ computations, as described below.

2.2.1 Low-rank approximation of the Gram matrix

The positive semi-definite Gram matrix K can always be factorized as GG^{T}. The aim of incomplete Cholesky decomposition is to find a matrix $\tilde{G}$ of size $N \times M$, such that $\tilde{G}\tilde{G}^{\mathrm{T}}$ is a good approximation for K for small M.

During the computation of the incomplete Cholesky decomposition, we have actually select M pivot basis vectors such that all the pairwise similarities in K are approximated by the similarities from all the basis vectors to these pivots. Here, M is determined by the algorithm online, which is controlled by a parameter ε that specifies the pre-defined accuracy of the approximation. See [18] for a more detailed description about incomplete Cholesky decomposition.

2.2.2 Construction of the feature maps

Once we obtain $\tilde{G}$ of size $N \times M$, the new patch-level feature can be constructed based on $\tilde{G}$. Algorithm 1 details the steps for constructing efficient kernel descriptor.

Algorithm 1 The EKD algorithm for constructing the feature map of an image patch.

```
Input: $\tilde{G}$: output from incomplete Cholesky decomposition.
       $P$: index of the selected pivots (also output from incomplete Cholesky decomposition).
       $M$: the number of pivots.
       similarity: an $M$ dimensional array of similarities from an image patch to the pivots.
Output: $G2$: feature map of the image patch.
for $i \leftarrow 1$ to $M$ do
    //computing the value for the $i$-th dimension of the feature map $G2$;
    $G2(i) = [similarity(i) - \sum_{j=1}^{i-1} G2(j)G(P(i), j)]/G(P(i), i)$;
end
```

2.3 Discussions

Note that the M pivot basis vectors selected by the EKD algorithm can be seen as a set of non-linear feature extractors for a new image patch. This is because one obtains the feature map of the image patch by computing the similarities from the patch to the pivots. In other words, the proposed EKD algorithm essentially learned how to extract a patch-level feature from the kernel functions. This is fundamentally different with those procedurally-defined image descriptors such as SIFT and HOGs. It is also different with the original kernel descriptor approach [1], which has no built-in feature selection mechanism. As a result, the EKD algorithm is the most appropriate one to be viewed as "growing up" features from scratch.

3 Efficient hierarchical kernel descriptor

In this section, we firstly introduce hierarchical kernel descriptor [16], which is a hierarchical extension of the original kernel descriptor [1]. Then, following the same idea, we extend the proposed EKD to the corresponding image-level representation EHKD.

3.1 An introduction to hierarchical kernel descriptor

The original kernel descriptors method [1] is a procedure for generating patch-level features by operating on sets of pixels. The HKD approach [16] is virtually a recursive application of the kernel descriptors method to generate higher levels of image features, which aggregates spatial neighboring patch-level features iteratively to obtain features of larger image regions and finally of the entire image.

Here, the match kernels used to aggregate patch-level features have similar structure to those used to aggregate pixel attributes in the kernel descriptor approach:

$$K(\bar{P},\bar{Q}) = \sum_{A\in\bar{P}}\sum_{A'\in\bar{Q}} \tilde{W}_A \tilde{W}_{A'} k_F(F_A,F_{A'}) k_C(C_A,C_{A'}), \tag{3}$$

where $\bar{P}$, $\bar{Q}$ are sets of image patches and A, A' are patches in the corresponding sets; the image patch position kernel $k_C(C_A, C_{A'}) = \exp(-\gamma_C\|C_A - C_{A'}\|^2) = \Phi_C(C_A)^{\mathrm{T}}\Phi_C(C_{A'})$ describes the spatial relationship of two image patches, where C_A represents the center position of patch A (normalized to $[0, 1]$); the image patch kernel $k_F(F_A, F_{A'}) = \exp(-\gamma_F\|F_A - F_{A'}\|^2) = \Phi_F(F_A)^{\mathrm{T}}\Phi_F(F_{A'})$ gives the similarity of two patch-level features, where F_A represents gradient, shape or color kernel descriptors; the linear kernel $\tilde{W}_A\tilde{W}_{A'}$ weighs the contribution of each patch-level feature where $\tilde{W}_A = W_A/\sqrt{\sum_{A\in\bar{P}} W_A^2 + \varepsilon_h}$, ε_h is a small positive constant. For gradient kernel, W_A is the average of gradient magnitudes; for shape kernel, W_A is the average of standard deviations and W_A is always set to 1 for color kernel.

