Object Recognition with Hierarchical Kernel Descriptors

Liefeng Bo[1]   Kevin Lai[1]   Xiaofeng Ren[2]   Dieter Fox[1,2]

University of Washington[1]   Intel Labs Seattle[2]

{lfb,kevinlai,fox}@cs.washington.edu   xiaofeng.ren@intel.com

Abstract

Kernel descriptors [1] provide a unified way to generate rich visual feature sets by turning pixel attributes into patch-level features, and yield impressive results on many object recognition tasks. However, best results with kernel descriptors are achieved using efficient match kernels in conjunction with nonlinear SVMs, which makes it impractical for large-scale problems. In this paper, we propose hierarchical kernel descriptors that apply kernel descriptors recursively to form image-level features and thus provide a conceptually simple and consistent way to generate image-level features from pixel attributes. More importantly, hierarchical kernel descriptors allow linear SVMs to yield state-of-the-art accuracy while being scalable to large datasets. They can also be naturally extended to extract features over depth images. We evaluate hierarchical kernel descriptors both on the CIFAR10 dataset and the new RGB-D Object Dataset consisting of segmented RGB and depth images of 300 everyday objects.

1. Introduction

Object recognition is a fundamental and challenging problem and is a major focus of research in computer vision, machine learning and robotics. The task is difficult partly because images are in high-dimensional space and can change with viewpoint, while the objects themselves may be deformable, leading to large intra-class variation. The core of building object recognition systems is to extract meaningful representations (features) from high-dimensional observations such as images, videos, and 3D point clouds. This paper aims to discover such representations using machine learning methods.

Over the past few years, there has been increasing interest in feature learning for object recognition using machine learning methods. Deep belief nets (DBNs) [8, 9] are appealing feature learning methods that can learn a hierarchy of features. DBNs are trained one layer at a time using contrastive divergence [3], where the feature learned by the current layer becomes the data for training the next

layer. Deep belief nets have shown impressive results on handwritten digit recognition, speech recognition and visual object recognition. Convolutional neural networks (CNNs) [11, 13] are another example that can learn multiple layers of nonlinear features. In CNNs, the parameters of the entire network, including a final layer for recognition, are jointly optimized using the back-propagation algorithm.

Current state-of-the-art object recognition algorithms [7, 30] are based on local descriptors extracted from local patches over a regular grid on an image. The most popular and successful local descriptors are orientation histograms including SIFT [19] and HOG [5], which are robust to minor transformations of images. Although they are very successful, we still lack a deep understanding of what are the design rules behind them and how they measure the similarity between image patches. Recent work on kernel descriptors [1] tries to answer these questions. In particular, they show that orientation histograms are equivalent to a certain type of match kernel over image patches. Based on this novel view, a family of kernel descriptors are proposed, which are able to turn pixel attributes (gradient, color, local binary pattern, *etc.*) into patch-level features. Kernel descriptors have shown higher accuracy than many state-of-the-art algorithms on standard object recognition benchmarks [1].

While kernel descriptors are great for visual object recognition, in the second stage, efficient match kernels and expensive nonlinear SVMs with Laplacian kernels are necessary for yielding good performance. Motivated by the recent work on deep belief nets and convolutional neural networks, in this work we extend kernel descriptors to hierarchical kernel descriptors that apply kernel descriptors recursively to aggregate lower level features into higher level features, layer by layer (see Fig. 1). Hierarchical kernel descriptors provide a conceptually simple and consistent way to generate rich features from various pixel attributes of RGB and depth images. Experiments on both CIFAR10 and the RGB-D Object Dataset (available at http://www.cs.washington.edu/rgbd-dataset) show that hierarchical kernel descriptors outperform kernel descriptors and many state-of-the-art algorithms including deep belief

nets, convolutional neural networks, and local coordinate coding with carefully tuned SIFT features. In addition, the feature learning framework has been applied to develop an RGB-D (Kinect style) camera based object-aware situated interactive system (OASIS) that was successfully shown live at the Consumer Electronics Show (CES) 2011 (see http://www.cs.washington.edu/rgbd-dataset/demos.html).

1.1. Related Work

This research focuses on hierarchical feature learning and its application to object recognition. In the past few years, a growing amount of research on object recognition has focused on learning rich features using unsupervised learning, hierarchical architectures, and their combination.

