Group-Sensitive Multiple Kernel Learning for Object Recognition

Jingjing Yang, *Student Member, IEEE*, Yonghong Tian, *Senior Member, IEEE*, Ling-Yu Duan, *Member, IEEE*, Tiejun Huang, *Member, IEEE*, and Wen Gao, *Fellow, IEEE*

***Abstract*—In this paper, a group-sensitive multiple kernel learning (GS-MKL) method is proposed for object recognition to accommodate the intraclass diversity and the interclass correlation. By introducing the "group" between the object category and individual images as an intermediate representation, GS-MKL attempts to learn group-sensitive multikernel combinations together with the associated classifier. For each object category, the image corpus from the same category is partitioned into groups. Images with similar appearance are partitioned into the same group, which corresponds to the subcategory of the object category. Accordingly, intraclass diversity can be represented by the set of groups from the same category but with diverse appearances; interclass correlation can be represented by the correlation between groups from different categories. GS-MKL provides a tractable solution to adapt multikernel combination to local data distribution and to seek a tradeoff between capturing the diversity and keeping the invariance for each object category. Different from the simple hybrid grouping strategy that solves sample grouping and GS-MKL training independently, two sample grouping strategies are proposed to integrate sample grouping and GS-MKL training. The first one is a looping hybrid grouping method, where a global kernel clustering method and GS-MKL interact with each other by sharing group-sensitive multikernel combination. The second one is a dynamic divisive grouping method, where a hierarchical kernel-based grouping process interacts with GS-MKL. Experimental results show that performance of GS-MKL does not significantly vary with different grouping strategies, but the looping hybrid grouping method produces slightly better results. On four challenging data sets, our proposed method has achieved encouraging performance comparable to the state-of-the-art and outperformed several existing MKL methods.**

***Index Terms*—Dynamic divisive grouping (DDG), interclass correlation, intraclass diversity, looping hybrid grouping, multiple kernel learning (MKL), object recognition.**

Manuscript received October 10, 2010; revised June 30, 2011 and December 05, 2011; accepted December 09, 2011. Date of publication January 09, 2012; date of current version April 18, 2012. This work was supported in part by grants from the Chinese National Natural Science Foundation under Contract 61035001 and Contract 60973055, and in part by the National Basic Research Program of China under Contract 2009CB320906. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Kenneth K. M. Lam.

J. Yang is with the National Engineering Laboratory for Video Technology, Peking University, Beijing 100871, China, with the Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China, and also with Guangdong Province Transmission and Substation Engineering Company, China.

Y. Tian, L.-Y. Duan, T. Huang, and W. Gao are with the National Engineering Laboratory for Video Technology, and also with Key Laboratory of Machine Perception (MoE), School of Electronic Engineering and Computer Science, Peking University, Beijing 100871, China (e-mail: yhtian@pku.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2012.2183139

I. Introduction

OBJECT RECOGNITION is an important yet challenging task in image processing and computer vision. Research from different aspects have been done to push the research of object recognition forward. On one hand, many efforts have been devoted to design robust features [24], [32], [33], [36] and kernels [3], [15], [16], [33]. On the other hand, numerous heuristic learning methods [5], [6], [13], [14], [19], [20], [37] have been proposed to boost the performance of object recognition. In addition, some institutes and organizations have been actively constructed open benchmark data sets (e.g., Caltech101 [6], PASCAL VOC [7], WikipediaMM [9], and ImageNet [56]) for fair evaluation of distinct features and learning methods. Although much progress has been made, most state-of-the-art methods are still insufficient to tackle object recognition in general and a practical data set of large size.

One of the essential difficulties of object recognition lies in that the images within an object category usually exhibit diversity, whereas the ones from different categories would have correlations in visual appearance. For images from one object category, many factors (e.g., variation of object pose, viewpoints, and sources of data collection) may incur diverse feature distributions and statistical properties. For instance, Fig. 1 illustrates some typical images from WikipediaMM data set [9]. Given an object category "bridges," positive images can be grouped into three subcategories, each of which demonstrates distinct visual appearance. Meanwhile, images from two different object categories (e.g., "bridges" and "buildings") may share similar feature distributions and visual attributes. In this paper, such phenomena are referred to as "intraclass diversity" and "interclass correlation," respectively. To elegantly recognize objects over extensive and practical image data sets, we argue that, it is meaningful to effectively model both intraclass diversity and interclass correlation.