Evaluating Eq. (3) is time consuming. In order to extract compact low-dimensional features efficiently, HKD [16] used KPCA method for dimensionality reduction, as the kernel descriptor approach [1]. In particular, the inner product representation of two kernels is given by:

$$\begin{aligned}&k_F(F_A, F_{A'})k_C(C_A, C_{A'})\\&= [\Phi_F(F_A)\otimes\Phi_C(C_A)]^T[\Phi_F(F_{A'})\otimes\Phi_C(C_{A'})].\end{aligned} \tag{4}$$

Following [1], compact features are obtained by projecting the infinite-dimensional vector $\Phi_C(C_A)\bigotimes\Phi_F(F_A)$ to a set of joint basis vectors $\{\Phi_C(C_{X_1}),\ldots,\Phi_C(C_{X_{d_c}})\}\bigotimes\{\Phi_F(F_{Y_1}),\ldots,\Phi_F(F_{Y_{d_F}})\}$.

The first set $\{\Phi_C(C_{X_1}),\ldots,\Phi_C(C_{X_{d_c}})\}$ is generated by sampling C(position) on 5 × 5 regular grids. However, kernel descriptors obtained from the previous layer F_A(gradient/shape/color) are high-dimensional and it is infeasible to sample them on dense and uniform girds. Instead, the basis set $\{\Phi_F(F_{Y_1}),\ldots,\Phi_F(F_{Y_{d_F}})\}$ is generated by clustering patch-level gradient/shape/color features from training images via the K-means algorithm.

The dimensionality of the current layer kernel descriptors is the number of joint basis vectors $\{\Phi_F(F_{Y_1})\bigotimes\Phi_C(C_{X_1}),\ldots,\Phi_F(F_{Y_{d_F}})\bigotimes\Phi_C(C_{X_{d_c}})\}$, which is still not computationally manageable. To avoid the high dimensionality, [16] used KPCA to perform dimensionality reduction. Then, kernel descriptors of the current layer can be more compactly represented using the KPCA coefficients.

$$\bar{F}^t(\bar{P}) = \sum_{i=1}^{d_F}\sum_{j=1}^{d_c}\beta^t_{ij}\{\sum_{A\in P}\tilde{W}_A k_F(F_A, F_{Y_i})k_C(C_A, C_{X_j})\}, \tag{5}$$

where β^t_{ij} are also projection coefficients generated from KPCA. By recursively applying the above method, low-level features 'growing up' to form high-level features until the final image-level feature is obtained.

3.2 Efficient hierarchical kernel descriptor

From the procedure of constructing HKD, the KPCA step, which is computational intensive, has to be executed many times, making it not an appropriate solution. Although [16] gives a way to compute the eigenvectors of a huge Gram matrix, the intrinsic computational problem in the original kernel descriptor approach [1] is not avoided. To address this problem, the incomplete Cholesky decomposition of the Gram matrix is used again as in the EKD approach described above, but now it has been extended in the context of constructing the hierarchical feature representation.

The efficiency of EKD has been shown in Section 2.2, but it only generates patch-level feature. To obtain the image-level representation, the EMK method [8] is used. Here, we propose the EHKD to derive image-level features. As in HKD, the first problem is how to construct joint basis vectors from features of the previous layer. Like [16], set $\{\Phi_C(C_{X_1}),\ldots,\Phi_C(C_{X_{d_c}})\}$ can be generated by sampling X on 5×5 regular grids and set $\{\Phi_F(F_{Y_1}),\ldots,\Phi_F(F_{Y_{d_F}})\}$ can be generated by clustering the previous layer EKD features via the K-means algorithm. Then, joint basis vectors $\{\Phi_F(F_{Y_1})\bigotimes\Phi_C(C_{X_1}),\ldots,\Phi_F(F_{Y_{d_F}})\bigotimes\Phi_C(C_{X_{d_c}})\}$ are defined accordingly.

Next, EHKD can be generated in two steps, as discussed in Section 2.2. i) Compute the incomplete Cholesky decomposition of the pairwise similarity matrix over the joint basis vectors; ii) based on the selected pivots in step i), we generate the feature map for next layer. By recursively applying the above steps, we can obtain efficient kernel descriptors of increasingly larger image patches. Below, we introduce some technical details.