Deep belief nets [8] learn the weights in multiple layers by greedily training each layer separately using unsupervised algorithms and provide a way to automatically build a hierarchy of features. The learned weights are then used to initialize multi-layer feedback networks that further adjust the weights using the back-propagation algorithm. Convolutional deep belief nets [18] have been shown to yield competitive performance in visual object recognition. Very recently, factorized third-order restricted Boltzmann machine [25] (mcRBM) have been introduced to capture high order dependencies of images. The model consists of two sets of hidden units, one representing the pixel intensities, and the other representing pairwise dependencies between pixel intensities. The binary features from mcRBM can then be fed as input to standard binary deep belief nets to yield hierarchical models of natural images with many layers of non-linear features. Convolutional Neural Networks [29] is another example of trainable hierarchical feed-forward models. CNNs have been successfully applied to a wide range of applications including character recognition, pose estimation, face detection, and recently generic object recognition.

Sparse coding [21] is a traditional method for feature learning. Recent work has focused on learning sparse representations for local features such as raw image patches and SIFT descriptors [19]. Raina et al. [17] used sparse coding to construct image-level features and showed that sparse representations outperform conventional representations, i.e. raw image patches. Yang et al. [30] proposed a spatial pyramid sparse coding model that learns sparse representations over SIFT features. In conjunction with max pooling, their approach achieves state-of-the-art performance on several standard object recognition tasks. Multi-layer feedback networks were proposed to speed up sparse coding at the test stage [11]. Yu et al. recently introduced local coordinate coding [32] and its improved version [27, 31], which quickly computes sparse representations based on nearest neighbors and can model manifold geometric structures in high dimensions.

Hua et al. [10] learned a linear transformation for SIFT using linear discriminant analysis and showed better results with lower dimensionality than SIFT on local feature matching problems. Philbin et al. [22] learned a non-linear transformation with deep networks by minimizing margin-based cost functions and presented impressive results on object retrieval tasks.

Though multilayer kernel machines [28] are able to extract features recursively, they use a very different family of kernel functions from our hierarchical kernel descriptors. The most relevant work is kernel descriptors [1], which learns patch-level features by transforming pixel attributes using match kernels. However, they use spatial pyramid efficient match kernels (EMK) [2] to create image-level features by applying projections in kernel space or random Fourier transformations to patch-level features. While the work on efficient match kernels is appealing, nonlinear SVMs with Laplacian kernels are required to obtain good accuracy [1]. In this paper, we present a consistent, conceptually simple way to construct image-level features by recursively using kernel descriptors. The resulting representations, called hierarchical kernel descriptors, combined with linear SVMs outperform many state-of-the-art algorithms.

2. Hierarchial Kernel Descriptors

Kernel descriptors [1] highlight the kernel view of orientation histograms, such as SIFT and HOG, and show that they are a particular type of match kernels over patches. This novel view suggests a unified framework for turning pixel attributes (gradient, color, local binary pattern, *etc.*) into patch-level features: (1) design match kernels using pixel attributes; (2) learn compact basis vectors using kernel principal component analysis (KPCA); (3) construct kernel descriptors by projecting the infinite-dimensional feature vectors to the learned basis vectors.

The key idea of this work is that we can apply the kernel descriptor framework not only over sets of pixels (patches), but also sets of kernel descriptors. Hierarchical kernel descriptors aggregate spatially nearby patch-level features to form higher level features by using kernel descriptors recursively, as shown in Fig. 1. This procedure can be repeated until we reach the final image-level features.

2.1. Kernel Descriptors

Patch-level features are critical for many computer vision tasks. Orientation histograms like SIFT and HOG are popular patch-level features for object recognition. Kernel descriptors include SIFT and HOG as special cases, and provide a principled way to generate rich patch-level features from various pixel attributes.

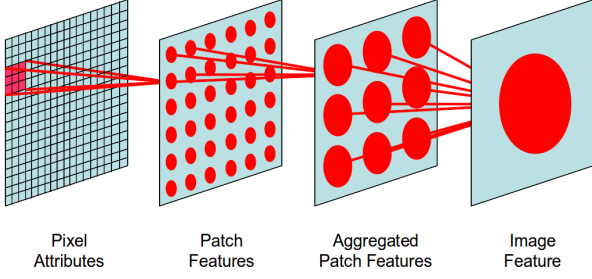The gradient match kernel, K_{grad}, is based on the pixel

Figure 1. Hierarchical Kernel Descriptors. In the first layer, pixel attributes are aggregated into patch-level features. In the second layer, patch-level features are turned into aggregated patch-level features. In the final layer, aggregated patch-level features are converted into image-level features. Kernel descriptors are used in every layer.

gradient attribute

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

where P and Q are patches from two different images, and z denotes the 2D position of a pixel in an image patch (normalized to $[0, 1]$).