To overcome the issues of intraclass diversity and interclass correlation, a lot of works have been done on designing features [24], [32], [33], [36], [47], [50], which are highly invariant to the intraclass variance and robust to classify a correlated object. Despite some improvements, not all features are the same discriminative for all object categories. Therefore, instead of using one single feature, fusing a set of diverse and complementary features is widely approved [5], [10], [19], [22], [37], [38]. In particular, multiple kernel learning (MKL) methods [10], [22], [30] have shown great advantages in this task recently (e.g., [5] and [18]). Instead of using a single kernel in a support vector machine (SVM) [21], MKL learns an optimal kernel combination and the associated classifier simultaneously, providing an

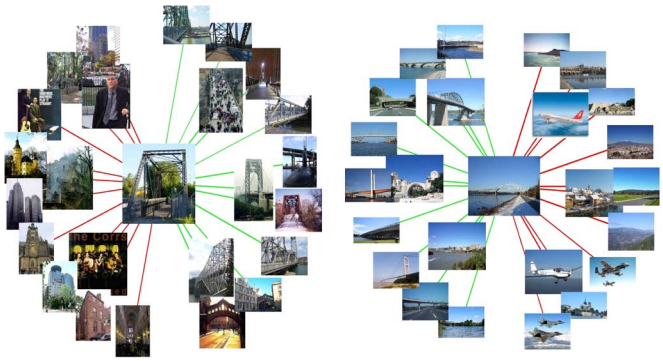


Fig. 1. Illustration of intraclass diversity and interclass correlation of "bridges." The green line connects two images from the same category, whereas the red line connects images from different categories.

effective way of fusing informative features and kernels. However, these methods basically adopt a globally uniform similarity measure over the whole input space. When an object category possesses high intraclass variation as well as correlation with other categories in appearance, the importance of different features/ kernels may vary among samples. Hence, performance of these global MKL methods can be degraded due to the complexity of data distribution.

Recently, there have been attempts [11], [19], [20], [29] to learn samplewise kernel combination/ distance function to accommodate the intraclass diversity. For example, a sample-specific ensemble kernel learning method is proposed in [29] to explore the relative contributions of distinct kernels for each sample. Frome *et al.* [11] proposed to learn per-sample distance by solving a convex optimization problem. In practice, such methods have yielded promising performance. However, expensive computation is incurred to learn sample-based similarity measures. More importantly, several dominant samples may overwhelm the intrinsic properties of an object category so as to make the classifier less reliable.

In this paper, "*group*" is introduced between object categories and individual images as an intermediate representation to seek a tractable solution for the tradeoff between capturing the diversity and keeping the invariance. Given an object category, the training corpus from the same category is partitioned into groups. Samples within the same group belong to the same object category and share similar appearance, corresponding to the *subclasses* of the object category. Intraclass diversity can be represented by the set of groups of the same category with diverse appearance. Interclass correlation can be represented by the correlation between groups from different categories.

Accordingly, by incorporating "*groups*" into the MKL framework, and a group-sensitive MKL (GS-MKL) method is proposed for object recognition to adapt kernel combination to the local data distributions for subcategories. In GS-MKL, the image-to-image similarity is represented by a weighted combination of multiple kernels, where the kernel weights depend not only on the corresponding kernel functions but also on the groups that the two comparing images belong to. Instead of a uniform or sample-specific similarity measure, GS-MKL learns group-sensitive multikernel combinations together with the associated GS-MKL classifier, which has been shown effective in dealing with both intraclass diversity and interclass correlation.

Since there is no prior knowledge available about the subclasses of an object category, a clustering method is employed for GS-MKL to partition training samples from the same category into groups. Rather than solving sample grouping and GS-MKL training independently, two grouping strategies, i.e., the looping hybrid grouping and the dynamic divisive grouping (DDG) methods are proposed to integrate sample grouping and GS-MKL training.

In the looping hybrid grouping method, GK-KMeans is combined with GS-MKL directly. Given a group number, unweighted multikernel combination (UMK) is first employed by global kernel k-means (GK-KMeans) to initially partition training samples from the same category into groups. The GS-MKL classifier is optimized over the initially grouped training samples together with group-sensitive multikernel combinations learning. Then, sample grouping interacts with GS-MKL by treating the newly learnt group-sensitive multikernel combination as an updated kernel metric in GK-KMeans to refine the grouping results. Such looping process iterates until the object function reaches a local maximum. The best sample grouping results are obtained by enumerating all candidate group numbers and choosing the one with the best recognition performance over the validation set. In DDG, a hierarchy of training samples from the same category is built for each object category to group training samples over different degrees. The hierarchical kernel-based grouping process interacts with GS-MKL by sharing the group-sensitive multikernel combinations. Compared with the looping hybrid grouping method, DDG provides a unified learning strategy where finding the optimal group number and sample grouping are integrated together with the GS-MKL training.

To evaluate the effectiveness of the proposed methods, experiments are carried out over four data sets (i.e., Caltech101, Pascal VOC2007, WikipediaMM, and Scene15). GS-MKL has gained superior performance against several existing MKL methods [18], [38], [39] and shown effectiveness to alleviate the negative influence of intraclass diversity and interclass correlation, coming up with a robust discriminative power for object recognition.

Our main contributions can be summarized as follows.