3.2.1 Selecting pivots from the Gram matrix

When the joint kernel is represented by the product of individual kernels of distinct features (suppose $K = K_F\bigotimes K_C$, where K_F and K_C represent the Gram matrix of set $\{\Phi_F(F_{Y_1}),\ldots,\Phi_F(F_{Y_{d_F}})\}$ and set $\{\Phi_C(C_{X_1}),\ldots,\Phi_C(C_{X_{d_c}})\}$), [16] gives a solution about how to perform KPCA on the huge Gram matrix. First, eigenvectors of the kernel matrices of K_F and K_C are computed respectively. Then, the eigenvectors of joint Gram matrix can be constructed from the Kronecker product of the eigenvectors of Gram matrices for K_F and K_C. Although this approach reduced the computing time, eigendecompostion of K_F and K_C via KPCA are nevertheless required. Moreover, when K does not have such a factorization, there is no way to avoid performing eigen-decomposion of the huge joint Gram matrix for K.

In order to avoid this general computational problem, we again use incomplete Cholesky decomposition of the joint Gram matrix for the current layer to select pivots. Specifically, our aim is to find a low-rank matrix $\tilde{G}$, such that $\tilde{G}\tilde{G}^{\mathrm{T}}$ is a good approximation for the Gram matrix of K, regardless whether it can be represented as the product of individual kernels. Fortunately, the computation scales linearly the dimensionality of K, which is efficient enough.

Besides, the precision of incomplete Cholesky decomposition can be monitored throughout the computation process. Once the precision requirement (which can be specified by a pre-defined parameter ε) is met, the selected pivots are sufficient to characterize the pairwise similarities of the joint basis vectors.

3.2.2 Construction of the feature map for the current layer

Once we obtained $\tilde{G}$, the new efficient kernel descriptor can be constructed based on $\tilde{G}$. The algorithm for constructing the feature map has been shown in Algorithm 1. But there is slight difference in computing the similarities.

Specifically, the variable similarity in Algorithm 1 represents an M dimensional array of similarities from an image patch to the pivots. All elements participating in the computation of similarity values are based on image pixels. In higher levels of EHKD computation, the efficient kernel descriptor and the relative position of the patches of the previous layer are the basic elements for computing the similarity values, rather than pixel for the first layer. The procedure of the construction efficient hierarchical kernel descriptor is shown in Fig. 1.

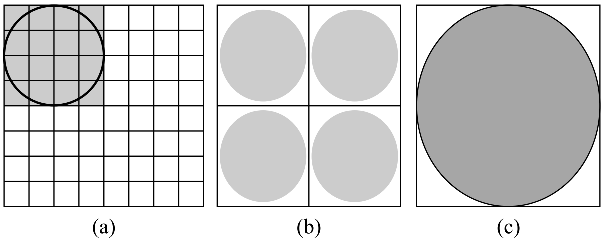

Fig. 1 An illustration for constructing EHKD. Supposing the image is divided into 8×8 patches and number of layers is 3. a) Every grid represents a patch-level efficient kernel descriptor, the 4×4 (tunable in the experiment) grids (colored gray) aggregate into one efficient kernel descriptor for the next layer; b) On the second layer, four (tunable in the experiment) efficient kernel descriptors aggregate into one efficient kernel descriptor of the final layer; c) Representing an image-level feature.

4 Experimental results

We use gradient, color and shape characterized by local binary pattem (LBP) descriptors attributes at the pixel-level for constructing EKD based on gradient (EKD-G), color (EKD-C) and shape (EKD-S) information. To test the performance of the EKD approach, we perform image classification experiments on four well-known datasets: Scene-15 [6, 7, 19], Caltech-101 [20], UIUC-8 [21], and MIT Indoor-67 [22]. Besides, we also construct EHKD to obtain the image-level feature representation and evaluate its performance on the CIFAR10 dataset [23].