Let θ_z, m_z be the orientation and magnitude of the image gradient at a pixel z. The normalized linear kernel $\tilde{m}_z \tilde{m}_{z'}$ weights the contribution of each gradient where $\tilde{m}_z = m_z / \sqrt{\sum_{z \in P} m_z^2 + \epsilon_g}$ and ϵ_g is a small positive constant; the position Gaussian kernel $k_p(z, z') = \exp(-\gamma_p \|z - z'\|^2) = \phi_p(z)^\top \phi_p(z')$ measures how close two pixels are spatially; the orientation kernel $k_o(\tilde{\theta}_z, \tilde{\theta}_{z'}) = \exp(-\gamma_o \|\tilde{\theta}_z - \tilde{\theta}_{z'}\|^2) = \phi_o(\tilde{\theta}_z)^\top \phi_o(\tilde{\theta}_{z'})$ computes the similarity of gradient orientations where $\tilde{\theta}_z = [\sin(\theta_z) \ \cos(\theta_z)]$.

The color kernel descriptor K_{col} is based on the pixel intensity attribute

$$K_{\mathrm{col}}(P, Q) = \sum_{z \in P} \sum_{z' \in Q} k_c(c_z, c_{z'}) k_p(z, z') \quad (2)$$

where c_z is the pixel color at position z (intensity for gray images and RGB values for color images) and $k_c(c_z, c_{z'}) = \exp(-\gamma_c \|c_z - c_{z'}\|^2)$ is a Gaussian kernel.

The shape kernel descriptor, K_{shape}, is based on the local binary pattern attribute [20]:

$$K_{\mathrm{shape}}(P, Q) = \sum_{z \in P} \sum_{z' \in Q} \tilde{s}_z \tilde{s}_{z'} k_b(b_z, b_{z'}) k_p(z, z') \quad (3)$$

where $\tilde{s}_z = s_z / \sqrt{\sum_{z \in P} s_z^2 + \epsilon_s}$, s_z is the standard deviation of pixel values in the 3×3 local window around z, ϵ_s a small constant, b_z is a binary column vector that binarizes the pixel value differences in the 3×3 local window around z, and $k_b(b_z, b_{z'}) = \exp(-\gamma_b \|b_z - b_{z'}\|^2)$ is a Gaussian kernel.

Match kernels are computationally expensive when image patches are large [2]. Kernel descriptors provide a way to extract compact low-dimensional features from match kernels: (1) uniformly and densely sample sufficient basis vectors from the support region to guarantee accurate approximation to match kernels; (2) learn compact basis vectors using KPCA. Gradient kernel descriptors have the form

$$F_{\mathrm{grad}}^t(P) = \sum_{i=1}^{d_o} \sum_{j=1}^{d_p} \alpha_{ij}^t \left\{ \sum_{z \in P} \tilde{m}_z k_o(\tilde{\theta}_z, x_i) k_p(z, y_j) \right\} \quad (4)$$

where $\{x_i\}_{i=1}^{d_o}$ and $\{y_j\}_{j=1}^{d_p}$ are uniformly sampled from their support region, d_o and d_p are the sizes of basis vectors for the orientation and position kernel, and α_{ij}^t are projection coefficients computed by applying KPCA to the joint basis vector set: $\{\phi_o(x_1) \otimes \phi_p(y_1), \cdots, \phi_o(x_{d_o}) \otimes \phi_p(y_{d_p})\}$ ($\otimes$ is Kronecker product).

Gradient, color and shape kernel descriptors are strong in their own right and complement one another. Their combination turns out to be always (much) better than the best individual feature. Kernel descriptors are able to generate rich visual feature sets by turning various pixel attributes into patch-level features, and are superior to the current state-of-the-art recognition algorithms on many standard visual object recognition datasets [1].

2.2. Kernel Descriptors over Kernel Descriptors

The match kernels used to aggregate patch-level features have similar structure to those used to aggregate pixel attributes:

$$K(\overline{P}, \overline{Q}) = \sum_{A \in \overline{P}} \sum_{A' \in \overline{Q}} \widetilde{W}_A \widetilde{W}_{A'} k_F(F_A, F_{A'}) k_C(C_A, C_{A'}) \quad (5)$$

where A and A' denote image patches, and $\overline{P}$ and $\overline{Q}$ are sets of image patches.