- A GS-MKL method is proposed for object recognition, where both intraclass diversity and interclass correlation are taken into account. GS-MKL provides a general and tractable solution to adapt MKL to local data distribution. When the number of group declines to one, GS-MKL is reduced to canonical MKL. When the number of group reaches up to the number of training images, GS-MKL becomes a sample-specific MKL (SS-MKL).
- Two grouping strategies are proposed and evaluated for GS-MKL. Experimental results show that the performance of GS-MKL does not significantly vary with different grouping strategies. A simple hybrid grouping strategy can boost GS-MKL against other multiple kernel methods. Furthermore, the looping hybrid grouping method, where GK-KMeans is integrated with GS-MKL and the sample grouping results are iteratively refined, provides slightly better results than the other grouping strategies.

- Promising experimental results comparable to the state-of-the-art results have been obtained on Caltech101, Pascal VOC2007, and Scene15 data sets, and significant improvements have been achieved over several existing MKL methods across the four data sets. A new bound is established for the performance of the state-of-the art MKL method on object recognition.

The remainder of this paper is organized as follows. Section II briefs the related work. In Section III, the GS-MKL framework is introduced for object recognition. The learning algorithm of GS-MKL is presented in Section IV. Section V presents two sample grouping strategies for GS-MKL. The experimental results are given in Section VI. Finally, Section VII concludes this paper.

A preliminary version of this work has been published in [53]. The main extensions include two grouping strategies, where sample grouping interacts with GS-MKL training, grouping strategy comparison, comparisons of GS-MKL and other MKL methods, and more extensive experiments.

II. Related Work

In the past decade, research efforts have been devoted to characterizing visual statistics for a number of object categories [2], [7], [13], [14], [19], [27]. Among them, the kernel method [3], [5], [15], [16], [18] is one of the attractive research areas. Generally speaking, the kernel method offers two advantages in learning object categories: (1) A kernel explicitly defines a visual similarity measure between image pairs and implicitly maps the input space to the feature space [13], thereby avoiding the explicit feature representation and the curse of dimension; (2) Combined with SVM, the kernel method can find out the optimal separating hyper-plane between positive and negative samples efficiently. Hence, the SVM-based kernel method has been applied to many recognition problems (e.g., object detection [40] and image and video annotation [41]–[43]), in addition to object recognition. Generally, SVM-based kernel methods used in object recognition can be categorized into four types, i.e., individual kernel designing, canonical MKL, SS-MKL, and SVM ensemble. We brief the related works as follows.

A. Individual Kernel Designing

Recently, many efforts have been made to delicately design individual kernels for the similarity of an image pair. A kernel based on a multiresolution histogram is introduced in [15] to measure the image similarity at different granularities. A spatial pyramid matching kernel (PMK) is introduced in [3] to enforce the loose spatial information, which matches images with spatial coordinates. A kernel based on the local feature distribution is presented in [16] to model the image local context. A chi-squared kernel based on the pyramid histogram of orientated gradients (PHOG) is presented in [33] to capture the shape similarity with spatial layout.

All these methods rely on the features that represent particular visual characteristics. However, not all kernels play the same role in differentiating object categories. Hence, kernel selection/fusion over a set of available kernels is usually desired for generic object recognition. It is worthy to note that individual kernels can be incorporated into the proposed GS-MKL framework to investigate the corresponding contributions in object recognition.

B. Canonical MKL

Recently, instead of using a single kernel, a classifier based on multikernel combination has been introduced into object recognition, yielding promising results [5], [18], [38], [45]. In [5] and [18], multiple features (e.g., appearance and shape) and kernels [e.g., PMK and spatial pyramid kernels (SPKs) with different hyper-parameters] are employed and combined in the MKL framework. Bosch *et al.* [45] strengthens MKL with a cross validation strategy. The initial weights of multiple kernels are learnt by an extended MKL [5] and then refined by an exhaustive search to minimize the classification error over a validation set. In [44], kernel alignment is utilized to optimize multikernel combination over color, shape, and appearance features.

Basically, these methods adopt a uniform multikernel combination over the whole input space. Hence, when training data exhibit high intraclass variation and interclass correlation on local training samples, these methods may suffer a degraded performance due to the choice of global uniform multikernel combination.

C. SS-MKL

More recently, SS-MKL methods have been proposed in [23], [27], and [29] by using sample-specific kernel weighting strategies. The basic idea is that kernel weights depend not only on the kernel functions but also on the samples themselves. Compared with canonical MKL, SS-MKL tends to reflect the relative importance of different kernels at the level of individual sample rather than at the level of object category. Despite some performance improvements, learning too many parameters may lead to the expensive computation cost and the risk of overfitting.

It has to be noted that, although the proposed GS-MKL and the methods [5], [18], [23], [27], [45] reviewed above are all extended from the MKL framework, GS-MKL provides a mechanism of evaluating multiple kernels over sample groups. From this view, GS-MKL is a more flexible framework that can be generalized to canonical MKL and SS-MKL by changing the number of groups. GS-MKL provides a tractable solution to adapt multikernel combination to the local data distributions for sample groups.