In the following experiments for EKD, all images are resized to be no larger than 300×300 pixels with preserved ratio and they are further normalized into grayscale ([0, 1]) for computing EKD-G and EKD-S. Patch-level features are extracted on dense regular grids with 8 pixels spacing. The size of each patch is 16×16 pixels. The image-level features are constructed on the patch-level features by the EMK approach [8] with $1\times1, 2\times2, 4\times4$ sub-regions and using 1000 visual words. EKD-All is the concatenation of the three efficient kernel descriptors (EKD-G, EKD-C, and EKD-S). After obtaining the image-level features, we train a linear SVM classifier using LIBLINEAR [24]. All experiments are repeated 10 times with randomly selected training/test images and the average/standard deviation of classification accuracies are recorded. Other hyper-parameters used in our experiments are the same as [1] to make our comparison fair. In the experiment about EHKD, we use the same parameter settings as the EKD for the first layer of EHKD. Similar to EKD, we focus on evaluating the performance of EHKD-All which is a combination of the three efficient hierarchical kernel descriptors (EHKD-G, EHKD-S, and EHKD-C) by concatenating the Gradient/Shape/Color image-level feature vectors.

4.1 Experimental results for EKD

Scene-15 Scene-15 contains a total of 4 485 images in 15 categories, including indoor/outdoor scenes. Each category contains 200 to 400 images. Following [1], we take 100 images per category for training and use the left images for testing. We also select 200 pivots using the incomplete Cholesky decomposition algorithm to construct 200d feature maps, as in [1]. The results are shown in Table 1 and the EKD algorithm achieved the best classification accuracy in this data set (86.3%). Moreover, we can see the classification accuracy of EKD is consistently better than the kernel descriptor (KD) over all the 4 types of attributes (color, gradient, shape and all the three above).

Table 1 A comparison of the classification accuracy of the kernel descriptor (KD) vs. EKD on Scene-15 dataset

KD [1] vs. EKD			
KD-C	38.5±0.4	EKD-C	**49.1±1.2**
KD-G	81.6±0.6	EKD-G	**82.7±0.6**
KD-S	79.8±0.5	EKD-S	**81.0±0.5**
KD-All	81.9±0.6	EKD-All	**86.3±0.4**

To our surprise, we find that the EKD algorithm can achieve close-to-optimal classification accuracy even with as small as 50 pivots, which firmly demonstrate its efficiency and robustness. In fact, as shown in Table 2, increasing the number of pivots from 50 to 300 only improved the performance marginally. An explanation for this phenomenon is that Gram matrices often have high accuracy low-rank approximations. Since the incomplete Cholesky decomposition algorithm explicitly constructs such approximations through matrix factorization $K \approx \tilde{G}\tilde{G}^T$, we believe the results in Table 2 simply suggest that 50-dimensional low rank approximations are sufficient to represent the key information in Gram matrices for scene classification. For efficiency and without loss of generality, we always use 100 pivots in the experiments below.

Table 2 The EKD classification accuracy on scene-15 dataset by changing the number of pivots from 50 to 300

Method	EKD-C	EKD-G	EKD-S
50d	48.9 ± 0.7	81.8 ± 0.5	78.9 ± 0.8
100d	49.1 ± 0.6	82.4 ± 0.4	79.6 ± 0.7
200d	49.1 ± 1.2	82.7 ± 0.6	81.0 ± 0.5
300d	49.1 ± 0.9	82.6 ± 0.6	81.2 ± 0.5

Caltech-101 Caltech-101 contains 9 144 images in 101 object categories and in one background category. The number of images per category varies from 31 to 800. In Table 3, we compare the performance of EKD to other representative algorithms. Following [25–27], we train on 30 images per category and test on no more than 50 images per category. We can see that the proposed EKD algorithm achieved the highest classification accuracy (EKD-All, with the classification accuracy: 76.9%). Even with gradient features only, EKD-G achieved a competitive classification accuracy of 73.4%.

Table 3 Comparing the classification accuracy of six algorithms on Caltech-101 dataset

Method	Classification accuracy
Shabou et al. [25]	73.23 ± 0.81
NBNN [28]	73.0
LLC [26]	73.4 ± 0.5
Jia et al. [27]	75.3 ± 0.7
KD-All [1]	74.5 ± 0.8
EKD-G	73.4 ± 0.6
EKD-All	**76.9±0.5**

UIUC-8 UIUC-8 contains 8 sport categories with 1 579 images. Each class has 137 to 250 images. Following [21, 29], we randomly select 70 training images per category and 60 testing images per category. The results are shown in Table 4. We can see that the performance of EKD-All (87.1%) is very close to the best classification accuracy achieved by [25] (87.23%) on this dataset.