The patch position Gaussian kernel $k_C(C_A, C_{A'}) = \exp(-\gamma_C \|C_A - C_{A'}\|^2) = \phi_C(C_A)^\top \phi_C(C_{A'})$ describes the spatial relationship between two patches, where C_A is the center position of patch A (normalized to $[0, 1]$). The patch Gaussian kernel $k_F(F_A, F_{A'}) = \exp(-\gamma_F \|F_A - F_{A'}\|^2) = \phi_F(F_A)^\top \phi_F(F_{A'})$ measures the similarity of two patch-level features, where F_A are gradient, shape or color kernel descriptors in our case. The linear kernel $\widetilde{W}_A \widetilde{W}_{A'}$ weights the contribution of each patch-level feature where $\widetilde{W}_A = W_A / \sqrt{\sum_{A \in \overline{P}} W_A^2 + \epsilon_h}$ and ϵ_h is a small positive constant. W_A is the average of gradient magnitudes for the gradient kernel descriptor, the average of standard deviations for the shape kernel descriptor and is always 1 for the color kernel descriptor.

Note that although efficient match kernels [1] used match kernels to aggregate patch-level features, they don't con-

sider spatial information in match kernels and so spatial pyramid is required to integrate spatial information. In addition, they also do not weight the contribution of each patch, which can be suboptimal. The novel joint match kernels (5) provide a way to integrate patch-level features, patch variation, and spatial information jointly.

Evaluating match kernels (5) is expensive. Both for computational efficiency and for representational convenience, we again extract compact low-dimensional features from (5) using the idea from kernel descriptors.

The inner product representation of two Gaussian kernels is given by

$$k_F(F_A, F_{A'})k_C(C_A, C_{A'}) =$$
$$= [\phi_F(F_A) \otimes \phi_C(C_A)]^\top [\phi_F(F_{A'}) \otimes \phi_C(C_{A'})] \quad (6)$$

Following [1], we learn compact features by projecting the infinite-dimensional vector $\phi_F(F_A) \otimes \phi_C(C_A)$ to a set of basis vectors. Since C_A is a two-dimensional vector, we can generate the set $\{\phi_C(X_1), \cdots, \phi_C(X_{d_C})\}$ of basis vectors by sampling X on 5×5 regular grids ($d_C = 25$). However, patch-level features F_A are in high-dimensional space and it is infeasible to sample them on dense and uniform grids. Instead, we cluster patch-level features from training images using K-means, similar to the bag of visual words method, and take the resulting centers as the set $\{\phi_F(Y_1), \cdots, \phi_F(Y_{d_F})\}$ of basis vectors.

The dimensionality of the second layer kernel descriptors is the total number of joint basis vectors $\{\phi_F(Y_1) \otimes \phi_C(X_1), \cdots, \phi_F(Y_{d_F}) \otimes \phi_C(X_{d_C})\}$. If 5000 basis vectors are generated from patch-level features, the dimensionality of the second layer kernel descriptors is $5000 \times 25 = 125,000$. To obtain the second layer kernel descriptors of reasonable size, we can reduce the number of basis vectors using KPCA. KPCA finds the linear combination of basis vectors that best preserves variance of the original data. The first kernel principal component can be computed by maximizing the variance of projected data with the normalization condition $\beta^\top \beta$:

$$\frac{[H_F \otimes H_C \beta]^\top [H_F \otimes H_C \beta]}{\beta^\top \beta} = \frac{\beta^\top \mathbf{K}_F \otimes \mathbf{K}_C \beta}{\beta^\top \beta} \quad (7)$$

where $H_F = [\phi_F(Y_1), \cdots, \phi_F(Y_{d_F})]$ and $H_C = [\phi_C(X_1), \cdots, \phi_C(X_{d_C})]$. The optimal β equals to the eigenvector having the largest eigenvalue:

$$\mathbf{K}_F \otimes \mathbf{K}_C \beta = \lambda \beta \quad (8)$$

If we consider an r-dimensional projection space, the optimal linear projection is defined by the r eigenvectors $\beta^1, \cdots, \beta^r$ of the kernel matrix $\mathbf{K}_F \otimes \mathbf{K}_C$ corresponding to the r largest eigenvalues $\lambda^1, \cdots, \lambda^r$. KPCA is

performed on the joint kernel, the product of spatial kernel k_C and feature kernel k_F, which can be written as a single Gaussian kernel. This procedure is optimal in the sense of minimizing the least square approximation error. However, it is intractable to compute the eigenvectors of a $125,000 \times 125,000$ matrix on a modern personal computer. Here we propose a fast algorithm for finding the eigenvectors of the Kronecker product of kernel matrices. Since kernel matrices are symmetric positive definite, we have

$$\mathbf{K}_F \otimes \mathbf{K}_C = [U_F^\top S_F U_F] \otimes [U_C^\top S_C U_C]$$
$$= [U_F \otimes U_C]^\top [S_F \otimes S_C][U_F \otimes U_C] \quad (9)$$

Eq. (9) suggests that the top r eigenvectors of $\mathbf{K}_F \otimes \mathbf{K}_C$ can be chosen from the Kronecker product of the eigenvectors of $\mathbf{K}_F$ and those of $\mathbf{K}_C$, which significantly reduces computational cost. The second layer kernel descriptors have the form

$$\overline{F}^t(\overline{P}) = \sum_{i=1}^{d_F} \sum_{j=1}^{d_C} \beta_{ij}^t \left\{ \sum_{A \in \overline{P}} \widetilde{W}_A k_F(F_A, Y_i) k_C(C_A, X_j) \right\} \quad (10)$$

Recursively applying kernel descriptors in a similar manner, we can get kernel descriptors of more layers, which represents features at different levels.