D. Learning With Classifier Ensemble

Instead of a single classifier, classifier ensemble has been proposed as an alternative technique to improve classification accuracy. Classifier ensemble can take place at data, feature, and classifier levels [46]. To cope with the diversity of data, a straightforward classifier ensemble method employs a data partitioning strategy where each base classifier is trained over a distinct subset of the training data. Such divide and conquer methods train multiple base classifiers that are experts in their specific parts of the data space. However, base classifiers are independently trained, leaving out the other partitions of the data. When such independence condition is not satisfied, it cannot be assured that the decision of the base classifier will improve the final classification performance.

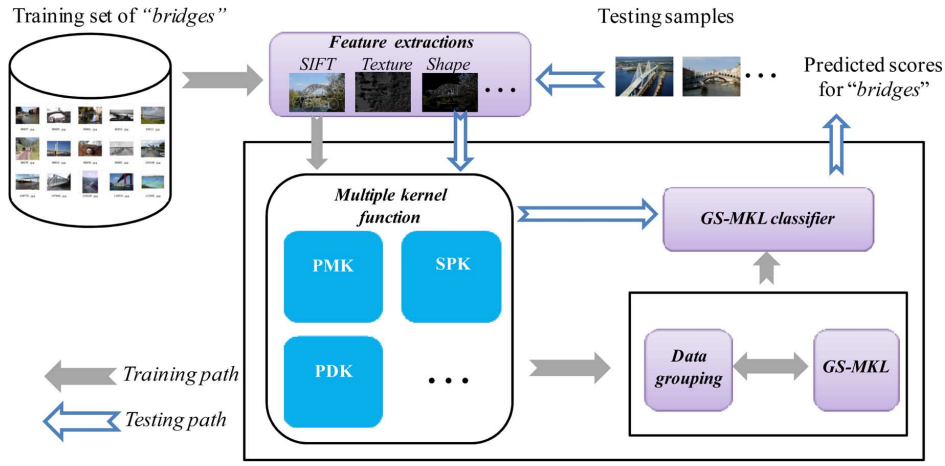


Fig. 2. Diagram of GS-MKL for object recognition.

Although both the classifier ensemble method and GS-MKL partition training data into disjoint sets, the underlying assumption and resulting model is different. Classifier ensemble methods partition the whole training data set into sets and assume each set to be independent of each other. Multiple classifiers are trained over the disjoint sets independently and further fused in the postprocessing. In GS-MKL, the training corpus from the same category is partitioned into groups. Accordingly, group-sensitive multikernel combinations are learnt together with the GS-MKL classifier to adapt multikernel combination to the local data distributions of sample groups.

III. Framework Overview

The main objective of this work is to identify the presence/absence of the predefined object categories within the image with one-versus-all setting. Let $C = (c_1, \ldots, c_l, \ldots, c_L)$ represent the lexicon of the predefined object categories. Let $D_l = \{x_i, y_i\}_{i=1}^{N_l}$ denote a training image corpus, where x_i denotes the ith sample, $y_i = \{\pm 1\}$ stands for the binary label, and N_l is the number of training samples for a given object category c_l. Based on the labeled data set $\{D_l\}_{l=1}^{L}$, we aim to train a classifier $f(x)$ for each object category based on the following multikernel combination:

$$K(x_i, x) = \sum_{m=1}^{M} \beta_m(x_i, x) K_m(x_i, x) \tag{1}$$

where $K_m(x_i, x)$ stands for the basic kernel, M is the total number of basic kernels, and $\beta_m(x_i, x)$ is the corresponding kernel weight that adapts to the local data distribution.

Fig. 2 illustrates the diagram of the proposed GS-MKL with the object category of "bridge." At the training phase, kernel similarities of an image pair in different low-level features (e.g., color, texture, and shape) are computed via multiple kernel functions [e.g., PMK, SPK, and proximity distribution kernel (PDK)]. These kernel matrices for the training corpus are saved and fed to the following processes. The training corpus from the same category is first partitioned into groups by a sample grouping process (i.e., looping hybrid grouping method or DDG). Based on the grouped training samples and their corresponding group assignments, the GS-MKL classifier is learnt together with the group-sensitive multikernel combinations. Then, the newly learnt group-sensitive multikernel combinations are shared to the following sample grouping process and serve as the updated kernel metric. Iteratively, sample grouping and GS-MKL training interact with each other to obtain a local optimal training corpus partition and its corresponding GS-MKL classifier. During the testing phase, the score of an unseen image is predicted by the learnt GS-MKL classifier with multikernel similarity between the testing sample and the support vectors of the GS-MKL classifier.

IV. GS-MKL

In this section, a brief introduction of existing multikernel combination and group-sensitive multikernel combination are presented in Section IV-A. Learning of GS-MKL is subsequently presented in Section IV-B.

