

Data-driven Hierarchical Structure Kernel for Multiscale Part Based Object Recognition

Botao Wang, Hongkai Xiong, *Senior Member, IEEE*, Xiaoqian Jiang, Yuan F. Zheng, *Fellow, IEEE*

Abstract—Detecting generic object categories in images and videos is a fundamental issue in computer vision. However, it faces the challenges from inter- and intra-class diversity, as well as distortions caused by viewpoints, poses, deformations, etc. To solve object variations, this paper constructs a structure kernel and proposes a multi-scale part-based model incorporating the discriminative power of kernels. The structure kernel would measure the resemblance of part-based objects in three aspects: (1) The global similarity term to measure the resemblance of the global visual appearance of relevant objects; (2) The part similarity term to measure the resemblance of the visual appearance of distinctive parts; (3) The spatial similarity term to measure the resemblance of the spatial layout of parts. In essence, the deformation of parts in the structure kernel is penalized in a multi-scale space with respect to horizontal displacement, vertical displacement, and scale difference. Part similarities are combined with different weights, which are optimized efficiently to maximize the intra-class similarities and minimize the inter-class similarities by the normalized stochastic gradient ascent algorithm. Moreover, the parameters of the structure kernel are learned during the training process with regard to the distribution of the data in a more discriminative way. With flexible part sizes on scale and displacement, it can be more robust to the intra-class variations, poses and viewpoints. Theoretical analysis and experimental evaluations demonstrate that the proposed multi-scale part-based representation model with structure kernel exhibits accurate and robust performance, and outperforms state-of-the-art object classification approaches.

Index Terms—Object recognition, structure kernel, multi-scale part-based model, support vector machine, feature extraction.

I. INTRODUCTION

As a fundamental problem in computer vision and artificial intelligence, object recognition aims to detect objects belonging to certain object category. Recently, there has been rapidly growing interests in object recognition for a variety of practical applications, e.g., security surveillance and vehicle navigation [1]-[3]. Although significant progress has been made in the past few decades, it is still challenging for intra-class and inter-class variation, partial occlusion, and distortions from viewpoints, poses and deformations.

B. Wang and H. Xiong are with the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai 200240, China. e-mail: xionghongkai@sjtu.edu.cn.

X. Jiang is with the Division of Biomedical Informatics, University of California, San Diego, CA 92093, USA. e-mail: xljiang@ucsd.edu.

Y. F. Zheng is with the Department of Electrical and Computer Engineering, The Ohio State University, Columbus, OH 43210, USA. e-mail: zheng@ece.osu.edu.

The work was supported in part by the NSFC, under grants U1201255, 61271218, and 61228101.

Copyright (c) 2013 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

In the past decades, efforts on pattern recognition have been largely concentrated on finding the linear relation between data, such as template matching [18] and dictionary learning [19]. For data that exhibit nonlinearity in the original feature space, researchers attempt to map them into high dimensional Hilbert space by kernel functions in order to seek the linear analysis. Kernel methods [4]-[14], where kernels are symmetric bivariate functions that capture resemblance between input data, tackle the problem by discriminating patterns in high dimensional feature space. In fact, kernel framework provides generic learning machines by decoupling learning algorithms from data representation and solving the optimization problem in quadratic programming. Recent work on deep learning approaches [35], e.g., deep belief networks, deep Boltzmann machines and deep auto-encoders, has shown success in object recognition and classification. The deep architecture aims to learn complex mappings by transforming their inputs through multiple layers of nonlinear processing. Huang et al. [36] adopted convolutional deep belief networks to learn features for face verification, where local convolutional restricted Boltzmann machines model the global structure of an object class. Krizhevsky et al. [37] developed a large deep convolutional neural networks to learn object classes from a large dataset. Salakhutdinov et al. [38] suggested an architecture that combines deep learning networks with hierarchical nonparametric Bayesian models to learn new concepts from very few examples. Compared to the elegance of kernel methods, deep architectures involve nonlinear optimizations and many heuristics, e.g. searching large tunable parameter space of deep architectures. Hopefully, kernel machines has also been shown the benefits from deep learning [39].

Assume $\Phi(\mathbf{x}) : \mathcal{X} \rightarrow \mathcal{H}$ is a function that maps \mathbf{x} from original data space \mathcal{X} to a high dimensional Hilbert space \mathcal{H} . A kernel function k is capable of attaining the inner product of two mapped data in $\mathcal{H} : k(\mathbf{x}, \mathbf{x}') = \Phi(\mathbf{x}) \cdot \Phi(\mathbf{x}')$ in the original space without explicitly computing the mapped data. To ensure the existence of such mapping, a kernel function must be positive definite, which is also called the Mercer condition. Many successful applications have been accomplished by incorporating kernels, e.g., polynomial kernel and Gaussian kernel, which can attain better performance than linear analysis in the original data space. Although the positive definiteness guarantees the implicit mapping of the kernel, the remaining problem is how we can be sure that the mapped data in the high dimensional space are “more linear” than the original data space because the mapping function can not be obtained analytically in most cases. As a result, kernel methods are also referred to as “kernel tricks” because of this

uncertainty. Furthermore, illumination, partial occlusion, and non-rigid deformation in object recognition would make data exhibit even larger degree of nonlinearity. Hence, conventional kernel methods usually inherit limited improvements. The critical issue in designing a desirable kernel for complex data is that the kernel should be apt to capture the knowledge and expectation of the data.

At this moment, the motivation of this paper is to incorporate the prior knowledge of the data into the kernel so as to ensure the linearity in the mapped feature space to the largest extent. Commonly, there are three expectations on an object category. First, the instances in an object category are expected to have similar global appearances although local details may vary from one to another. Second, these objects are much likely to share some distinctive sub-regions, known as parts, that are closely related to their functions and categories. Third, these distinctive parts approximately but not necessarily occur at similar regions relative to the object, which is the main reason of the nonlinearity in object recognition. As a result, we are devoted to taking the three prior knowledge into kernel design in order to increase the discriminative capability of the proposed kernel for object recognition.

To exploit the discriminative capability of kernels for object recognition, two families of kernel methods have been studied in recent years, namely, the multiple kernel learning and local kernels. Multiple kernel learning approaches [9]-[14] combine multiple types of kernels and various kinds of features into a strong kernel to address the variations of the object class. MKL-DR [9] uses multiple kernel learning approaches for dimension reduction. It fuses multiple base kernels, each of which is based on an image descriptor, into the domain of kernel matrices with graph embedding. GS-MKL [10] introduces the groups between an object category as the an intermediate representation of the object category. The value of the GS-MKL is not only dependent on the similarity of the two images but also on the groups (subcategories) of the images. Sun et al. [11] proposed a forward feature-selection technique and a coarse-to-fine learning scheme to find a set of good exemplars for multiple kernel learning. Nilufar et al. [12] developed a multiple kernel learning method to weight and select the DoG scale space features by a shift invariant kernel.

The main drawback of the multiple kernel learning methods is the high computational complexity. On one hand, multiple image features have to be extracted, and each feature depicts certain aspect of image information. On the other hand, each feature is evaluated by a particular base kernel, and the final kernel is the weighted combination of the base kernels. These two steps introduce large dimension of features and intense computation of the kernel function.

Local kernels [4]-[8] aim to capture the resemblance of two sets of local features extracted from keypoints. A simple local kernel is the summation kernel [5] that calculates the sum of cross-similarities between all possible combinations of feature vectors. Its discriminant power is often poor because only a small number of good matchings of local features contribute to the kernel, which are often undermined by a large number of bad matchings. “Max kernel” [8] sums only the

similarities of the best matched feature vectors. Unfortunately, the max kernel is not a Mercer kernel, so that such implicit mapping to the high dimensional feature space may not exist. Circular-shift invariant kernel [7] takes the geometric configuration of a semigroup of keypoints into consideration. The semi-local constraint of a keypoint is defined to be the k neighboring angles spanned by the central keypoint and its k -nearest neighbors, which is invariant under circular-shifts. Context-dependent kernel [4] takes the context as a part of the alignment process in designing kernels. The “context” of a keypoint is defined to be its neighbors, which are indexed by the relative distances and orientations. The pyramid match kernel [6] maps unordered feature sets to multi-resolution histograms, and computes a weighted histogram intersection in order to find implicit correspondences based on the finest resolution histogram cell where a matched pair first appears.

There are two major limitations in the local kernels. The first limitation lies in the bag-of-features representation [15]-[17], on which the local kernels are defined. Existing keypoint detectors, e.g., [20]-[22], are very sensitive to background clutters, so a large number of keypoints may be located in the background objects. Therefore, the computational complexity of local kernels is quite large. The second limitation of local kernels is that they do not make full use of the features of objects. Keypoint descriptors only encode the local image features in the neighborhood of keypoints, but global features, which provide information of the global appearance, are often ignored. On the other hand, all keypoints have equivalent contributions to the local kernels, but in fact, some local patterns are more distinctive for certain object categories. Finally, the spatial configuration, which is a critical cue for object recognition, is often overlooked in local kernels. Although some local kernels, e.g., [4] [7], use the geometric information of keypoints in the kernel evaluation, these spatial configurations are only constrained in local groups of keypoints, which is very unstable owing to the inconsistency of keypoints.

To tackle these shortcomings of multiple kernel learning and local kernels, we are motivated to design a strong kernel that possesses the following four properties. First, the kernel should incorporate global features and local features in kernel evaluation. Global features capture the global visual information of the objects, but they are sensitive to occlusion and deformation. Local features are invariant to deformation and occlusion, but susceptible to the background clutters. It is desirable to incorporate global features and local features in a hierarchical manner to obtain the most informative observations of the data. Second, the kernel should be flexible and data-driven. Specifically, the parameters of the kernel should be automatically adapted to the underlying data, so that the kernel can be discriminative for different types of data. Third, the kernel should exploit the structure of data in a semantic level. For object recognition, the semantic information of an object class can be the global appearance, local distinctive regions, spatial layout, etc. Therefore, the kernel should take these aspects into account. Fourth, the kernel should be computed efficiently. To achieve this, the input data should be arranged in a structured way for efficient evaluation.

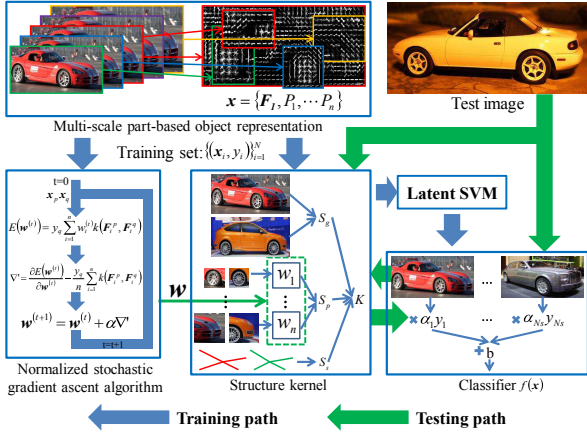


Fig. 1. Diagram of the structure kernel. The structure kernel measures the similarity of two objects in global visual appearance, part visual appearances and the spatial layout. The multi-scale part-based object model represents the parts in multiple scales relative to the object. The normalized stochastic gradient ascent algorithm optimizes the kernel parameters efficiently.