3. Everyday Object Recognition using RGB-D

We are witnessing a new wave of sensing technologies capable of providing high quality synchronized videos of both color and depth: the RGB-D (Kinect style) camera [23]. With active sensing capabilities and the potential for mass consumer adoption, RGB-D cameras represents an opportunity to dramatically increase the robustness of object recognition toward real-life recognition applications.

An interesting scenario and benchmark of RGB-D object recognition is presented in the recent study of [15]. Their RGB-D Object Dataset contains color and depth images of 300 physically distinct objects taken from multiple views. The chosen objects are commonly found in home and office environments. The RGB-D camera simultaneously records both color and depth images at 640×480 resolution. In other words, each 'pixel' in an RGB-D frame contains four channels: red, green, blue and depth. The 3D location of each pixel in physical space can be computed using known sensor parameters. Unlike stereo-based cameras that compute depth images using visual correspondence, the RGB-D camera projects an infrared pattern and measures its deformation. This results in much more reliable depth readings, particularly for textureless regions (Fig. 3).

This dataset contains video sequences of each object as it is spun around on a turntable at constant speed. The camera is placed about one meter from the turntable. Data was

Figure 2. Sample objects from the RGB-D Object Dataset. Each object shown here comes from a different category.
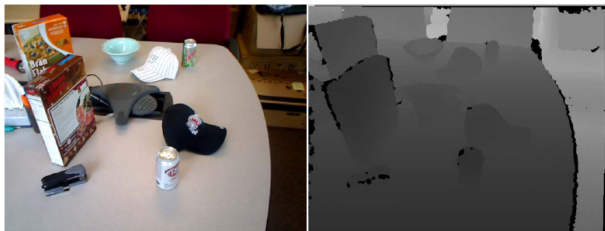

Figure 3. RGB and Depth image captured by an RGB-D camera. The black pixels in the right image are missing depth values.

recorded with the camera mounted at three different heights relative to the turntable, giving viewing angles of approximately 30, 45 and 60 degrees with the horizon. One revolution of each object was recorded at each height. Each video sequence is recorded at 20 Hz and contains around 250 frames, giving a total of 250,000 RGB + Depth frames. A combination of visual and depth cues (Mixture-of-Gaussian fitting on RGB, RANSAC plane fitting on depth) produces a segmentation for each frame separating the object of interest from the background. The objects are organized into a hierarchy taken from WordNet hypernym/hyponym relations and is a subset of the categories in ImageNet [6]. Each of the 300 objects in the dataset belong to one of 51 categories.

Our hierarchical kernel descriptors, being a generic approach based on kernels, has no trouble generalizing from color images to depth images. Treating a depth image as a grayscale image, i.e. using depth values as intensity, gradient and shape kernel descriptors can be directly extracted and they capture edge and shape information in the depth

channel. However, color kernel descriptors extracted over the raw depth image does not have any significant meaning. Instead, we make the observation that the distance d of an object from the camera is inversely proportional to the square root of its area s in RGB images. For a given object, $d\sqrt{s}$ is approximately constant. Since we have the segmentation of objects, we can represent s using the number of pixels belonging to the object mask. Finally, we multiply depth values by $\sqrt{s}$ before extracting color kernel descriptors over this normalized depth image. This yields a feature that is sensitive to the physical size of the object.

In the experiments section, we will compare in detail the performance of our hierarchical kernel descriptors on RGB-D object recognition to that in [15]. Our approach consistently outperforms the state of the art in [15]. In particular, our hierarchical kernel descriptors on the depth image perform much better than the combination of depth features (including spin images) used in [15], increasing the depth-only object category recognition from 53.1% (linear SVMs) and 64.7% (nonlinear SVMs) to 75.7% (hierarchical kernel descriptors and linear SVMs). Moreover, our depth features served as the backbone in the object-aware situated interactive system that was successfully demonstrated at the Consumer Electronics Show 2011 despite adverse lighting conditions (see http://www.cs.washington.edu/rgbd-dataset/demos.html).