A. Multikernel Combination

Kernel-based SVMs have been proven to be efficient tools for solving classification problems. The main idea is to map samples from the input space to a feature space where they are linearly separable. For binary classification, the decision function of a kernel-based SVM is defined as

$$f(x) = \sum_{i=1}^{N} \alpha_i y_i K(x_i, x) + b \tag{2}$$

where N is the number of training samples, and $\{\alpha_i\}_{i=1}^{N}$ and b are the coefficients of the classifier, corresponding to the Lagrange multipliers and the bias of the kernel-based SVM problem. $K(x_i, x) = \langle \phi(x_i), \phi(x) \rangle$ is the kernel function corresponding to the inner product of samples in the feature space ϕ. How to find the appropriate kernel function is an important step of SVM training. Rather than enumerating all candidate kernels by cross validation over the validation data set, researchers [5], [18], [34] advocate to fuse multiple kernels and learn the corresponding classifier together over the same training set. In the following part, several multikernel combinations widely used in object recognition are introduced together with the proposed group-sensitive multikernel combination.

1) Global Constant Multikernel Combination: In recent studies [5], [10], [18], [22], [34], it has been reported that using multikernel combination instead of a single kernel can help improve the classification performance. One straightforward

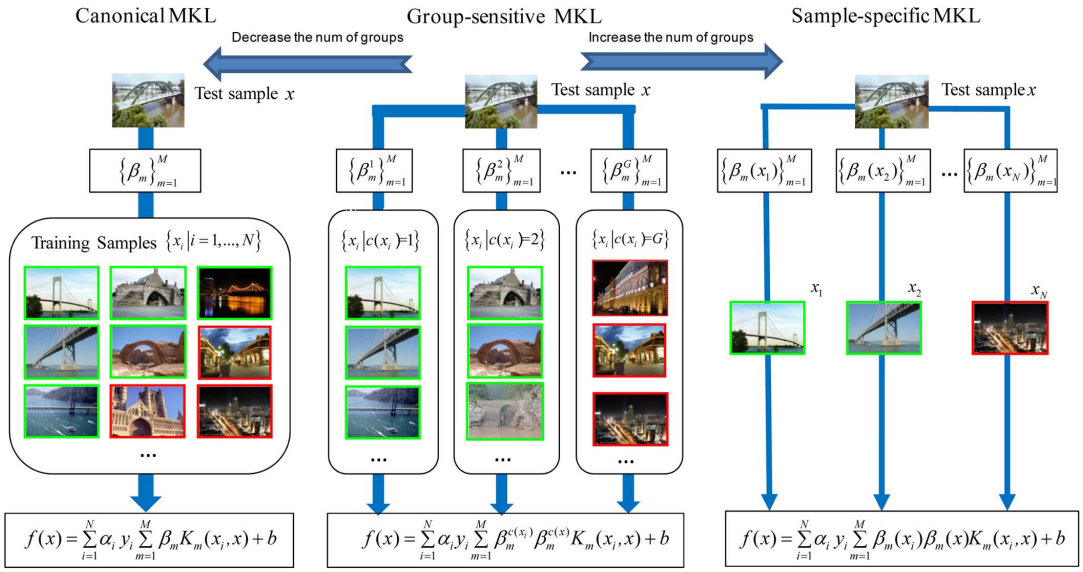


Fig. 3. Three paradigms of object recognition using (a) Canonical MKL, (b) GS-MKL, and (c) SS-MKL. In the figure, images with green bounding boxes are positive samples, whereas those with red bounding boxes are negative samples for "bridge." Note that SS-MKL will learn two sets of kernel weights even for two images with quite similar appearance (e.g., x_1 and x_2).

strategy is to assume that the kernel combination is constant throughout the data space. The simplest choice is to use an unweighted sum of multiple kernels (UMK) in (3) as a substitution of (1), i.e.,

$$K(x_i,x)=\sum_{m=1}^{M}\frac{1}{M}K_m(x_i,x). \tag{3}$$

In UMK, all basic kernels are treated equally. However, it is usually the case that different kernels may have different contributions in the recognition.

Alternatively, canonical MKL [10] [see Fig. 3(a)] is proposed to employ a convex kernel combination. Thus

$$K(x_i,x)=\sum_{m=1}^{M}\beta_m K_m(x_i,x) \tag{4}$$

with $\sum_{m=1}^{M}\beta_m=1$ and $\beta_m\geq 0\,\forall m$. $\{\beta_m\}_{m=1}^{M}$ are the kernel weights of different kernels to evaluate the corresponding contributions in the recognition. Accordingly, the decision function of canonical MKL is given as

$$f(x)=\sum_{i=1}^{N}\alpha_i y_i\sum_{m=1}^{M}\beta_m K_m(x_i,x)+b. \tag{5}$$

The coefficients α_i and the kernel weights β_m in (5) can be obtained by solving in a joint optimization problem (details can be found in [30])

$$\max_{\boldsymbol{\beta}}\ \min_{\boldsymbol{\alpha}} J, \text{ where}$$
$$J=\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j\left(\sum_{m=1}^{M}\beta_m K_m(x_i,x_j)\right)-\sum_{i=1}^{N}\alpha_i,$$
$$\text{s.t.}\sum_{i=1}^{N}\alpha_i y_i=0,\quad 0\leq\alpha_i\leq C\,\forall i. \tag{6}$$