Hierarchical representation of images has been addressed in literature. For example, a three-level hierarchical model adopted nested fixed-size rectangular patches to represent images at different scales [31]. A family of hierarchical kernels, which are invariant to translation and some degree of scaling, were advanced to handle complex data representation in [34]. The Log-polar descriptor has ever been introduced to be invariant to scaling and rotation [32] [33]. Here, the proposed model is built on a two-layer, object-part hierarchy with more flexible and trainable patch sizes, which is very effective for object recognition. From the perspective of semantic structured modeling for object recognition, we tackle the invariance in image representation by exploiting high-level priors in generic object classes, which is more informative than textures [31] and local patterns [33].

In this paper, we propose part-based representation of objects with the structure kernel. Each part is described by a region-based feature vector, which is densely extracted from the rectangular window that covers the part. Compared to keypoint-based features, the desirable part-based representation with structured kernel would encode local visual patterns nearby, region-based feature characterizes the gradient distribution, and shape information of the patch. It is robust to background clutters, because in general, there are much less similar shapes than similar keypoints in the background. The diagram of the structure kernel is illustrated in Fig. 1. In the training path, objects are represented in a multi-scale part-based manner, in which the optimal positions and scales of parts are localized by a set of detectors. The proposed structure kernel is tailored to measure the similarity of such multi-scale part-based objects in a semantic level by increasing the weights of the discriminative parts and decreasing the weights of the indistinctive parts via a normalized stochastic gradient ascent algorithm. Finally, the object classifier is trained with the latent SVM, which is used to evaluate a new input in the testing path.

The contributions of this paper are three-fold. First, a structure kernel is constructed that incorporates the discriminative capability of local kernels into structured, and part-based

object models. Unlike holistic kernels, the proposed structure kernel is particularly designed for object recognition by incorporating the semantic information of object classes into kernel evaluation. Based on the fact that instances of an object class would exhibit similar global appearances, distinctive parts, and certain patterns of spatial layouts, the structure kernel hierarchically measures the similarity of two objects from these three aspects, i.e., global appearance, part appearance and the spatial layout of parts. On the other hand, the structure kernel also offers a flexible configuration of kernel parameters, so it can achieve robust discriminative capability for different object classes by fitting the parameters to the underlying data. In addition, the structure kernel gives an elegant measurement of the spatial similarity of two objects from the horizontal displacement, vertical displacement, and scale difference.

The second contribution of the paper is to develop a multi-scale part-based object model, which enriches the deformable part-based object model with multi-scale part representation. The proposed model represents an object with a global feature vector that encodes the visual appearance of the entire object, and several part feature vectors that encode the visual appearances of distinctive parts. In particular, parts are relaxed to be at neighboring scales of the object, which is more robust to poses, viewpoints, and intra-class variations, because the part sizes observed in real scenarios may be slightly different to the part sizes of the standard model. The spatial layout of parts is represented in a three-dimensional space consisting of a two-dimensional spatial coordinates and the scale as the third dimension. This multi-scale representation of parts is critical for the extraction of optimal object configurations.

The third contribution of the paper is to propose a learning algorithm to optimize the parameters of the structure kernel. Instead of fixed parameterized kernel, the parameters of the structure kernel are learned during the training process with regard to the distribution of the data in a more flexible and discriminative way. To be concrete, the objective function aims at simultaneously maximizing the intra-class similarities while minimizing the inter-class similarities. Therefore, the structure kernel can be adaptive to the structure and distribution of the underlying data, so that it is discriminative and robust for various kinds of object classes. A normalized stochastic gradient ascent algorithm is developed to efficiently solve the optimization problem.

The rest of the paper is organized as follows: Section II describes the structure kernel, involving the object model, local image feature, the definition of structure kernel and its generalization to the multiple component model. Section III discusses the optimization of the kernel parameters and the training of the classifier; Section IV provides the experimental results on the INRIA *person* dataset and PASCAL 2007 dataset; Section V draws the conclusions of the paper.

II. STRUCTURE KERNEL

A. Object Representation

In the proposed multi-scale part-based object model with structure kernel, an object is represented from coarse to fine with a global feature, a set of part features, and the spatial

layout of parts. An example from the *bicycle* class in the PASCAL 2007 dataset [25] is shown in Fig. 2, where the part sizes differ among different instances of the *bicycle* class because of viewpoint and intra-class variations. Hence, the optimal sizes of parts range within certain scales, as shown in Fig. 2 (c). In existing deformable models, parts would be captured either at a fixed interval of scale relative to the whole object [30] or at the same scale of the object [23], as in Fig. 3. It would lead to sub-optimal part representation.

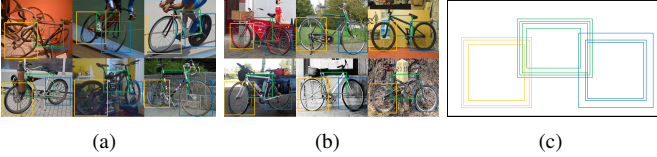


Fig. 2. In real scenarios, part sizes relative to the entire object vary within certain scales owing to viewpoint difference and intra-class variability. In the images, objects are normalized to the same size, and the optimal part sizes are labeled with bounding boxes. (a) Part sizes vary owing to viewpoint; (b) part sizes vary owing to intra-class variations; (c) the optimal part sizes of observations range within certain scales.

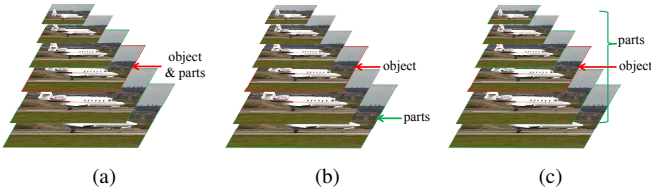


Fig. 3. Illustrations of different part representations of part-based models. (a) Parts are in the same scale of the object [23]; (b) parts are in the fixed interval of scale relative to the object [30]; (c) parts are in the neighboring scales of the object (proposed).

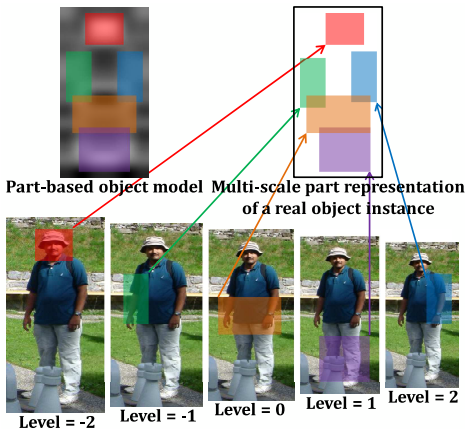


Fig. 4. Multi-scale part-based representation of objects. The scale of the object is denoted as *Level 0*, and parts are represented in the neighboring scales of the object, i.e., from *Level -2* to *Level 2*.

To be specific, an object is represented as a $n + 1$ tuple:

$$\mathbf{x} = (\mathbf{F}_0, P_1, \dots, P_n), \quad (1)$$

where \mathbf{F}_0 is the feature vector of the entire object and \mathbf{F}_i is the feature vector of part i . $P_i = (\mathbf{F}_i, \mathbf{g}_i)$, $i = 1, \dots, n$ are part models. $\mathbf{g}_i = (x_i, y_i, s_i)$ is the three dimensional vector denoting the spatial layout of part i , where (x_i, y_i) is the coordinate of the part location and s_i is the part scale. Both the location and scale of the part is measured with respect to the entire object for consistent measurement. Specifically,

if the scale of the object is S^0 , and the scale of the part s_i should satisfy $S^{-L} \leq s_i \leq S^L$, where S is the scale factor and L is the radius of the scale space. The part location (x, y) is normalized by the width and height of the entire object, respectively.

The proposed multi-scale part-based representation model can be illustrated in Fig. 4, where the scale of the entire object is denoted as *Level = 0*, and parts are located in the neighboring scales of the object, i.e., from *Level = -2* to *Level = 2*. The optimal scales of parts are obtained by searching from a range of neighboring scales relative to the object.

B. Local Features

Local features encode the visual appearance of the whole object and parts, and typical histogram of oriented gradients (HOG) [18] exhibits success in many object detection algorithms [24] [27]. Considering HOG is invariant to small translation and rotation of local shape, it is adopted as feature descriptor consisting of two steps: (1) weighted voting into spatial and orientation cells; (2) contrast normalization over overlapping spatial blocks.

In the first stage, local orientation histograms in terms of the gradient orientation $\theta(x, y)$ and the gradient magnitude $M(x, y)$ for each pixel via central difference are attained in local spatial regions called *cells*. An image is divided into 8×8 non-overlapping cells, and each pixel contributes a weighted vote for 9 contrast insensitive orientation bins evenly spaced over 0° to 180° , and 18 contrast sensitive orientation bins evenly spaced over 0° to 360° . To reduce aliasing, votes are interpolated tri-linearly between neighboring orientation bins and spatial cells. The gradient magnitude $M(x, y)$ is taken as the voting weight of a pixel. Each cell would generate a 9-D local histogram of oriented gradients of contrast insensitive orientation bins, and an 18-D local histogram of oriented gradients of contrast sensitive orientation bins. Namely, an $18 + 9 = 27$ dimensional local histogram of oriented gradients is generated for a cell. In the second stage, local histograms of oriented gradients are normalized in spatial blocks. Each block consists of 2×2 cells and blocks are overlapping, namely, each cell is covered by four blocks. The local histogram of oriented gradients in each cell is normalized by four ℓ^2 norms of the blocks that cover it. After normalization, each cell generates a $4 \times 9 = 36$ dimensional feature vector of contrast insensitive orientation bins, and a $4 \times 18 = 72$ dimensional local feature vector of contrast sensitive orientation bins. In total, a $36 + 72 = 108$ dimensional feature vector is generated for a cell after block normalization.

To reduce dimensionality, the 108 dimensional local feature vector is projected to 27 sums over 9 contrast insensitive and 18 contrast sensitive orientations, and 4 sums over different normalization factors. Finally, the 108 dimensional local feature vector in each cell is compressed into a $27 + 4 = 31$ dimensional local feature vector.

C. Definition of Structure Kernel

Given two part-based objects $\mathbf{x} = \{\mathbf{F}_0, P_1, \dots, P_n\}$ and $\mathbf{x}' = \{\mathbf{F}'_0, P'_1, \dots, P'_n\}$, the structure kernel \mathcal{K} measures their similarity in visual appearance and spatial layout.

Definition 1 (Global similarity). Let \mathbf{F}_0 and \mathbf{F}'_0 be the global feature vectors of two part-based represented objects \mathbf{x} and \mathbf{x}' . The global similarity term is defined as:

$$S_g = k(\mathbf{F}_0, \mathbf{F}'_0), \quad (2)$$

where $k(\cdot)$ is a pre-defined standard kernel called base kernel.

The global similarity term measures the resemblance of the visual appearance of the entire objects. It is analogous to the Dalal-Triggs detector, which is defined on a rectangular sliding window. Through various kinds of base kernels, the global similarity term can be more discriminative to the global shape of the object class.