Table 4 Comparing the classification accuracy of three algorithms on UIUC-8 dataset

Method	Classification accuracy
Liu et al. [29]	84.56 ± 1.5
Shabou et al. [25]	**87.23±1.14**
EKD-All	87.1 ± 1.4

MIT Indoor-67 MIT Indoor-67 contains 15 620 images in 67 indoor scene categories. All images have a minimum resolution of 200 pixels in the smallest axis. This dataset raises a challenging classification problem, since yet some indoor scenes can be well characterized by global spatial properties, others are only characterized by the object contained in the image. Following the same training/test split strategy as in [22], we randomly select 80 training images and 20 testing images in each category. The comparison results are shown in Table 5. We can see that the performance of EKD-All achieves the best classification accuracy (50.8%) on this dataset.

Table 5 Comparing the classification accuracy of seven algorithms on MIT Indoor-67 dataset

Method	Classification accuracy
ROI+gist [22]	26.5
MM-scene [30]	28.0
CENTRIST [31]	36.9
Object Bank [32]	37.6
GIST-color+SP+DPM [33]	43.1
Mid-level Patches+GIST+SP+DPM [34]	49.4
EKD-All	**50.8**

Finally, we compare the computational efficiency of constructing efficient kernel descriptors vs. kernel descriptor. For the most time consuming shape feature, EKD-S algorithm takes about 2 seconds to compute in Matlab on a typical image (300 × 300 resolution and 16 × 16 image patches over 8 pixel spacing), while the original KD-S algorithm takes about 2.5 seconds. For the gradient feature, EKD-G takes about 0.9 seconds while KD-G took 1.0 seconds. This comparison is done in the context of generating 100-dimensional patch-level features. If we further increase the feature dimensionality, the advantage of EKD will be more obvious.

Robustness to hyper-parameter changes Recently, we found that the authors of KD [1] suggested an alternative set of hyper-parameters on their demo after publication. To investigate the sensitivity of KD and EKD to the hyper-parameter changes, we also compare our EKD approach to the original KD approach [1] with this alternative parameter setting used in the gradient/shape/color/position kernel and EMK's [8] match kernel. The comparison results are shown in Table 6, where $\gamma_o/\gamma_b/\gamma_c/\gamma_p$ represent the hyper-parameters used in gradient/shape/color/position kernel function for generating patch-level features, and $kpara$ represents the hyper-parameters used in EMK's [8] gradient/shape/color match kernel function for generating image-level features. For the three sets of hyper-parameters, both the two approaches achieved roughly the same best performance (86.9% and 86.8%) on the Scene-15 dataset, but EKD-All's performance is more consistent across the three repeats with higher average classification accuracy (86.5% vs. 85.2%). This result suggests our EKD approach is more robust to parameter changes than the original KD method [1].

Table 6 A comparison of the classification accuracy of KD-All vs. EKD-All on the Scene-15 dataset with three sets of hyper-parameters. "Ave." represents average classification accuracy

Method	$\gamma_o = 5$ $\gamma_b = 2$ $\gamma_c = 4$ $\gamma_p = 3$ $kpara =$ $[1, 1, 1]$	$\gamma_o = 5$ $\gamma_b = 2$ $\gamma_c = 4$ $\gamma_p = 3$ $kpara =$ $[0.001, 0.01, 0.01]$	$\gamma_o = 5$ $\gamma_b = 3$ $\gamma_c = 5$ $\gamma_p = 3$ $kpara =$ $[0.001, 0.01, 0.01]$	Ave.
KD-All	81.9 ± 0.6	86.7 ± 0.7	86.9 ± 0.7	85.2
EKD-All	86.3 ± 0.4	86.4 ± 0.5	86.8 ± 0.7	86.5

3.1 Experimental results for EHKD

CIFAR10 CIFAR10 is a subset of a 80 million tiny images dataset. All images in this dataset are down-sampled to 32×32 pixels. For each category, there are 5 000 images in the training set and 1 000 images in the test set. Following [16], we use two-layer EHKD to obtain image-level feature and the construction of first layer is the same as EKD. Since the images here are small, EKDs (EKD-G, EKD-S, and EKD-C) are extracted over 8 × 8 image patches over dense regular grids with 2 pixels spacing. On second layer, to enable a fair comparison with [16], we also used 1 000 basis vectors for the patch-level Gaussian kernel k_F. The classification accuracy of different algorithms on this data set is shown in Table 7.