4. Experiments

In this section, we evaluate hierarchical kernel descriptors on CIFAR10 and the RGB-D Object Dataset. We also

Features	KDES [1]	HKDES (this work)
Color	53.9	**63.4**
Shape	68.2	**69.4**
Gradient	66.3	**71.2**
Combination	76.0	**80.0**

Table 1. Comparison of kernel descriptors (KDES) and hierarchical kernel descriptors (HKDES) on CIFAR10.

Method	Accuracy
Logistic regression [25]	36.0
Support Vector Machines [1]	39.5
GIST [25]	54.7
SIFT [1]	65.6
fine-tuning GRBM [24]	64.8
GRBM two layers [24]	56.6
mcRBM [25]	68.3
mcRBM-DBN [25]	71.0
Tiled CNNs [16]	73.1
improved LCC [31]	74.5
KDES + EMK + linear SVMs [1]	76.0
Convolutional RBM [4]	78.9
K-means (Triangle, 4k features) [4]	79.6
HKDES + linear SVMs (this work)	**80.0**

Table 2. Comparison of state-of-the-art algorithms on CIFAR10.

provide extensive comparisons with current state-of-the-art algorithms in terms of accuracy.

In all experiments we use the same parameter settings as the original kernel descriptors for the first layer of hierarchical kernel descriptors. For SIFT as well as gradient and shape kernel descriptors, all images are transformed into grayscale ([0, 1]). Image intensity and RGB values are normalized to [0, 1]. Like HOG [5], we compute gradients using the mask $[-1, 0, 1]$ for gradient kernel descriptors. We also evaluate the performance of the combination of the three hierarchical kernel descriptors by concatenating the image-level feature vectors. Our experiments suggest that this combination always improves accuracy.

4.1. CIFAR10

CIFAR10 is a subset of the 80 million tiny images dataset [26, 14]. These images are downsampled to 32×32 pixels. The training set contains 5,000 images per category, while the test set contains 1,000 images per category.

Due to the tiny image size, we use two-layer hierarchical kernel descriptors to obtain image-level features. We keep the first layer the same as kernel descriptors. Kernel descriptors are extracted over 8×8 image patches over dense regular grids with a spacing of 2 pixels. We split the whole training set into 10,000/40,000 training/validation set, and optimize the kernel parameters of the second layer kernel descriptors on the validation set using grid search. Finally, we train linear SVMs on the full training set using the optimized kernel parameter setting. Our hierarchical model can handle large numbers of basis vectors. We tried both 1000 and 5000 basis vectors for the patch-level Gaussian kernel k_F, and found that a larger number of visual words is slightly better (0.5% to 1% improvement depending on the type of kernel descriptor). In the second layer, we use 1000 basis vector, enforce KPCA to keep 97% of the energy for all kernel descriptors, and produce roughly 6000-dimensional image-level features. Note that the second layer of hierarchical kernel descriptors are image-level features, and should be compared to that of image-level features formed by EMK, rather than that of kernel descriptors over image patches. The dimensionality of EMK features [1] in is 14000, higher than that of hierarchical kernel descriptors.

We compare kernel descriptors and hierarchical kernel descriptors in Table 1. As we see, hierarchical kernel descriptors consistently outperform kernel descriptors. The shape hierarchical kernel descriptor is slightly better than the shape kernel descriptor. The other two hierarchical kernel descriptors are much better than their counterparts: gradient hierarchical kernel descriptor is about 5 percent higher than gradient kernel descriptor and color hierarchical kernel descriptor is 10 percent better than color kernel descriptor. Finally, the combination of all three hierarchical kernel descriptors outperform the combination of all three kernel descriptors by 4 percent. We were not able to run nonlinear SVMs with Laplacian kernels on the scale of this dataset in reasonable time, given the high dimensionality of image-level features. Instead, we make comparisons on a subset of 5,000 training images and our experiments suggest that nonlinear SVMs have similar performance with linear SVMs when hierarchical kernel descriptors are used.

We compare hierarchical kernel descriptors with the current state-of-the-art feature learning algorithms in Table 2. Deep belief nets and sparse coding have been extensively evaluated on this dataset [25, 31]. mcRBM can model pixel intensities and pairwise dependencies between them jointly. Factorized third-order restricted Boltzmann machine, followed by deep belief nets, has an accuracy of 71.0%. Tiled CNNs has the best accuracy among deep networks. The improved LCC extends the original local coordinate coding by including local tangent directions and is able to integrate geometric information. As we have seen, sophisticated feature extraction can significantly boost accuracy and is much better than using raw pixel features. SIFT features have an accuracy of 65.2% and works reasonably even on tiny images. The combination of three hierarchical kernel descriptors has an accuracy of **80.0%**, higher than all other competing techniques; its accuracy is 14.4 percent higher than SIFT, 9.0 percent higher than mcRBM combined with

DBNs, and 5.5 percent higher than the improved LCC. Hierarchical kernel descriptors slightly outperform the very recent work: the convolutional RBM and the triangle K-means with 4000 centers [4].