Owing to the fact that global features are coarse and sensitive to partial occlusions and background clutters, we further define the part similarity term.

Definition 2 (Part similarity). Let $\{\mathbf{F}_i\}_{i=1}^n$ and $\{\mathbf{F}'_i\}_{i=1}^n$ be two sets of part feature vectors of two part-based represented objects \mathbf{x} and \mathbf{x}' . The part similarity term is defined as:

$$S_p = \sum_{i=1}^n w_i k(\mathbf{F}_i, \mathbf{F}'_i), \quad (3)$$

where $\{w_i\}_{i=1}^n$ are part weights, and $k(\cdot)$ is the base kernel.

The part similarity term measures the resemblance of the appearance of distinctive parts. For each pair of corresponding parts \mathbf{F}_i and \mathbf{F}'_i , the visual similarity would be measured as $k(\mathbf{F}_i, \mathbf{F}'_i)$. In this paper, the total part similarity term is obtained as the weighted combination of each individual part similarity. The weight w_i of a part reflects the significance of the part to the recognition of the object class, and $\{w_i\}_{i=1}^n$ shall be determined during training in Section III-A. The similarity of parts is reflected by both visual appearance and spatial layout.

Definition 3 (Spatial similarity). Let $\{g_i\}_{i=1}^n$ and $\{g'_i\}_{i=1}^n$ be the locations of parts of two part-based represented objects \mathbf{x} and \mathbf{x}' , where $g = (x, y, s)$, (x, y) is the position and s is the scale of the part relative to the object. The spatial similarity term is defined as:

$$S_s = \sum_{i=1}^n \exp \{-\gamma |g_i - g'_i|^2\}. \quad (4)$$

The spatial similarity between a pair of corresponding parts is measured by a radial basis function, which preserves the positive-definiteness of the structure kernel. Compared to existing part deformation on the horizontal and vertical penalty [30][4], it takes the part similarity of the scale space into account, i.e., the difference of the part areas. Given a pair of corresponding parts P_i and P'_i , their spatial similarity would be measured in three aspects: horizontal displacement, vertical displacement, and scale difference. It provides more robustness to viewpoint variations and intra-class differences.

Let $\mathcal{F}_i, i = 0, 1, \dots, n$ be the feature space of the whole object ($i = 0$) and parts ($i = 1, \dots, n$), and $\mathcal{G} = \mathbb{R}^3$ is the feature space of three dimensional part locations. The feature space of part i can be denoted as $\mathcal{P}_i = \mathcal{F}_i \times \mathcal{G}, i = 1, \dots, n$. In sum, the feature space of an object

is $\mathcal{X} = \mathcal{F}_0 \times \mathcal{P}_1 \times \dots \times \mathcal{P}_n$. Given two part-based represented objects \mathbf{x}_i and \mathbf{x}_j , a structure kernel $\mathcal{K} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is defined as:

Definition 4 (Structure Kernel). Let \mathcal{X} be the input space of part-based objects, and $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$ are two part-based represented instances. The structure kernel $\mathcal{K} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ between \mathbf{x} and \mathbf{x}' is defined as

$$\begin{aligned} \mathcal{K}(\mathbf{x}, \mathbf{x}') &= S_g + S_p + \lambda S_s = k(\mathbf{F}_0, \mathbf{F}'_0) \\ &+ \sum_{i=1}^n w_i k(\mathbf{F}_i, \mathbf{F}'_i) + \lambda \sum_{i=1}^n \exp \{-\gamma (g_i - g'_i)^2\}, \end{aligned} \quad (5)$$

where $k(\cdot) : \mathcal{F}_i \times \mathcal{F}_i \rightarrow \mathbb{R}$ is the base kernel, $\lambda \in \mathbb{R}^+$ is a kernel parameter that balances the relative weights between the appearance similarity and the spatial similarity.

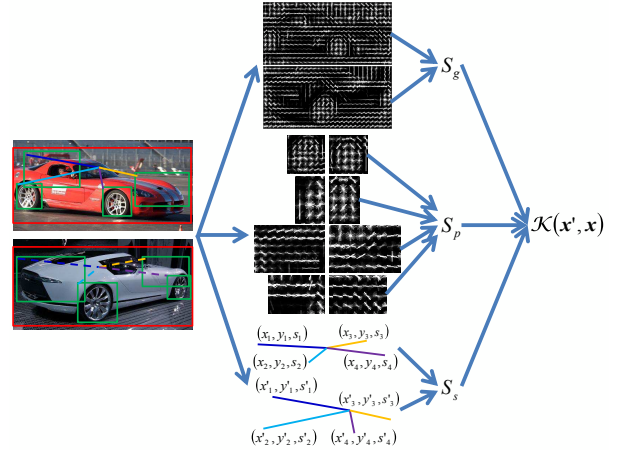


Fig. 5. Illustration of the structure kernel. Structure kernel measures the similarity of two part-based objects in three aspects: global visual appearance, part visual appearances and spatial layout of parts.

The idea of the structure kernel is illustrated in Fig. 5. Given two part-based represented objects, the structure kernel measures their similarity in three aspects: global similarity, part similarity, and spatial similarity. The global similarity term measures the resemblance of the global shapes of the objects. The part similarity term measures the resemblance of the visual appearance of the corresponding parts. The spatial similarity term measures the resemblance of the spatial layouts of parts.

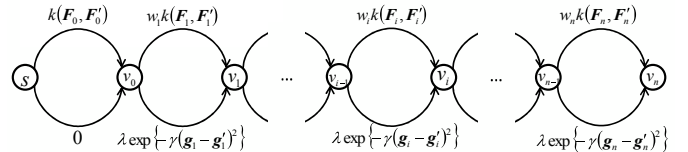


Fig. 6. The graph representation of the structure kernel.

On the other hand, the proposed structure kernel can also be interpreted as a kernel graph, which is illustrated in Fig. 6. To be specific, the structure kernel can be represented as a directed graph $G = (V, E)$, where V is the set of vertices and E is the set of directed edges. In the graph G , there are $n + 2$ vertices,

including a source vertex s of in-degree 0, followed by $n + 1$ vertices $\{v_i\}_{i=0}^n$ of in-degree 2 representing the object and n distinctive parts. The last part vertex v_n is also the sink vertex of out-degree 0 of the graph. The evaluation of the structure kernel is to calculate the energy of the directed graph from the leftmost to the rightmost. For each vertex $v_i \in \{v_i\}_{i=0}^n$, there are two edges directed to it from the previous vertex. The upper edges encode the weighted visual similarities of parts and the whole object, and the lower edges encode the spatial similarities of parts. The energy of a vertex is given by summing over the energy of all edges reaching it and the energy of its previous vertex. The source vertex s has zero energy. Therefore, the structure kernel can be efficiently calculated in this cascading manner when the number of parts are large.

To guarantee the existence of the high dimensional reproducing kernel Hilbert space, the structure kernel \mathcal{K} must satisfy the Mercer condition: For any selection of examples $\mathbf{x}_1, \dots, \mathbf{x}_m \in \mathcal{X}$, the Gram matrix \mathbf{K} of the structure kernel $\mathcal{K} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$, which is defined as $\mathbf{K}(i, j) = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$, is positive definite.

Proposition 1. *The structure kernel is a Mercer kernel.*

Proof. Recall that a matrix \mathbf{K} is positive definite if and only if $\alpha^T \mathbf{K} \alpha > 0$ for all non-zero vector α . We denote the Gram matrices for the base kernel k as $\mathbf{K}_i, i = 0, \dots, n$, and the Gram matrix for $d_i(\mathbf{g}_i, \mathbf{g}'_i) = \exp\{-\gamma(\mathbf{g}_i - \mathbf{g}'_i)^2\}$ as $\mathbf{D}_i, i = 0, \dots, n$. As the base kernel k is positive definite and $\exp\{-\gamma(\mathbf{g}_i - \mathbf{g}'_i)^2\}$ is a radial basis function, \mathbf{K}_i and \mathbf{D}_i are positive definite. Thus, for any m -dimensional non-zero vector α ,

$$\begin{aligned} \alpha^T \mathbf{K} \alpha &= \alpha^T \left(\sum_{i=1}^n \mathbf{K}_i + \lambda \sum_{i=1}^n \mathbf{D}_i \right) \alpha \\ &= \sum_{i=0}^n \alpha^T \mathbf{K}_i \alpha + \lambda \sum_{i=1}^n \alpha^T \mathbf{D}_i \alpha \geq 0. \end{aligned} \quad (6)$$

Hence, the structure kernel satisfies the Mercer condition. \square

D. Generalization

Although the structure kernel is tailored for the multi-scale part-based model, it is also flexible to be utilized in other object detection frameworks with certain modifications. In the simplest case, the structure kernel can be degraded to one single holistic kernel by setting the weights of the part similarity terms $\{w_i\}_{i=1}^n$ and λ to zeros. That is, there is only the global similarity term in the structure kernel, which is evaluated by a base kernel k :

$$\mathcal{K}(\mathbf{x}, \mathbf{x}') = k(\mathbf{F}_0, \mathbf{F}'_0). \quad (7)$$

Eq. (7) can be used to evaluate rigid template models [18] [27] for object recognition, and any other pattern recognition tasks that need to measure the similarity of two inputs with kernels.

If we use linear kernel as the base kernel and use quadratic loss function $l(x, y) = a_1 x^2 + a_2 x + a_3 y^2 + a_4 y + a_5$, where

$\{a_i\}_{i=1}^5$ are constants, to penalize the horizontal and vertical deformations, the structure kernel becomes

$$\mathcal{K}(\mathbf{x}, \mathbf{x}') = \sum_{i=0}^n \mathbf{F}_i \cdot \mathbf{F}'_i + \sum_{i=1}^n l(x_i - x'_i, y_i - y'_i), \quad (8)$$

where $w_i = 1$ for $i = 1, \dots, n$ and $\lambda = 1$, and the scale differences are not penalized. Eq.(8) is exactly in the same form as the filter function of the deformable part model [30]. Therefore, the structure kernel can also be used to evaluate the discriminatively trained part based models.

The structure kernel becomes a ‘‘max’’ kernel [8] if the spatial similarity term is omitted, i.e., $\lambda = 0$, and part similarities have unity weights, i.e., $w_i = 1$ for $i = 1, \dots, n$.

$$\mathcal{K}(\mathbf{x}, \mathbf{x}') = \sum_{i=0}^n k(\mathbf{F}_i, \mathbf{F}'_i). \quad (9)$$

The definition of a ‘‘max’’ kernel is

$$K_{\max} = \frac{1}{2} \sum_{i=0}^n \max_j k(\mathbf{F}_i, \mathbf{F}'_j) + \frac{1}{2} \sum_{j=0}^n \max_i k(\mathbf{F}_i, \mathbf{F}'_j) \quad (10)$$

For the structure kernel, the part similarity can be maximized if and only if the same parts are chosen, that is,

$$\max_{j=0, \dots, n} k(\mathbf{F}_i, \mathbf{F}'_j) = k(\mathbf{F}_i, \mathbf{F}'_i). \quad (11)$$

Therefore,

$$\begin{aligned} \mathcal{K}(\mathbf{x}, \mathbf{x}') &= \sum_{i=0}^n k(\mathbf{F}_i, \mathbf{F}'_i) \\ &= \frac{1}{2} \sum_{i=0}^n k(\mathbf{F}_i, \mathbf{F}'_i) + \frac{1}{2} \sum_{j=0}^n k(\mathbf{F}_j, \mathbf{F}'_j) \\ &= \frac{1}{2} \sum_{i=0}^n \max_j k(\mathbf{F}_i, \mathbf{F}'_j) + \frac{1}{2} \sum_{j=0}^n \max_i k(\mathbf{F}_i, \mathbf{F}'_j) \end{aligned} \quad (12)$$

E. Multiple component model

To handle the variations of instances in the same category caused by viewpoint and intra-class diversity, a popular solution is to train a multiple component model [28]-[30]. The basic idea is to divide an object class into several sub-classes, each of which represents either a sub-category (parrot/hummingbird/eagle) or a view from certain angle (front view/side view) of the class, and train a classifier for each sub-class. The proposed structure kernel can be implemented in a multiple component framework as well.