One can also employ kernel alignment to optimize β_m and then solve the kernel-based SVM with the learnt multikernel combination. The kernel alignment technique optimizes problem

$$\max_{\boldsymbol{\beta}}\quad\frac{\langle\mathbf{K},\mathbf{G}\rangle_F}{\sqrt{\langle\mathbf{K},\mathbf{K}\rangle_F\langle\mathbf{G},\mathbf{G}\rangle_F}}$$
$$\text{s.t.}\quad\mathbf{K}=\sum_{m=1}^{M}\beta_m\mathbf{K}_m,\ \text{trace}(\mathbf{K})=1,\beta_m\geq 0. \tag{7}$$

In (7), $\langle\mathbf{K},\mathbf{G}\rangle_F=\sum_{i,j}^{N}K(x_i,x_j)G(x_i,x_j)$, where $\mathbf{G}$ is the target matrix for $\mathbf{K}$ to align and is defined as $\mathbf{G}=yy^T$. Details about the solving process can be found in the literature [27], [44].

2) Sample-Specific Multikernel Combination: It can be also assumed that multikernel combination is dependent on training samples to compare. In [27], sample-specific multikernel combination is obtained by solving a maximal problem with the local target matrix for each training sample. During test, the nearest neighbor of the test sample is located among training corpus, and the SVM classifier with the corresponding sample-specific multikernel combination is utilized during the classification. However, such local method classifies a test sample according to the multikernel combination of the nearest neighbor and hence have a risk of overfitting caused by the noisy data.

As shown in Fig. 3(c), another strategy (i.e., SS-MKL) is to assume that kernel combination depends on the sample pair to compare

$$K(x_i,x)=\sum_{m=1}^{M}\beta_m(x_i,x)K_m(x_i,x)$$
$$=\sum_{m=1}^{M}\beta_m(x_i)\beta_m(x)K_m(x_i,x). \tag{8}$$

Then, the decision function in (5) can be rewritten as

$$f(x) = \sum_{i=1}^{N} \alpha_i y_i \sum_{m=1}^{M} \beta_m(x_i)\beta_m(x) K_m(x_i, x) + b \quad (9)$$

where $\beta_m(x_i)$ are the samplewise kernel weights, and coefficients $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_N]^T$ and bias b are similarly defined as in canonical MKL. In [54], samplewise kernel weights are learnt by semi-infinite linear program wrapping canonical SVM solver. In [39], a gating model is employed to model the sample-specific multikernel combination. However, these methods are infeasible to the training classifier over a large number of training samples.

3) Group-Sensitive Multikernel Combination: Instead of learning a global constant multikernel combination or sample-specific multikernel combination, we argue that multikernel combination should adapt to the data distribution of groups. As shown in Fig. 3(b), the training corpus from the same category is partitioned into groups (see Section V for details). Let $c(x)$ and $c(x_i)$ be the group IDs of image x and x_i, respectively. The combined kernel form in (1) can be rewritten as

$$K(x_i, x) = \sum_{m=1}^{M} \beta_m^{c(x_i)} \beta_m^{c(x)} K_m(x_i, x) \quad (10)$$

where $\beta_m^{c(x)}$ and $\beta_m^{c(x_i)}$ are *group-sensitive kernel weights* of x and x_i. Let G denote the total group number, then $\beta_m^{c(x)} \in \{\beta_m^1, \ldots, \beta_m^g, \ldots, \beta_m^G\}$ for $m \in (1, \ldots, M)$. Accordingly, the decision function in (5) can be reformulated as

$$f(x) = \sum_{i=1}^{N} \alpha_i y_i \sum_{m=1}^{M} \beta_m^{c(x_i)} \beta_m^{c(x)} K_m(x_i, x) + b \quad (11)$$

where the coefficients $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_N]^T$ and bias b are the same as those in canonical MKL. This decision function can be derived from the GS-MKL primal problem discussed in Section IV-C. Compared with M kernel weights in the canonical MKL case, the number of group-sensitive kernel weights rises up to $G \times M$. The coefficients of the classifier and the group-sensitive kernel weights can be optimized in a joint manner, which will be discussed in Section IV-D.

GS-MKL can be generalized to canonical MKL and SS-MKL by changing the group number. In the special case of $G = 1$, all samples belong to the unique group and share a set of kernel weights $\{\beta_m^1\}_{m=1}^M$. In this case, GS-MKL is simplified to canonical MKL [see Fig. 3(a)], where β_m^1 in (10) is equal to the square root of β_m in (4).

For the case of $G = N$, each individual group is composed of only one training sample, and thus, a sample-specific kernel weighting strategy is employed. This way, $\beta_m^{c(x)}$ only depends on the kernel function and the sample x. It has to be noted that $\beta_m^{c(x)}$ is equivalent to $\beta_m(x)$ in localized MKL [23]. The number of group-sensitive kernel weights increases to $N \times M$, where $N \gg G$. In this case, GS-MKL scales up to SS-MKL [see Fig. 3(c)].