4.2. RGB-D Object Dataset

We evaluated hierarchical kernel descriptors on the RGB-D Object Dataset. The goal of this experiment is to: 1) verify that hierarchical kernel descriptors work well for both RGB and depth images; 2) test whether using depth information can improve object recognition. We subsampled the turntable video data by taking every fifth frame, giving around 41,877 RGB-depth image pairs. To the best of our knowledge, the RGB-D Object Dataset presented here is the largest multi-view object dataset where both RGB and depth images are provided for each view.

We use two-layer hierarchical kernel descriptors to construct image-level features. We keep the first layer the same as kernel descriptors and tune the kernel parameters of the second layer kernel descriptors by cross validation optimization. We extract the first layer of kernel descriptors over 16×16 image patches in dense regular grids with spacing of 8 pixels. In the second layer, we use 1000 basis vectors for the patch-level Gaussian kernel k_F, enforce that KPCA keep 97% of the energy for all kernel descriptors as mentioned in Section 4.1, and produce roughly 3000-dimensional image-level features. Finally, we train linear SVMs on the training set and apply them on the test set. We also tried three layer kernel descriptors, but they gave similar performance to two-layer ones.

As in [15], we distinguish between two levels of object recognition: instance recognition and category recognition. Instance recognition is recognizing distinct objects, for example a coffee mug with a particular appearance and shape. Category recognition is determining the category name of an object (e.g. coffee mug). One category usually contains many different object instances.

To test the generalization ability of our approaches, for category recognition we train models on a set of objects and at test time present to the system objects that were not present in the training set [15]. At each trial, we randomly leave one object out from each category for testing and train classifiers on the remaining 300 - 51 = 249 objects. For instance recognition we also follow the experimental setting suggested by [15]: train models on the video sequences of each object where the viewing angles are $30°$ and $60°$ with the horizon and test them on the $45°$ video sequence.

For category recognition, the average accuracy over 10 random train/test splits is reported in the second column of Table 3. For instance recognition, the accuracy on the test set is reported in the third column of Table 3. As we expect, the combination of hierarchical kernel descriptors is much better than any single descriptor. The underlying rea-

Method	Category	Instance
Color HKDES (RGB)	60.1±2.1	58.4
Shape HKDES (RGB)	72.6±1.9	74.6
Gradient HKDES (RGB)	70.1±2.9	75.9
Combination of HKDES (RGB)	76.1±2.2	79.3
Color HKDES (depth)	61.8±2.4	28.8
Shape HKDES (depth)	65.8±1.8	36.7
Gradient HKDES (depth)	70.8±2.7	39.3
Combination of HKDES (depth)	75.7±2.6	46.8
Combination of all HKDES	**84.1±2.2**	**82.4**

Table 3. Comparisons on the RGB-D Object Dataset. RGB denotes features over RGB images and depth denotes features over depth images.

Approaches	Category	Instance
Linear SVMs [15]	81.9±2.8	73.9
Nonlinear SVMs [15]	83.8±3.5	74.8
Random Forest [15]	79.6±4.0	73.1
Combination of all HKDES	**84.1±2.2**	**82.4**

Table 4. Comparisons to existing recognition approaches using a combination of depth features and image features. Nonlinear SVMs use Gaussian kernel.

son is that each depth descriptor captures different information and the weights learned by linear SVMs using supervised information can automatically balance the importance of each descriptor across objects.

In Table 4, we compare hierarchical kernel descriptors with the rich feature set used in [15], where SIFT, color and textons were extracted from RGB images, and 3-D bounding boxes and spin images [12] over depth images. Hierarchical kernel descriptors are slightly better than this rich feature set for category recognition, and much better for instance recognition.

It is worth noting that, using depth alone, we improve the category recognition accuracy in [15] from 53.1% (linear SVMs) to 75.7% (hierarchical kernel descriptors and linear SVMs). This shows the power of our hierarchical kernel descriptor formulation when being applied to a nonconventional domain. The depth-alone results are meaningful for many scenarios where color images are not used for privacy or robustness reasons.

As a comparison, we also extracted SIFT features on both RGB and depth images and trained linear SVMs over image-level features formed by spatial pyramid EMK. The resulting classifier has an accuracy of 71.9% for category recognition, much lower than the result of the combination of hierarchical kernel descriptors (**84.2%**). This is not surprising since SIFT fails to capture shape and object size information. Nevertheless, hierarchical kernel descriptors provide a unified way to generate rich feature sets over both RGB and depth images, giving significantly better accuracy.