The diagram of the multiple component model with structure kernel is illustrated in Fig. 7. First, positive examples $X = \{\mathbf{x}_i\}_{i=1}^N$ are grouped into C sub-classes $\{X_c\}_{c=1}^C$ satisfying $X_1 \cup \dots \cup X_C = X$ and $\forall p \neq q, X_p \cap X_q = \emptyset$, where $X_c = \{\mathbf{x}_i\}_{i=1}^{N_c}, \sum_{i=1}^C N_c = N$. Clustering of positive examples can be based on the aspect ratios or the features of the positive examples by the K-means algorithm. Then, a specific structure kernel \mathcal{K}_c is defined on each sub-class X_c , and a corresponding classifier $f_{\mathcal{K}_c}(\mathbf{x})$ can be trained. For evaluation, an input object \mathbf{x} is processed by the C classifiers to produce C component scores $f_{\mathcal{K}_1}(\mathbf{x}), \dots, f_{\mathcal{K}_C}(\mathbf{x})$. The final classification score is the maximum of the component scores.

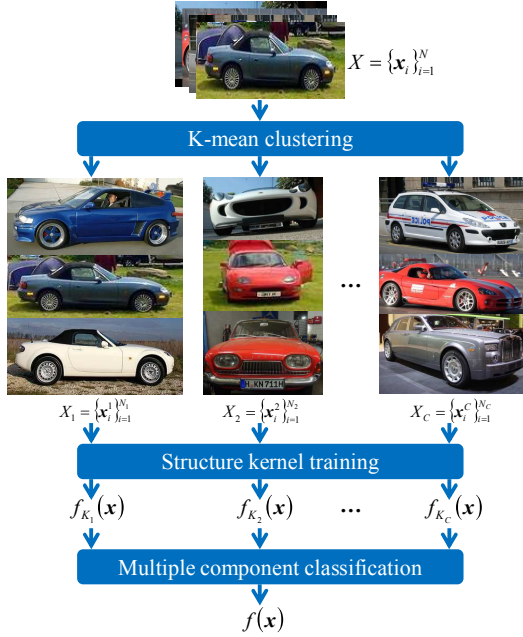


Fig. 7. Multiple component model with structure kernel.

III. TRAINING OF CLASSIFIER

The proposed classifier evaluates an input data \mathbf{x} with the following discriminant function

$$f(\mathbf{x}) = \boldsymbol{\rho} \cdot \phi(\mathbf{x}) + b, \quad (13)$$

where $\phi(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}^h$ is a function that maps the input data from its original d -dimensional space to h -dimensional Hilbert space, where h is greater than d and possibly infinite. $\boldsymbol{\rho} \in \mathbb{R}^h$ is a vector in the h -dimensional space, and b is a constant. However, the explicit analytic expression of $\phi(\cdot)$ can not be obtained, and its inner product can be calculated by the structure kernel:

$$\mathcal{K}(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x}) \cdot \phi(\mathbf{x}'). \quad (14)$$

As the structure kernel \mathcal{K} is parameterized by $\mathbf{w} = \{w_i\}_{i=1}^n$, where n is the number of parts, the mapping function $\phi(\cdot)$ is also parameterized by \mathbf{w} . Finally, the classifier $f(\mathbf{x})$ is parameterized by $\boldsymbol{\rho}$, b and \mathbf{w} . It is supposed to produce high value if \mathbf{x} belongs to the object class of interest.

The optimization of Eq. (13) over $\boldsymbol{\rho}$, b and \mathbf{w} is intractable, because the relation of $\phi(\cdot)$ and \mathbf{w} is unknown. Hopefully, with the help of the structure kernel, a two-step iterative optimization algorithm is proposed to train the classifier in Eq. (13). In the first step, we optimize $\phi(\cdot)$ over \mathbf{w} so that the distance between two positive examples is small, and the distance between a positive example and a negative example is large in the h -dimensional space. In the second step, we fix \mathbf{w} , and optimize a SVM formulation over $\boldsymbol{\rho}$ and b to get a maximum margin classifier. By repeating the two steps recursively, the optimal classifier parameters and kernel parameters can be obtained in a joint manner. Section III-A introduces the optimization of the kernel parameters, and Section III-B introduces the optimization of the classifier parameters.

A. Optimizing kernel parameters

Given the unordered training set $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, where $\mathbf{x}_i \in \mathcal{X}$ and $y_i \in \{-1, +1\}$, we expect the structure kernel to produce high values on intra-class objects (implying high similarity) and low values on inter-class objects (implying low similarity). Let \mathbf{x}_p and \mathbf{x}_q be two objects, their similarity measured by the structure kernel is

$$\begin{aligned} \mathcal{K}(\mathbf{x}_p, \mathbf{x}_q) = & k(\mathbf{F}_0^{(p)}, \mathbf{F}_0^{(q)}) + \sum_{i=1}^n w_i k(\mathbf{F}_i^{(p)}, \mathbf{F}_i^{(q)}) \\ & + \lambda \sum_{i=1}^n \exp\{-\gamma(\mathbf{g}_i^{(p)} - \mathbf{g}_i^{(q)})^2\} \end{aligned} \quad (15)$$

For all pairs of positive examples \mathbf{x}_p and \mathbf{x}_q satisfying $y_p = +1$ and $y_q = +1$, we expect high similarity between them. The formula of maximizing the intra-class similarity is attained as

$$\begin{aligned} \max_{\mathbf{w}} E^+(\mathbf{w}) = & \sum_{p=1}^N \sum_{q=p+1}^N \mathcal{K}(\mathbf{x}_p, \mathbf{x}_q) \\ \text{s.t.} \quad & \begin{cases} \sum_{i=1}^n w_i = 1 \\ w_i \geq 0, i = 1, \dots, n \\ y_p + y_q = +2 \end{cases} \end{aligned} \quad (16)$$

where $\mathbf{w} = (w_1, \dots, w_n)$. As the global similarity term and spatial similarity term in the structure kernel is irrelevant of \mathbf{w} , Eq. (16) is consequently equivalent to

$$\begin{aligned} \max_{\mathbf{w}} E^+(\mathbf{w}) = & \sum_{p=1}^N \sum_{q=p+1}^N \sum_{i=1}^n w_i k(\mathbf{F}_i^p, \mathbf{F}_i^q) \\ \text{s.t.} \quad & \begin{cases} \sum_{i=1}^n w_i = 1 \\ w_i \geq 0, i = 1, \dots, n \\ y_p + y_q = +2 \end{cases} \end{aligned} \quad (17)$$

where only the part similarity term is involved. In Eq. (17), we maximize the sum of all possible cross similarities of intra-class objects, while satisfying $\sum_{i=1}^n w_i = 1$ and the positiveness of w_i .

On the other hand, if \mathbf{x}_p is a positive example and \mathbf{x}_q is a negative example, we expect low similarity between them. Likewise, the formula of minimizing the inter-class similarity is

$$\begin{aligned} \min_{\mathbf{w}} E^-(\mathbf{w}) = & \sum_{p=1}^N \sum_{q=p+1}^N \sum_{i=1}^n w_i k(\mathbf{F}_i^p, \mathbf{F}_i^q) \\ \text{s.t.} \quad & \begin{cases} \sum_{i=1}^n w_i = 1 \\ w_i \geq 0, i = 1, \dots, n \\ y_p + y_q = 0 \end{cases} \end{aligned} \quad (18)$$

Since minimizing the similarities between two inter-class objects is equivalent to maximizing its negative similarities, Eq. (17) and Eq. (18) can be unified by multiplying the part

similarity term $\sum_{i=1}^n w_i k(\mathbf{F}_i^p, \mathbf{F}_i^q)$ by $y_q y_p$. Thus, the optimal values for the part weights $\{w_i\}_{i=1}^n$ is the maximizer of

$$\begin{aligned} \max_{\{w_i\}_{i=1}^n} E(\mathbf{w}) &= \sum_{p=1}^N \sum_{q=p+1}^N y_p y_q \sum_{i=1}^n w_i k(\mathbf{F}_i^p, \mathbf{F}_i^q) \\ \text{s.t.} \quad &\begin{cases} \sum_{i=1}^n w_i = 1 \\ w_i \geq 0, i = 1, \dots, n \\ y_p + y_q \geq 0 \end{cases} \end{aligned} \quad (19)$$

Eq. (19) maximizes the cross-similarities between two positives examples and minimizes the cross-similarities between a positive example and a negative example. For two negative examples, we have no prior expectation about whether they are alike or not. Therefore, the evaluation of two negative objects is not involved in Eq. (19).

To solve Eq. (19) efficiently, we develop a normalized stochastic gradient ascent algorithm. Obviously, the gradient of $E(\mathbf{w})$ in Eq. (19) with respect to \mathbf{w} is

$$\frac{\partial E(\mathbf{w})}{\partial \mathbf{w}} = \begin{bmatrix} \sum_{p=1}^N \sum_{q=p+1}^N y_p y_q k(\mathbf{F}_1^p, \mathbf{F}_1^q) \\ \dots \\ \sum_{p=1}^N \sum_{q=p+1}^N y_p y_q k(\mathbf{F}_n^p, \mathbf{F}_n^q) \end{bmatrix}. \quad (20)$$

In the standard gradient ascent method for unconstrained maximization problems, \mathbf{w} is updated towards the gradient ascent direction in each step by

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + \alpha \frac{\partial E(\mathbf{w}^{(t)})}{\partial \mathbf{w}^{(t)}}, \quad (21)$$

where α is a small positive constant. However, the computation of the gradient is very computationally intensive in Eq. (20), which includes all the combinations of inter-class examples and intra-class examples.

In the normalized stochastic gradient ascent algorithm, we stochastically sample two examples \mathbf{x}_u and \mathbf{x}_v in each iteration, at least one of which is a positive example (note that the formula does not contain negative-negative similarity). We approximate the gradient of Eq. (20) with the sub-gradient of the samples

$$\nabla = y_v y_u [k(\mathbf{F}_1^u, \mathbf{F}_1^v), k(\mathbf{F}_2^u, \mathbf{F}_2^v), \dots, k(\mathbf{F}_n^u, \mathbf{F}_n^v)]^T, \quad (22)$$

and in each iteration, we move \mathbf{w} a step towards the gradient ascent direction.