B. Learning GS-MKL-Based Classifier

1) GS-MKL Primal Problem: In GS-MKL, sample x is transformed via mappings $\{\phi_m(x) \mapsto \mathbb{R}^{d_m}\}_{m=1}^M$ from the input space into M feature spaces $(\phi_1(x), \ldots, \phi_M(x))$, where d_m denotes the dimensionality of the mth feature space. Each feature map is associated with a weight vector $\mathbf{w}_m$. To allow the multikernel combination in (10), the decision function of canonical MKL in (5) can be rewritten as follows:

$$f(x) = \sum_{m=1}^{M} \beta_m^{c(x)} \langle \mathbf{w}_m, \phi_m(x) \rangle + b. \quad (12)$$

Inspired by the SVM [21], the training procedure can be implemented by solving the following optimization problem:

$$\begin{aligned} \min_{\mathbf{w}_m, b, \xi, \beta} \quad & \frac{1}{2} \sum_{m=1}^{M} \|\mathbf{w}_m\|^2 + C \sum_{i=1}^{N} \xi_i, \\ \text{s.t.} \quad & y_i \left(\sum_{m=1}^{M} \beta_m^{c(x_i)} \langle \mathbf{w}_m, \phi_m(x_i) \rangle + b \right) \geq 1 - \xi_i \, \forall i, \\ & \xi_i \geq 0 \, \forall i. \end{aligned} \quad (13)$$

In (13), $\|\mathbf{w}_m\|^2$ is a regularization term that is inversely related to the margin, ξ_i is a slack variable for each training sample to allow soft margin violation, $\sum_{i=1}^N \xi_i$ measures the total classification error, and C is the misclassification penalty. The optimal C can be obtained by cross validation. The object function in (13) maximizes the margin between positive and negative samples and minimizes the empirical classification error.

2) GS-MKL Dual Problem: Through introducing Lagrange multipliers $\{\alpha_i\}_{i=1}^N$ into the inequalities constraint in (13) and formulating the Lagrangian dual function, which satisfies the Karush–Kuhn–Tucker condition [10], the former optimization problem reduces to a max-min problem as follows:

$$\begin{aligned} & \max_{\boldsymbol{\beta}} \min_{\boldsymbol{\alpha}} J, \text{ where} \\ & J = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \\ & \qquad \times \left(\sum_{m=1}^{M} \beta_m^{c(x_i)} \beta_m^{c(x_j)} K_m(x_i, x_j) \right) - \sum_{i=1}^{N} \alpha_i, \\ & \qquad \text{s.t.} \sum_{i=1}^{N} \alpha_i y_i = 0, \; 0 \leq \alpha_i \leq C \, \forall i. \end{aligned} \quad (14)$$

This max-min problem is called the GS-MKL dual problem. J is a multiobject function of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. When $\boldsymbol{\beta}$ is fixed, minimizing J over the coefficient $\boldsymbol{\alpha}$ is equivalent to minimizing the global classification error and maximizing the margin between positive and negative classes. When $\boldsymbol{\alpha}$ is fixed, maximizing J over the group-sensitive kernel weights $\boldsymbol{\beta}$ is to maximize the intraclass similarity and minimize the interclass similarity simultaneously.

3) Optimization Algorithm: Similar to the parameter learning in canonical MKL, a two-stage alternant optimization approach is adopted.

a) Computation of $\boldsymbol{\alpha}$ given $\boldsymbol{\beta}$: Fixing $\boldsymbol{\beta}$, the classifier coefficient $\boldsymbol{\alpha}$ can be estimated by minimizing J under the constraint $0 \leq \alpha_i \leq C\ \forall i$ and $\sum_{i=1}^{N} \alpha_i y_i = 0$. Minimization of J is identical to solving the canonical SVM dual problem with the multikernel combination in (10). Consequently, minimizing J over $\boldsymbol{\alpha}$ can be easily implemented as there exist several efficient SVM solvers.

b) Computation of $\boldsymbol{\beta}$ given $\boldsymbol{\alpha}$: To optimize the group-sensitive kernel weights $\boldsymbol{\beta}$ with a fixed $\boldsymbol{\alpha}$, the objective function in (14) can be expressed as

$$J(\boldsymbol{\beta}) = \sum_{g=1}^{G}\sum_{g'=1}^{G}\sum_{m=1}^{M} \beta_m^g \beta_m^{g'} S_m^{gg'}(\alpha) - \sum_{i=1}^{N} \alpha_i \qquad (15)$$

where

$$S_m^{gg'}(\boldsymbol{\alpha}) = \frac{1}{2} \sum_{\{i|c(x_i)=g\}} \sum_{\{j|c(x_j)=g'\}} \alpha_i y_i \alpha_j y_j K_m(x_i, x_j). \qquad (16)$$