Table 7 Comparing the classification accuracy of seven algorithms on CIFAR10 dataset (The first six classification accuracies marked with asterisk are based on [16].)

Method	Classification accuracy
(∗)mcRBM-DBN [35]	71.0
(∗)Tiled CNNs [36]	73.1
(∗)Improved LCC [37]	74.5
(∗)KDES+EMK+Linear SVMs [1]	76.0
(∗)Convolutional RBM [38]	78.9
(∗)HKDES+Linear SVMs [16]	80.0
EHKD-All	**80.7**

From Table 7, we can see the EHKD-All method outperforms the HKDES [16] 0.7% (**80.7%** vs. 80%) and is also higher than all the other competing approaches.

5 Conclusion

In this paper, we propose EKD to derive patch-level features and EHKD to obtain image-level features. Both the two approaches are built upon the incomplete Cholesky decomposition technique. Under the EKD framework, the most discriminative non-linear feature extractors can be learned automatically from data, which reduced the computational complexity of the kernel descriptor (KD) approach and improved its performance. By recursively applying EKD, we construct EHKD for extracting image-level features in a bottom-up manner. Experimental results demonstrate EKD outperforms KD and EHKD outperforms HKD, and both of them are competitive with state-of-the-art algorithms on publicly available image/scene data sets. In the future, we plan to investigate better ways to pick the pivots more efficiently for generating patch/image-level features, which are helpful for object recognition.

Acknowledgements The authors acknowledge support from the National Natural Science Foundation of China (Grant No. 61300072, 61033013), the Fundamental Research Funds for the Central Universities (2014JBZ005), the Beijing Committee of Science and Technology, China (Z131110002813118), Beijing Jiaotong University (K12RC00090) and Research Foundation of Education Bureau of Hebei Province (Z2013124).

References

1. Bo L F, Ren X F, Fox D. Kernel descriptor for visual recognition. In: Proceedings of the Annual Conference on Neural Information Processing Systems. 2010, 244–252
2. Bosch A, Munõz X, Marti R. Which is the best way to organize/classify images by content? Image and Vision Computing, 2007, 25(6): 778–791
3. Lowe D G. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 2004, 60(2): 91–110
4. Dalal N, Triggs B. Histograms of oriented gradients for human detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2005, 886–893
5. Vogel J, Schiele B. Semantic modeling of natural scenes for content-based image retrieval. International Journal of Computer Vision, 2007, 72(2): 133–157
6. Li F F, Perona P. A bayesian hierarchical model for learning natural scene categories. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2005, 524–531
7. Lazebnik S, Schmid C, Ponce J. Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2006, 2169–2178
8. Bo L F, Sminchisescu C. Efficient match kernel between sets of features for visual recognition. In: Proceedings of the Annual Conference on Neural Information Processing Systems. 2009, 135–143
9. Schölkopf B, Smola A, Müller K. Nonlinear component analysis as a kernel eigenvalue problem. Neurocomputing, 1998, 10(5): 1299–1319
10. Xie B J, Liu Y, Zhang H, Yu J. Efficient kernel descriptor for image categorization via pivots selection. In: Proceedings of the IEEE International Conference on Image Processing. 2013, 3479–3483
11. Wang P, Wang J D, Zeng G, Xu W W, Zha H B, Li S P. Supervised kernel descriptors for visual recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2013, 2858–2865
12. LeCun Y, Huang F J, Bottou L. Learning methods for generic object recognition with invariance to pose and lighting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2004, 97–104
13. Hinton G, Osindero S, Teh Y. A fast learning algorithm for deep belief nets. Neural Computation, 2006, 18(7): 1527–1554
14. Hinton G, Salakhutdinov R. Reducing the dimensionality of data with neural networks. Science, 2006, 313(5786): 504–507
15. Krizhevsky A, Sutskever I, Hinton G. Imagenet classification with deep convolutional neural networks. In: Proceedings of the Annual Conference on Neural Information Processing Systems. 2012, 1106–1114
16. Bo L F, Lai K, Ren X F, Fox D. Object recognition with hierarchical kernel descriptors. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2011, 1729–1736
17. Fine S, Scheinberg K. Efficient svm training using low-rank kernel representation. Journal of Machine Learning Research, 2001, 2: 243–264
18. Bach F R, Jordan M I. Kernel independent component analysis. Journal of Machine Learning Research, 2002, 3: 1–48
19. Oliva A, Torralba A. Modeling the shape of the scene: a holistic representation of the spatial envelope. International Journal of Computer Vision, 2001, 42(3): 145–175
20. Li F F, Fergus R, Perona P. Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. Computer Vision and Image Understanding, 2007, 106(1): 59–70
21. Li L J, Li F F. What, where and who? Classifying events by scene and object recognition. In: Proceedings of the IEEE International Conference on Computer Vision. 2007, 1–8
22. Quattoni A, Torralba A. Recognizing indoor scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2009, 413–420
23. Torralba A, Fergus R, Freeman W. 80 million tiny images: a large data set for nonparametric object and scene recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2008, 30(11): 1958–1970
24. Fan R E, Chang K W, Hsieh C J, Wang X R, Lin C J. Liblinear: a library for large linear classification. Journal of Machine Learning Research, 2008, 9: 1871–1874
25. Shabou A, Borgne H L. Locality-constrained and spatially regularized coding for scene categorization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2012, 3618–3625
26. Wang J J, Yang J C, Yu K, Lv F J, Huang T, Gong Y H. Locality-constrained linear coding for image classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2010, 3360–3367
27. Jia Y Q, Huang C, Darrell T. Beyond spatial pyramids: receptive field learning for pooled image features. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2012, 3370–3377
28. Boiman O, Shechtman E, Irani M. In defense of nearest-neighbor based image classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2008, 1–8
29. Liu L Q, Wang L, Liu X W. In defense of soft-assignment coding. In:

Proceedings of the IEEE International Conference on Computer Vision. 2011, 2486–2493

30. Zhu J, Li L J, Li F F, Xing E. Large margin learning of upstream scene understanding models. In: Proceedings of the Annual Conference on Neural Information Processing Systems. 2010, 2586–2594
31. Wu J, Rehg J. Centrist: a visual descriptor for scene categorization. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011, 33(8): 1489–1501
32. Li L J, Su H, Xing E, Li F F. Object bank: a high-level image representation for scene classification & semantic feature sparsification. In: Proceedings of the Annual Conference on Neural Information Processing Systems. 2010, 1378–1386
33. Pandey M, Lazebnik S. Scene recognition and weakly supervised object localization with deformable part-based models. In: Proceedings of the IEEE International Conference on Computer Vision. 2011, 1307–1314
34. Singh S, Gupta A, Efros A A. Unsupervised discovery of mid-level discriminative patches. In: Proceedings of the European conference on Computer Vision. 2012, 73–86
35. Ranzato M, Hinton G. Modeling pixel means and covariances using factorized third-order boltzmann machines. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2010, 2551–2558
36. Le Q, Ngiam J, Chia Z C, Koh P, Ng A. Tiled convolutional neural networks. In: Proceedings of the Annual Conference on Neural Information Processing Systems. 2010, 1279–1287
37. Yu K, Zhang T. Improved local coordinate coding using local tangents. In: Proceedings of International Conference on Machine Learning. 2010, 1215–1222
38. Coates A, Lee H, Ng A. An analysis of single-layer networks in unsupervised feature learning. In: Proceedings of International Conference on Artificial Intelligence and Statistics. 2011, 215–223

Bojun Xie received the BS degree in computer science and technology, MS degree in computer application from Hebei University, China in 2003 and 2006, respectively. He is currently a PhD candidate in Beijing Jiaotong University China. His interests include machine learning and computer vision.

Yi Liu received BS and PhD degrees from Peking University, China in 2004 and 2009, respectively. His current research interests include reasoning and uncertainty modeling in systems biology, machine learning, information retrieval and 3D geometric processing.

Hui Zhang received the BS degree in information and computing science, MS degree in applied mathematics from Hebei University, China in 2003 and 2006, respectively. He is currently a PhD candidate in Beijing Jiaotong University China. His interests include machine learning and computer vision.

Jian Yu received the BS degree in applied mathematics, MS degree in mathematics, and PhD degree in applied mathematics from Peking University, China in 1991, 1994, and 2000, respectively. He is a professor and head of Institute of Computer Science Beijing Jiaotong University China. His current research interests include fuzzy clustering, pattern recognition, and data mining.