Notice Eq. (19), the constraint $\sum_{i=1}^n w_i = 1$ may be violated during the update of \mathbf{w} . Thus, we normalize the gradient ∇ as

$$\nabla' = y_v y_u \begin{bmatrix} k(\mathbf{F}_1^u, \mathbf{F}_1^v) - \frac{1}{n} \sum_{i=1}^n k(\mathbf{F}_i^u, \mathbf{F}_i^v) \\ \dots \\ k(\mathbf{F}_n^u, \mathbf{F}_n^v) - \frac{1}{n} \sum_{i=1}^n k(\mathbf{F}_i^u, \mathbf{F}_i^v) \end{bmatrix}. \quad (23)$$

In each iteration, if the initial $\mathbf{w}^{(t)}$ satisfies $\sum_{i=1}^n w_i^{(t)} = 1$, we have $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + \alpha \nabla'$ and

$$\begin{aligned} \sum_{i=1}^n w_i^{(t+1)} &= \sum_{i=1}^n \left(k(\mathbf{F}_i^u, \mathbf{F}_i^v) - \frac{1}{n} \sum_{j=1}^n k(\mathbf{F}_j^u, \mathbf{F}_j^v) \right) \\ &\quad \cdot \alpha y_v y_u + \sum_{i=1}^n w_i^{(t)} = \sum_{i=1}^n w_i^{(t)} + \alpha \cdot 0 = 1. \end{aligned} \quad (24)$$

Given an initial $\mathbf{w}^{(0)}$ satisfying $\sum_{i=1}^n w_i^{(0)} = 1$, it can be guaranteed that the constraint is always satisfied during the iterations of the normalized stochastic gradient ascent method.

Next, we prove that the energy function $E(\mathbf{w})$ on the samples \mathbf{x}_u and \mathbf{x}_v would keep increasing by the normalized stochastic gradient ascent algorithm.

$$\begin{aligned} E(\mathbf{w}^{(t+1)}) - E(\mathbf{w}^{(t)}) &= y_v y_u \sum_{i=1}^n w_i^{(t+1)} k(\mathbf{F}_i^u, \mathbf{F}_i^v) - y_v y_u \sum_{i=1}^n w_i^{(t)} k(\mathbf{F}_i^u, \mathbf{F}_i^v) \\ &= \sum_{i=1}^n \left[\alpha y_v y_u \left(k(\mathbf{F}_i^u, \mathbf{F}_i^v) - \frac{1}{n} \sum_{j=1}^n k(\mathbf{F}_j^u, \mathbf{F}_j^v) \right) \right] \\ &\quad \cdot y_v y_u k(\mathbf{F}_i^u, \mathbf{F}_i^v) \\ &= \alpha y_v^2 y_u^2 \left[\sum_{i=1}^n k(\mathbf{F}_i^u, \mathbf{F}_i^v) k(\mathbf{F}_i^u, \mathbf{F}_i^v) \right. \\ &\quad \left. - \frac{1}{n} \sum_{i=1}^n k(\mathbf{F}_i^u, \mathbf{F}_i^v) \sum_{i=1}^n k(\mathbf{F}_i^u, \mathbf{F}_i^v) \right] \geq 0. \end{aligned} \quad (25)$$

As the root square mean is always greater than or equal to the arithmetic mean, the increment of $E(\mathbf{w})$ in every iteration is always non-negative. In turn, the optimal solution of \mathbf{w} can be obtained with sufficient number of samples.

B. Training classifier

The object classifier based on the proposed structure kernel can be trained in a semi-supervised way with a latent SVM. The positive examples are only labeled with bounding boxes covering the whole object, but parts are not labeled. During training, the most distinctive parts can be automatically identified, and their locations are treated as latent variables which are iteratively optimized with the classifier.

The set of positive images is denoted as $T_P = \{I_i^P\}_{i=1}^{N_P}$, and each positive image patch contains only one object of interest. The set of negative images is denoted as $T_N = \{I_i^N\}_{i=1}^{N_N}$, and each negative image has no instances of the object category. Candidate objects and parts are extracted by a global detector $D_0 = (\beta_0, b_0)$ and n part detectors $\{D_i = (\beta_i, b_i)\}_{i=1}^n$. Both the global detector and part detectors are linear SVM classifiers, and an image patch \mathbf{F}_i is scored by

$$f_{D_i}(\mathbf{F}_i) = \beta_i \cdot \mathbf{F}_i + b_i, i = 0, \dots, n. \quad (26)$$

Given the structure kernel \mathcal{K} and the training set $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, where $\mathbf{x}_i \in \mathcal{X}$ and $y_i \in \{-1, +1\}$, a new input $\mathbf{x} \in \mathcal{X}$ is scored by the discriminate function $f(\mathbf{x})$ parameterized by $\{\alpha_i\}_{i=1}^N$ and b :

$$f(\mathbf{x}) = \sum_{i=1}^N \alpha_i y_i \mathcal{K}(\mathbf{x}_i, \mathbf{x}) + b, \quad (27)$$

or equivalently,

$$f(\mathbf{x}) = \sum_{i=1}^{N_s} \alpha_{si} y_{si} \mathcal{K}(\mathbf{x}_{si}, \mathbf{x}) + b, \quad (28)$$

where the subscript s denotes support vectors, because $\alpha = 0$ for all non-support vectors.

In the training stage, we determine the global detector and part detectors $\{D_i\}_{i=0}^n$, $\{\alpha_i\}_{i=1}^n$ and the bias term b . The procedure of the training algorithm is shown in Algorithm 1.

Algorithm 1: Training of classifier

Input: Positive images: $T_P = \{I_i^P\}_{i=1}^{N_P}$
 Negative images: $T_N = \{I_i^N\}_{i=1}^{N_N}$
 Part number: n

Output: Global detector and part detectors $\{D_i\}_{i=0}^n$
 Classifier: f

$D_0 = \text{train_linear_SVM}(T_P, T_N);$
 $\{\beta_i\}_{i=1}^n = \text{part_extraction}(\beta_0, n);$
for $i = 1$ **to** n **do**
 | $P_i = \text{maximum_response}(\beta_i, T_P);$
 | $D_i = \text{train_linear_SVM}(P_i, T_N);$
end
for $r = 1$ **to** *relabel* **do**
 | $[X_P, X_N] = \text{relabel}(X_P, X_N, f, \{D_i\}_{i=0}^n);$
 | $\mathbf{w} = \text{optimal_part_weights}(X_P, X_N);$
 | $X'_P \subset X_P, X'_N \subset X_N;$
 | **for** $m = 1$ **to** *datamine* **do**
 | | $f = \text{train_classifier}(X'_P, X'_N);$
 | | $[X'_P, X'_N] = [X'_P, X'_N] - \text{easy examples};$
 | | $[X'_P, X'_N] = [X'_P, X'_N] + \text{hard examples};$
 | **end**
end

1) *Training detectors and extracting parts:* The global detector is trained with a linear SVM, which is similar to [18]. First, the aspect ratio of the global detector is the mean of the aspect ratios of the bounding boxes in the positive images. The area of global detector is not larger than 80% of the bounding boxes so that it can be put in most images to search for the positive examples. Second, all the positive images and negative images are resized to the size of the global detector, and HOG features are computed for all the images, which is described in Section II-B. Finally, a linear SVM classifier D_0 is trained with the positive and negative feature vectors.

After the initial global detector is trained, we run the global detector on the image pyramid of each positive image I_i^P . The image patch that produces the highest response serves as a new positive example in place of I_i^P . To ensure the effectiveness of the global detector, it is re-trained with the new training set, and negative patches are also randomly generated from the original negative image set. Eventually, the final global detector is obtained. Fig. 8 (b) visualizes the global detector of the *person* class by training with INRIA *person* dataset.

Parts are defined as sub-regions of objects which have similar visual appearance in the object class. An image patch F is scored by the global detector D_0 as $\beta_0 \cdot F + b_0$. As

all elements in the feature vector F are non-negative, a larger value in β_0 would increase the probability that the image patch belongs to the object category. In another word, the regions in β_0 that have high values are more distinctive in identifying the object category, therefore they are considered as “parts”.

Once the global detector is obtained, distinctive parts can be found by localizing high energy regions in β_0 , which is illustrated in Fig. 8 (c). To be specific, given the number of parts n , a set of rectangular mean filters are generated. These filters have different aspect ratios, but have similar areas, so that n parts can cover around 80% of the area of the entire object. Then, β_0 is convoluted with all mean filters, and the filter responses undergo a non-maximum suppression to exclude overlapping responses. Finally, the n (or $n - 1$ in some cases) symmetric regions that give the highest responses are chosen as parts, which is demonstrated in Fig. 8 (a).

The initial part detectors $\{\beta_i\}_{i=1}^n$ are defined to be the corresponding coefficients in β_0 . Similar to the re-training of the global detector, each initial part detector extracts a highest-response patch in each positive image, and the final part detector is trained with this new training set.

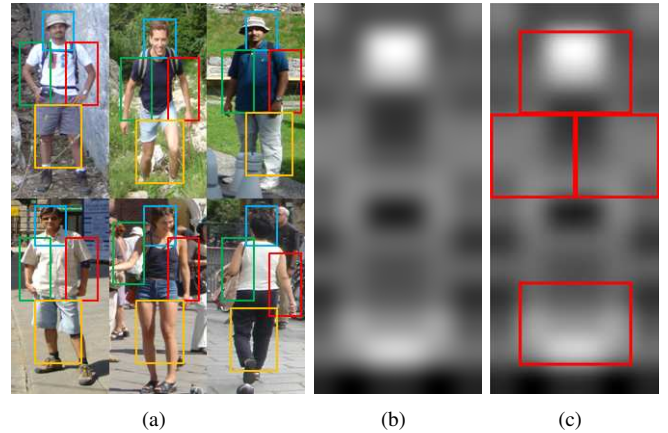


Fig. 8. (a) Parts are sub-regions in an object with similar visual appearances in the object category; (b) Global detector of the “person” category; (c) four parts are detected by localizing high energy regions in the global detector.

2) *Training Classifier:* We train the classifier with a latent SVM, which treats the configuration of the object and parts as latent variables. In each round, we relabel the training set by detecting the best configuration of each example with the global detector, part detectors and the object classifier, and use this new training set to re-train the classifier.

Specifically, for an image F , the responses of the global detector and part detectors at multiple scales can be denoted as $R_i^l = F^l * D_i, i = 0, \dots, n$, where F^l is the feature map of the image at scale l . Subsequently, a set of positive patches $P_i = \{P_{ij}\}_{j=1}^m, i = 0, \dots, n$ can be obtained, where P_{ij} is a patch at location \mathbf{p} and scale l satisfying $R_i^l(\mathbf{p}) > 0$. For a global location $P_0 \in \mathbf{P}_0$, the optimal object configuration $\mathbf{x} = (F_0, P_1, \dots, P_n)$ can be obtained by

$$\begin{aligned} & \max f(\mathbf{x}) \\ \text{s.t. } & \begin{cases} P_i \in \mathbf{P}_i, i = 1, \dots, n \\ |l_i - l_0| \leq L, i = 1, \dots, n \end{cases} \end{aligned} \quad (29)$$

That is, the optimal part is the one in the patch set that gives the highest classification score. For the positive examples, the relabeled objects should overlap with the bounding box at least 50%, and for the initial positive examples, where the object classifier is not available, the optimal parts are the ones within the bounding box that gives the highest responses.