5. Conclusion

We have proposed hierarchical kernel descriptors for extracting image features layer by layer. Our approach is based on the observation that kernel descriptors can be recursively used to produce features at different levels. We have compared hierarchical kernel descriptors to current state-of-the-art algorithms and shown that our hierarchical kernel descriptors have the best accuracy on CIFAR10, a large scale visual object recognition dataset to date. In addition, we also evaluated our hierarchical kernel descriptors on a large RGB-D dataset and demonstrated their ability to generate rich feature set from multiple sensor modalities, which is critical for boosting accuracy. In the future, we plan to investigate deep hierarchies of kernel descriptors to see whether more layers are helpful for object recognition.

References

[1] L. Bo, X. Ren, and D. Fox. Kernel Descriptors for Visual Recognition. In *NIPS*, December 2010. 1729, 1730, 1731, 1732, 1734

[2] L. Bo and C. Sminchisescu. Efficient Match Kernel between Sets of Features for Visual Recognition. In *NIPS*, 2009. 1730, 1731

[3] M. Carreira-Perpinan and G. Hinton. On Contrastive Divergence Learning. In *AISTATS*, 2005. 1729

[4] A. Coates, H. Lee, and A. Ng. An analysis of single-layer networks in unsupervised feature learning. In *NIPS*2010 Workshop on Deep Learning*, 2010. 1734, 1735

[5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 1729, 1734

[6] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009. 1733

[7] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE PAMI*, 32(9):1627–1645, 2009. 1729

[8] G. Hinton, S. Osindero, and Y. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006. 1729, 1730

[9] G. Hinton and R. Salakhutdinov. Reducing the Dimensionality of Data with Neural Networks. *Science*, 313(5786):504–507, July 2006. 1729

[10] G. Hua, M. Brown, and S. Winder. Discriminant embedding for local image descriptors. In *ICCV*, 2007. 1730

[11] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun. What is the best multi-stage architecture for object recognition? In *ICCV*, 2009. 1729, 1730

[12] A. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE PAMI*, 21(5), 1999. 1735

[13] K. Kavukcuoglu, M. Ranzato, R. Fergus, and Y. LeCun. Learning invariant features through topographic filter maps. In *CVPR*, 2009. 1729

[14] A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009. 1734

[15] K. Lai, L. Bo, X. Ren, and D. Fox. A Large-Scale Hierarchical Multi-View RGB-D Object Dataset. In *IEEE International Conference on on Robotics and Automation*, 2011. 1732, 1733, 1735

[16] Q. Le, J. Ngiam, Z. C. Chia, P. Koh, and A. Ng. Tiled convolutional neural networks. In *NIPS*. 2010. 1734

[17] H. Lee, A. Battle, R. Raina, and A. Ng. Efficient sparse coding algorithms. In *NIPS*, pages 801–808, 2006. 1730

[18] H. Lee, R. Grosse, R. Ranganath, and A. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *ICML*, 2009. 1730

[19] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60:91–110, 2004. 1729, 1730

[20] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE PAMI*, 24(7):971–987, 2002. 1731

[21] B. Olshausen and D. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, 1996. 1730

[22] J. Philbin, M. Isard, J. Sivic, and A. Zisserman. Descriptor learning for efficient retrieval. In *ECCV*, 2010. 1730

[23] PrimeSense. http://www.primesense.com/. 1732

[24] M. Ranzato, K. A., and G. Hinton. Factored 3-way restricted boltzmann machines for modeling natural images. In *AISTATS*, 2010. 1734

[25] M. Ranzato and G. Hinton. Modeling pixel means and covariances using factorized third-order boltzmann machines. In *CVPR*, 2010. 1730, 1734

[26] A. Torralba, R. Fergus, and W. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *IEEE PAMI*, 30(11):1958–1970, 2008. 1734

[27] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Guo. Locality-constrained linear coding for image classification. In *CVPR*, 2010. 1730

[28] C. Y and L. Saul. Kernel methods for deep learning. In *NIPS*, 2009. 1730

[29] Y. B. Y. LeCun, L. Bottou and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 1730

[30] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, 2009. 1729, 1730

[31] K. Yu and T. Zhang. Improved local coordinate coding using local tangents. In *ICML*, pages 1215–1222, 2010. 1730, 1734

[32] K. Yu, T. Zhang, and Y. Gong. Nonlinear Learning using Local Coordinate Coding . In *NIPS*, December 2009. 1730