Once the configuration of parts is fixed, the problem becomes training a standard kernelized SVM classifier:

$$\begin{aligned} \max_{\alpha_i} & -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) + \sum_{i=1}^N \alpha_i \\ \text{s.t.} & \begin{cases} 0 \leq \alpha_i \leq C, i = 1, \dots, N \\ \sum_{i=1}^N \alpha_i y_i = 0 \end{cases} \end{aligned} \quad (30)$$

With the matrix notation, it can be written as

$$\begin{aligned} \min_{\alpha} & \frac{1}{2} \alpha^T \mathbf{K} \alpha + c \alpha \\ \text{s.t.} & \begin{cases} \alpha \geq 0 \\ \alpha^T \mathbf{y} = 0 \end{cases} \end{aligned} \quad (31)$$

where \mathbf{K} is the Gram matrix of the structure kernel \mathcal{K} satisfying $\mathbf{K}(i, j) = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$. Eq. (31) is a quadratic programming problem and can be solved by the quadratic optimization algorithms.

Due to the limitation of computational capability and storage, we use a subset of the examples for training in each round, denoted as X'_P and X'_N . After the classifier is obtained, we remove easy examples from the current training set, and add hard examples from the rest of the examples to the current training set. Specifically, an easy example is correctly classified and also far from the separating hyperplane, i.e., \mathbf{x}_i is an easy example if $y_i f(\mathbf{x}_i) > 1 + \sigma$, where σ is a small positive constant. Otherwise, it is set as a hard example.

IV. EXPERIMENTAL RESULTS

We evaluate the proposed algorithm on two datasets: the INRIA person dataset [18] and the more challenging PASCAL 2007 dataset [25]. For each dataset, we shall compare our method with other state-of-the-art methods in both accuracy and computational complexity. The experiments are performed on a computer with 3.40GHz 8-core Intel Core i7-3770 CPU and Ubuntu 12.04 operating system.

A. Evaluation on INRIA person dataset

Initially, we use the INRIA *person* dataset [18] to evaluate the performance of the proposed structure kernel. The positive training set consists of 1,208 images and their left-right reflections, i.e., 2,416 images in all. The negative training set contains more than 10,000 image patches randomly sampled from 1,218 images which do not contain any instance of *person*. The test set is composed of 1,216 positive image patches and 12,160 negative image patches. The parameters of the classifier are: $\lambda = 0.1$, $C = 0.01$, $\gamma = 1$, part number is 4, the scale factor is 0.9, and the scale radius $L = 2$. The

structure kernel \mathcal{K} would be tested with three different base kernels:

- 1) Linear kernel: $k(\mathbf{F}, \mathbf{F}') = \mathbf{F} \cdot \mathbf{F}'$.
- 2) Polynomial kernel: $k(\mathbf{F}, \mathbf{F}') = (\mathbf{F} \cdot \mathbf{F}' + 1)^2$.
- 3) RBF kernel: $k(\mathbf{F}, \mathbf{F}') = \exp(-|\mathbf{F} - \mathbf{F}'|^2)$.

For comparison, we also test the Dalal-Triggs detector [18], deformable part-based model (DPM) [30], and three local kernels - summation kernel [5], max kernel [8] and p -kernel [7] - on the INRIA person dataset. The Dalal-Triggs detector trains a linear SVM on the HOG features of the training set. The detector of DPM on the INRIA person dataset is provided in [26], which contains one component with 6 parts. Therefore, only the testing time of their method is recorded. Local kernels are defined on two sets of unordered local features. Similar to [7] [17] [8], we use the Harris corner detector to extract keypoints, and encode the local feature of a keypoint with the SIFT descriptor [20]. The order of the p -kernel is 9, which is the same as the experiment in [7]. A linear kernel is served as the base kernel of the three local kernels. The average precision (AP), training time and testing time (in minutes) of the above methods are provided in Table I.

In general, the proposed structure kernel with three different base kernels obtains the best performance, and the three local kernels obtain the worst performances. Within the three base kernels in the structure kernel, the RBF base kernel achieves the best performance. Although DPM offers a more flexible representation of objects than the Dalal-Triggs detector, it gets slightly worse result. It can be derived from the fact that the intra-class variations and poses of the *person* images in INRIA person dataset are quite limited, in turn, a root filter is appropriate and discriminative enough to classify the *person* class. To some extent, the parts and deformation costs may result in unexpected errors. Within the three local kernels, the p -kernel gets the best performance, and the summation kernel gets the worst performance. As the summation kernel combines all pairs of features equivalently, good matches may be swamped by large number of bad matches. However, the p -kernel significantly shrinks the scores of the bad matches, so the good matches dominate the output. Therefore, p -kernel can get better than the summation kernel.

TABLE I
PERFORMANCE OF DIFFERENT DETECTORS

	AP	Training time	Testing time
SK (linear)	0.938	133	93
SK (polynomial)	0.946	142	97
SK (RBF)	0.954	140	96
DPM	0.917	N/A	77
Dalal-Triggs	0.927	87	52
Sum-kernel	0.811	192	174
Max-kernel	0.852	188	161
p -kernel	0.910	190	166

TABLE II
AVERAGE PRECISION OF DIFFERENT PART NUMBERS

Number of parts	3	4	5	6
Average precision	0.934	0.954	0.930	0.945

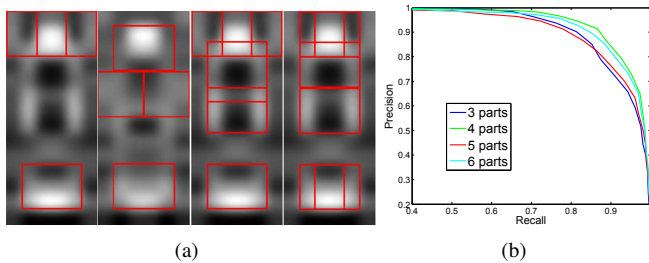


Fig. 9. Impact of part number on the performance of structure kernel. (a) Four different configurations of parts; (b) precision-recall curves.

To evaluate the impact of the part number on the performance of the structure kernel, we change the part number from 3 to 6 as shown in Fig. 9 (a) and train classifiers where k is a RBF kernel. Fig. 9 (b) shows the precision recall curves, and the average precision values are listed in Table II. It can be seen that the structure kernel is fairly robust with different part numbers, and the part number with the best performance is 4 for the INRIA person dataset.

B. Evaluation on PASCAL 2007 dataset

In addition to the INRIA person dataset, we also evaluate our method on a much more challenging dataset – the PASCAL 2007 dataset [25], which is composed of 20 categories of objects. The configuration of the training set and the test set of the PASCAL 2007 dataset can be found in [25]. For each object category, we divide the object into 4 parts and train a classifier using the proposed structure kernel with the RBF kernel as the base kernel, $\lambda = 0.1$, $\gamma = 1$ and $C = 0.01$. The global detector for each category is visualized in Fig. 10.

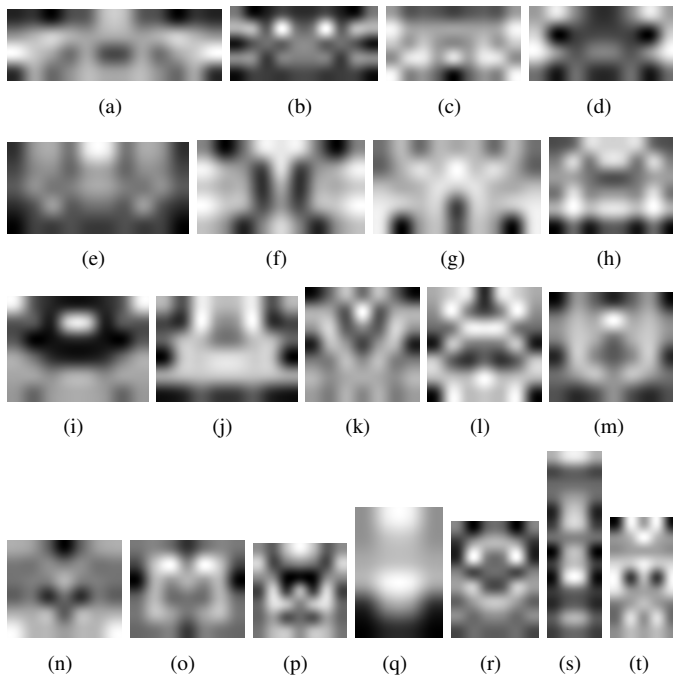


Fig. 10. Visualization of the global detectors of 20 object classes in PASCAL 2007 dataset. (a) aeroplane, (b) sofa, (c) bus, (d) car, (e) dining table, (f) train, (g) boat, (h) motorbike, (i) cow, (j) bicycle, (k) dog, (l) bird, (m) cat, (n) sheep, (o) monitor, (p) horse, (q) plant, (r) chair, (s) bottle, (t) person.

We compare the performance of our method with two

state-of-the-art methods, i.e., Dalal-Triggs detector [18] and deformable part-based model (DPM) [30], on the PASCAL 2007 dataset. The configuration of the Dalal-Triggs detector is similar to Section IV-A. The DPM detectors on the 20 object classes in PASCAL 2007 dataset are provided in [26]. The average precision values and ranks of the three approaches on the 20 object classes are displayed in Fig. 11 and Table III. The last column of Table III shows the mean AP and mean rank of each method.

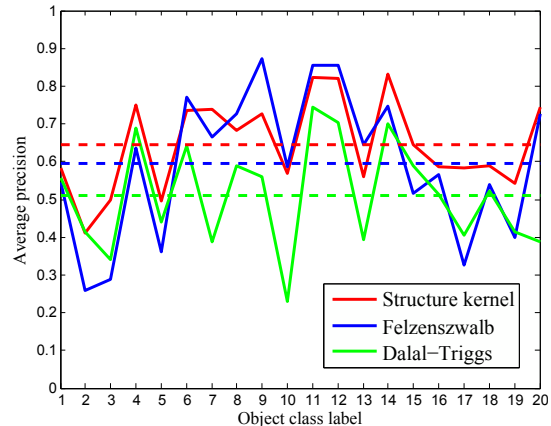


Fig. 11. Average precisions of the structure kernel, DPM and Dalal-Triggs detector on the 20 object classes in the PASCAL 2007 dataset. The dash lines indicate the mean APs of the three methods.

TABLE III
AVERAGE PRECISIONS ON THE PASCAL 2007 DATASET (AP(RANK))

	SK	DPM	Dalal-Triggs
1. person	0.584 (1)	0.546 (3)	0.558 (2)
2. bird	0.410 (2)	0.259 (3)	0.414 (1)
3. cat	0.499 (1)	0.289 (3)	0.340 (2)
4. cow	0.823 (1)	0.636 (3)	0.688 (2)
5. dog	0.495 (1)	0.361 (3)	0.440 (2)
6. horse	0.734 (2)	0.771 (1)	0.643 (3)
7. sheep	0.739 (1)	0.664 (2)	0.388 (3)
8. aeroplane	0.683 (2)	0.726 (1)	0.588 (3)
9. bicycle	0.726 (2)	0.872 (1)	0.559 (3)
10. boat	0.568 (2)	0.586 (1)	0.229 (3)
11. bus	0.823 (2)	0.855 (1)	0.745 (3)
12. car	0.819 (2)	0.854 (1)	0.703 (3)
13. motorbike	0.559 (2)	0.646 (1)	0.394 (3)
14. train	0.931 (1)	0.746 (2)	0.701 (3)
15. bottle	0.645 (1)	0.515 (3)	0.590 (2)
16. chair	0.587 (1)	0.565 (2)	0.512 (3)
17. table	0.584 (1)	0.326 (3)	0.406 (2)
18. plant	0.588 (1)	0.539 (2)	0.522 (3)
19. sofa	0.542 (1)	0.399 (3)	0.413 (2)
20. monitor	0.743 (1)	0.726 (2)	0.386 (3)
mean	0.655 (1.35)	0.594 (2.05)	0.511 (2.55)

Over the 20 object classes in the PASCAL 2007 dataset, the structure kernel obtains the highest APs in 12 object classes, the DPM obtains the highest APs in 7 object classes, and the Dalal-Triggs detector obtains the highest AP in 1 object class. From the perspective of mean AP, the structure kernel gets the highest mean AP (0.655) among the three approaches, DPM comes second (0.594), and Dalal-Triggs detector has the lowest mean AP (0.511). As the mean AP metric can possibly be affected by outliers, we further evaluate the average rank

of each method. The structure kernel has the highest mean rank (1.35) in the three approaches, DPM comes second in mean rank (2.05), and Dalal-Triggs detector has the lowest mean rank (2.55). In conclusion, the proposed structure kernel achieves the best overall result in the 20 object classes in the PASCAL 2007 dataset compared with the DPM and Dalal-Triggs detector.

1) *Performances of different base kernels*: We also evaluate the performances of different base kernels with 4 object classes in the PASCAL 2007 dataset: *bus*, *aeroplane*, *cat* and *cow*, each of which is divided into 4 parts. Three base kernels are tested, i.e., linear kernel, polynomial kernel, and RBF kernel, as defined in Section IV-A. Fig. 12 and Table IV, respectively, depict the precision-recall curves and the average precision values.

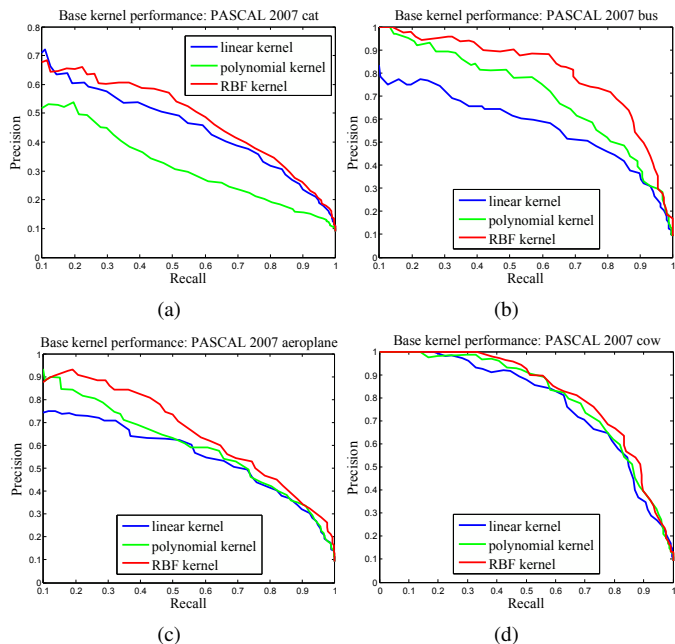


Fig. 12. Performances of different base kernels. (a) *cat*, (b) *bus*, (c) *aeroplane*, (d) *cow*.

TABLE IV
AVERAGE PRECISIONS OF DIFFERENT BASE KERNELS

	cat	bus	aeroplane	cow
linear kernel	0.472	0.599	0.579	0.782
polynomial kernel	0.343	0.733	0.622	0.803
RBF kernel	0.499	0.823	0.683	0.823

In all of the test objects classes, it can be observed that the RBF kernel achieves the best effect. Thus, the RBF kernel can be taken as the more discriminative base kernel in the structure kernel.

2) *Performance of multi-scale part representation*: We evaluate the performance of the multi-scale part representation on the *car*, *aeroplane* and *horse* object classes in the PASCAL 2007 dataset. RBF kernel is utilized as the base kernel, and each object is divided into 4 parts. The scale factor between two consecutive scales is 0.9, and the radius of the scale space L , which is defined in Section II-A, ranges from 0 to 4 to demonstrate its impact on the average precision and runtime.

If $L = 0$, parts are in the same scale as the whole object, which is a mono-scale object model. If $L > 0$, there are $2L + 1$ different scales that parts may exist, consisting of the scale of the whole object, L scales that are larger than the scale of the whole object and L scales that are smaller than the scale of the whole object.

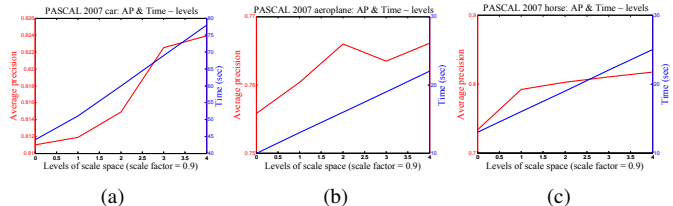


Fig. 13. $AP \sim L$ and $runtime \sim L$ curves. (a) *car*, (b) *aeroplane*, (c) *horse*.

The $AP \sim L$ curves and $runtime \sim L$ curves of the three object classes are shown in Fig. 13. The figure shows that the performance of the classifiers can be improved by introducing the multi-scale part representation. As the relative portion of a part may vary with respect to the viewpoint and intra-class variations in real scenarios, to capture the optimal parts in multiple scales is both intuitively reasonable and experimentally effective. However, to search for parts in multiple scales can take more time than to do so in just one scale. The $runtime \sim L$ curves in Fig. 13 exhibit that the relation between the runtime and the scale radius L is almost linear. Empirically, we choose $L = 2$ in the experiments for both computational efficiency and performance.

3) *Performance of weighted combination of part similarities*: We also evaluate the influence of the weighted combination of part similarities on the performance of the structured kernel with *bicycle*, *sofa*, *bird* and *diningtable* object classes in the PASCAL 2007 dataset. We choose the RBF kernel as the basic kernel, and assign 4 parts for each object class. The radius of the scale space is 2 and the scale factor is 0.9. For each object class, we train a classifier with the weighted combination of part similarities in the structure kernel and a classifier without it, i.e., all parts are endowed with the same weight $w_i = 1/n$, for $i = 1, \dots, n$. The precision-recall curves of the two kinds of classifiers on the test object classes are demonstrated in Fig. 14, and the average precision values are listed in Table V.

TABLE V
AVERAGE PRECISIONS OF WEIGHTED COMBINATION OF PARTS

	bicycle	sofa	bird	diningtable
w/o weighted combination	0.720	0.515	0.403	0.546
with weighted combination	0.726	0.542	0.411	0.584

Experimental results show that the performance of the classifier can be improved by giving different weights to parts based on their distinctiveness of the object class.

4) *Computational complexity*: The training time and testing time varies with several factors, such as the number of images in the training set, the number of parts of the object category. With 3.40GHz 8-core Intel Core i7-3770 CPU and Ubuntu 12.04 operating system, it typically takes about 4 hours to train a 4-part person classifier on the PASCAL 2007 dataset with

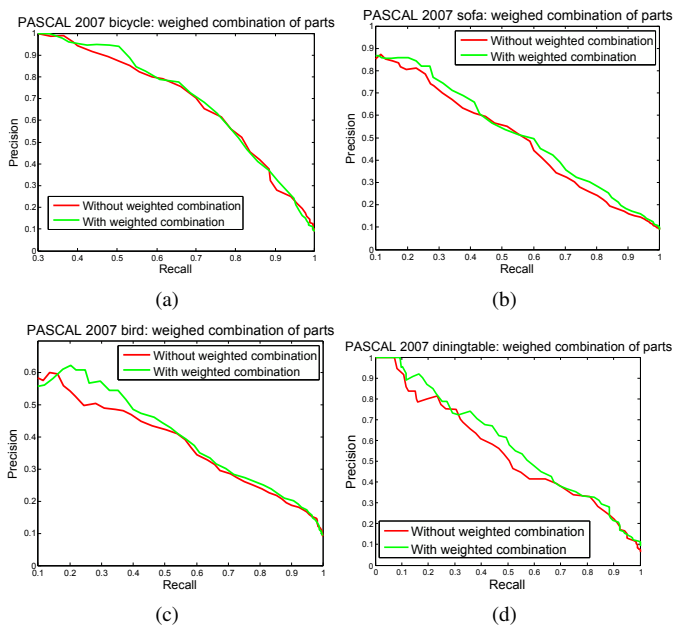


Fig. 14. Precision-recall curves of classifiers with and without weighted combination of part similarities. (1) *bicycle*, (2) *sofa*, (3) *bird*, (4) *diningtable*.

4,390 positive examples (*trainval*) and approximately 43,900 negative examples. On the other hand, it takes about 2 hours to test the classifier on the testing set consisting of 4,192 positive images and 4,1920 negative images.

V. CONCLUSIONS

In this paper, we propose a novel positive definite kernel called “structure kernel” which measures the similarity of part-based represented objects in both global appearance, parts appearance, and spatial layout of parts. It incorporates the discriminative power of kernels into flexible part-based object models, and the deformation of parts in the structure kernel is penalized in a multi-scale sense with respect to horizontal displacement, vertical displacement, and scale difference. Part similarities are combined with different weights, which are optimized efficiently to maximize the intra-class similarities and minimize the inter-class similarities by the normalized stochastic gradient ascent algorithm. The parameters of the structure kernel are learned during the training process with regard to the distribution of the data in a more flexible and discriminative way. Theoretical analysis and experimental evaluations demonstrate that the proposed multi-scale part-based representation is more robust to viewpoint variations, poses, and intra-class differences.

REFERENCES

- [1] J. Zhu, Y. Lao, and Y. Zheng, “Object Tracking in Structured Environments for Video Surveillance Applications,” *IEEE Trans. Circuits and Systems for Video Technology*, vol. 20, no. 2, pp. 223-235, Feb. 2010.
- [2] X. Jing, S. Li, D. Zhang, J. Yang, and J. Yang, “Supervised and Unsupervised Parallel Subspace Learning for Large-Scale Image Recognition,” *IEEE Trans. Circuits and Systems for Video Technology*, vol. 22, no. 10, pp. 1497-1511, Oct. 2012.
- [3] H. Cheng, C. Weng, and Y. Chen, “Vehicle Detection in Aerial Surveillance Using Dynamic Bayesian Networks,” *IEEE Trans. Image Processing*, vol. 21, no. 4, pp. 2152-2159, April 2012.

- [4] H. Sahbi, J. Audibert, and R. Keriven, “Context-Dependent Kernels for Object Classification,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 33, no. 4, pp. 699-708, April 2011.
- [5] K. Hotta, “Support Vector Machine with Local Summation Kernel for Robust Face Recognition,” in *Proc. Int’l Conf. Pattern Recognition*, Cambridge, UK, vol. 3, pp. 482-485, 2004.
- [6] K. Grauman and T. Darrell, “The Pyramid Match Kernel: Efficient Learning with Sets of Features,” *Journal of Machine Learning Research*, vol. 8, pp. 725-760, April 2007.
- [7] S. Lyu, “Mercer Kernels for Object Recognition with Local Features,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, San Diego, CA, USA, pp. 223-229, 2005.
- [8] C. Wallraven and B. Caputo, “Recognition with Local Features: the Kernel Recipe,” in *Proc. IEEE Int’l Conf. Computer Vision*, Nice, France, pp. 257-264, Oct. 2003.
- [9] Y. Y. Lin, T. L. Liu, and C. S. Fuh, “Multiple Kernel Learning for Dimensionality Reduction,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 33, no. 6, pp. 1147-1160, June 2011.
- [10] J. Yang, Y. Tian, L. Y. Duan, T. Huang, and W. Gao, “Group-Sensitive Multiple Kernel Learning for Object Recognition,” *IEEE Trans. Image Processing*, vol. 21, no. 5, pp. 2838-2852, May 2012.
- [11] C. Sun, and K. Lam, “Multiple-Kernel, Multiple-Instance Similarity Features for Efficient Visual Object Detection,” *IEEE Trans. Image Processing*, vol. 22, no. 8, pp. 3050-3061, Aug. 2013.
- [12] S. Nilufar, N. Ray, H. Zhang, “Object Detection With DoG Scale-Space: A Multiple Kernel Learning Approach,” *IEEE Trans. Image Processing*, vol. 21, no. 8, pp. 3744-3756, Aug. 2012.
- [13] Z. Wang, S. Chen, T. Sun, “MultiK-MHKS: A Novel Multiple Kernel Learning Algorithm,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 348-353, Feb. 2008.
- [14] B. McFee, C. Galleguillos, G. Lanckriet, “Contextual Object Localization With Multiple Kernel Nearest Neighbor,” *IEEE Trans. Image Processing*, vol. 20, no. 2, pp. 570-585, Feb. 2011.
- [15] S. Lazebnik, C. Schmid, and J. Ponce, “Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, New York, NY, USA, pp. 2169-2178, June 2006.
- [16] L. Wu, S. Hoi, and N. Yu, “Semantics-Preserving Bag-of-Words Models and Applications,” *IEEE Trans. Image Processing*, vol. 19, no. 7, pp. 1908-1920, July 2010.
- [17] S. Zhang, Q. Tian, G. Hua, Q. Huang, and W. Gao, “Generating Descriptive Visual Words and Visual Phrases for Large-Scale Image Applications,” *IEEE Trans. Image Processing*, vol. 20, no. 9, pp. 2664-2677, Sept., 2011.
- [18] N. Dalal and B. Triggs, “Histograms of Oriented Gradients for Human Detection,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, San Diego, CA, USA, pp. 886-893, 2005.
- [19] I. Tosic and P. Frossard, “Dictionary Learning for Stereo Image Representation,” *IEEE Trans. Image Processing*, vol. 20, no. 4, pp. 921-934, April 2011.
- [20] D. Lowe, “Distinctive Image Features from Scale-invariant Keypoints,” *Int’l Journal of Computer Vision*, vol. 60, no. 2, pp. 91-110, Nov. 2004.
- [21] T. Kadir and M. Brady, “Scale, Saliency and Image Description,” *Int’l Journal of Computer Vision*, vol. 45, no. 2, pp. 83-105, Nov. 2001.
- [22] K. Mikolajczyk and C. Schmid, “Scale and Affine Invariant Interest Point Detectors,” *Int’l Journal of Computer Vision*, vol. 60, no. 1, pp. 63-86, Oct. 2004.
- [23] A. Mohan, C. Papageorgiou, and T. Poggio, “Example-Based Object Detection in Images by Components,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 4, pp. 349-361, April 2001.
- [24] L. Zhu, Y. Chen, A. Yuille, and W. Freeman, “Latent Hierarchical Structural Learning for Object Detection,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, San Francisco, CA, USA, pp. 1062-1069, June 2010.
- [25] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results,” <http://www.pascal-network.org/challenges/VOC/voc2007/worksop/index.html>.
- [26] <http://www.cs.berkeley.edu/~rbg/latent/index.html>
- [27] D. Park, D. Ramanan, and C. Fowlkes, “Multiresolution Models for Object Detection,” in *Proc. European Conf. Computer Vision*, San Francisco, CA, USA, pp. 241-254, June 2010.
- [28] H. Schneiderman and T. Kanade, “A Statistical Method for 3D Object Detection Applied to Faces and Cars,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Hilton Head Island, SC, USA, vol. 1, pp. 746-751, 2000.

- [29] E. Bernstein and Y. Amit, "Part-based Statistical Models for Object Classification and Detection," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, San Diego, CA, USA, vol. 2, pp. 734-740, 2005.
- [30] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object Detection with Discriminatively Trained Part Based Models," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627-1645, Sept. 2010.
- [31] H. Li, Y. Wei, L. Li and C.L. Philip Chen, "Hierarchical Feature Extraction With Local Neural Response for Image Recognition," *IEEE Trans. Cybernetics*, vol. 43, no. 2, pp. 412-424, Apr. 2013.
- [32] C. M. Pun and M. C. Lee, "Log-polar Wavelet Energy Signatures for Rotation and Scale Invariant Texture Classification," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 590-603, May 2003.
- [33] Z. Arican and P. Frossard, "Scale-Invariant Features and Polar Descriptors in Omnidirectional Imaging," *IEEE Trans. Image Processing*, vol. 21, no. 5, pp. 2412-2423, May 2012.
- [34] A. Y. Wibisono, J. Bouvrie, L. Rosasco, and T. Poggio, "Learning and Invariance in a Family of Hierarchical Kernels," MIT-CSAIL-TR-2010-035/CBCL-290, MIT 2010.
- [35] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504-507, July 2006.
- [36] G. Huang, H. Lee, and E. Miller, "Learning Hierarchical Representations for Face Verification with Convolutional Deep Belief Networks," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Providence, RI, USA, pp. 2518-2525, 2012.
- [37] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems*, pp. 1106-1114, 2012.
- [38] R. Salakhutdinov, J. Tenenbaum, and A. Torralba, "Learning with Hierarchical-Deep Models," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1958-1971, Aug. 2013.
- [39] G. Montavon, M. L. Braun, K.-R. Muller, "Kernel Analysis of Deep Networks," *Journal of Machine Learning Research*, vol. 12, pp. 2563-2581, 2011.

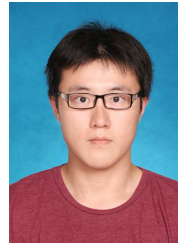


Hongkai Xiong (M'01-SM'10) received the Ph.D. degree in communication and information system from Shanghai Jiao Tong University (SJTU) in 2003. Since then, he has been with the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai, where he is currently a Professor. From Dec. 2007 to Dec. 2008, he was with the Department of Electrical and Computer Engineering, Carnegie Mellon University (CMU), PA, as Research Scholar. From 2011 to 2012, he was a scientist with the Division of Biomedical Informatics at the University

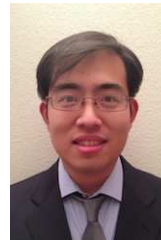
of California, San Diego (UCSD).

Dr. Xiong's research interests include source coding/network information theory, signal processing, computer vision and graphics, and statistical machine learning. He has published over 100 refereed journal/conference papers. In SJTU, he directs "Image, Video, and Multimedia Communications (IVM)" Lab and "Multimedia Communication" area in the Key Lab of Ministry of Education of China - "Intelligent Computing and Intelligent System" which is also co-granted by Microsoft Research.

He is the recipient of the Top 10% Paper Award for "Super-Resolution Reconstruction with Prior Manifold on Primitive Patches for Video Compression" at the 2011 IEEE International Workshop on Multimedia Signal Processing (IEEE MMSP'11). In 2011, he obtained the First Prize of the Shanghai Technological Innovation Award. In 2010, he obtained the SMC-A Excellent Young Faculty Award of Shanghai Jiao Tong University. In 2009, he was awarded a recipient of New Century Excellent Talents in University, Ministry of Education of China. He is a senior member of the IEEE, and serves as a technical program committee member or session chair for a number of international conferences.



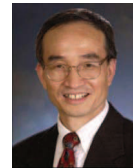
Botao Wang received the B.S. degree in electronic engineering, in 2010, from Shanghai Jiao Tong University, Shanghai, China, where he is currently working toward the Ph.D. degree. His main research interests include object detection, visual tracking, and stereo vision.



Xiaoqian Jiang, Ph.D., is an Assistant Professor in the Division of Biomedical Informatics, School of Medicine, University of California San Diego. His expertise is in data privacy and machine learning. He has done research on imbalanced data analysis, predictive model calibration, and privacy-preserving data mining.

He received a distinguished paper award from AMIA Summits on Transnational Science in 2012 and served as the tutorial chair for the 2nd IEEE conference of Health Informatics, Imaging, and System Biology (HISB). His current research focuses on developing practical and scalable technologies for large data analysis.

tem Biology (HISB). His current research focuses on developing practical and scalable technologies for large data analysis.



Yuan F. Zheng (F97) received the MS and Ph.D. degrees in Electrical Engineering from The Ohio State University, in Columbus, Ohio in 1980 and 1984, respectively. His undergraduate education was received at Tsinghua University, Beijing, China in 1970. From 1984 to 1989, he was with the Department of Electrical and Computer Engineering at Clemson University, Clemson, South Carolina. Since August 1989, he has been with The Ohio State University, where he is currently Professor and was the Chairman of the Department of Electrical and Computer Engineering from 1993 to 2004. From 2004 to 2005, Professor Zheng spent sabbatical year at the Shanghai Jiao Tong University in Shanghai, China and continued to be involved as Dean of School of Electronic, Information and Electrical Engineering until 2008. Professor Zheng is an IEEE Fellow.

Professor Zheng's research interests include two aspects. One is in wavelet transform for image and video, and object classification and tracking, and the other is in robotics which includes robotics for life science applications, multiple robots coordination, legged walking robots, and service robots. Professor Zheng was and is on the editorial board of five international journals. Professor Zheng received the Presidential Young Investigator Award from Ronald Reagan in 1986, and the Research Awards from the College of Engineering of The Ohio State University in 1993, 1997, and 2007, respectively. Professor Zheng along with his students received the best conference and best student paper award a few times in 2000, 2002, and 2006, and received the Fred Diamond for Best Technical Paper Award from the Air Force Research Laboratory in Rome, New York in 2006. In 2004, Professor Zheng was appointed to the International Robotics Assessment Panel by the NSF, NASA, and NIH to assess the robotics technologies worldwide in 2004 and 2005.