When $G = 1$, $S_m^{gg'}$ corresponds to $S_k(\boldsymbol{\alpha})$ in canonical MKL [22]. When $G > 1$, the samples within a group have the same label ($\{\pm 1\}$) based on the assumption that the intermediate representation *group* is introduced to capture the local data distribution for each subcategory. In this case, $S_m^{gg'}$ stands for the correlation of group g and g' over the mth kernel function. When g and g' have the same label, maximizing J over $\boldsymbol{\beta}$ is to maximize the intraclass similarity. When g and g' have different labels, maximizing J over $\boldsymbol{\beta}$ is to minimize the interclass similarity. Correspondingly, the optimization of J over $\boldsymbol{\beta}$ can be rewritten as

$$\max_{\boldsymbol{\beta}} \sum_{g=1}^{G}\sum_{g'=1}^{G}\sum_{m=1}^{M} \beta_m^g \beta_m^{g'} S_m^{gg'}(\boldsymbol{\alpha}). \qquad (17)$$

Note that the problem in (14) is not convex. Inspired by [23], instead of solving $\boldsymbol{\beta}$ directly, we use a normalized exponential weighting function to approximate the nonnegative $\boldsymbol{\beta}$. Particularly, $\boldsymbol{\beta}$ is determined by statistical property of the group and the parameters of the function, which are also learned from data. In this paper, such weighting function is defined as

$$\beta_m^g = \frac{\exp\left(a_m^g K_m^g + b_m^g\right)}{\sum_{m'=1}^{M} \exp\left(a_{m'}^g K_{m'}^g + b_{m'}^g\right)} \qquad (18)$$

where a_m^g and b_m^g are the parameters of the function, and K_m^g corresponds to a certain statistical property for the gth group over the mth kernel function. Let n_g be the number of samples in the gth group. In this paper, K_m^g is defined as

$$K_m^g = \frac{\sum_{\{i|c(x_i)=g\}} \sum_{\{j|c(x_j)=g\}} K_m(x_i, x_j)}{n_g^2}. \qquad (19)$$

As stated in [31], $J(\boldsymbol{\beta})$ is differentiable if the SVM solution is unique. Such condition can be guaranteed by the fact that all kernel matrices are strictly positive definite. Thus, we take derivatives of $J(\boldsymbol{\beta})$ w.r.t. a_m^g, b_m^g and use a gradient descent method [cf. (20) and (21)] to train the weighting function. Thus

$$\frac{\partial J(\boldsymbol{\beta})}{\partial a_m^g} = 2\sum_{l=1}^{M}\left(\sum_{i=1}^{G}\left(\beta_l^i S_l^{ig}(\boldsymbol{\alpha})\right)\beta_m^g K_m^g\left(\delta_m^l - \beta_l^g\right)\right) \qquad (20)$$

$$\frac{\partial J(\boldsymbol{\beta})}{\partial b_m^g} = 2\sum_{l=1}^{M}\left(\sum_{i=1}^{G}\left(\beta_l^i S_l^{ig}(\boldsymbol{\alpha})\right)\beta_m^g\left(\delta_m^l - \beta_l^g\right)\right). \qquad (21)$$

In (20) and (21), δ_m^l is 1 if $l = m$ and 0 otherwise. After updating the parameters of the weighting function, we get a new $\boldsymbol{\beta}$ and then solve a single kernel SVM as in Section IV-C.1.

c) Summarization of the GS-MKL optimization process:

Algorithm 1 GS-MKL Training Algorithm

1: Initialize a_m^g and b_m^g with small random numbers for $g = 1, \ldots, G$ and $m = 1, \ldots, M$.

2: **While** the termination criterion is not met **do**

3: Calculate kernel weights $\boldsymbol{\beta}$ as (18)

4: Calculate $K(x_i, x_j) = \sum_{m=1}^{M} \beta_m^{c(x_i)} \beta_m^{c(x_j)} K_m(x_i, x_j)$

5: Solve $\boldsymbol{\alpha}$ using the canonical SVM with $K(x_i, x_j)$

6: $a_m^g \Leftarrow a_m^g + \gamma^{(t)}(\partial J(\boldsymbol{\beta})/\partial a_m^g)$ for $g = 1, \ldots, G$ and $m = 1, \ldots, M$

7: $b_m^g \Leftarrow b_m^g + \lambda^{(t)}(\partial J(\boldsymbol{\beta})/\partial b_m^g)$ for $g = 1, \ldots, G$ and $m = 1, \ldots, M$

8: **end while**

The training algorithm of GS-MKL is summarized in Alg. 1. It simultaneously optimizes the coefficients of the classifier $\boldsymbol{\alpha}$ and the group-sensitive kernel weights $\boldsymbol{\beta}$. The termination criterion of the algorithm is that the minimal distance of $\boldsymbol{\beta}$ between two loops is below a predefined threshold or the count of the iteration process reaches the maximal iteration number. In Alg. 1, the step size of each iteration, $\gamma^{(t)}$ and $\lambda^{(t)}$, can be fixed as a small constant or determined with a line search method that needs additional canonical SVM optimizations for better convergence. Optimizing the classifier coefficients and group-sensitive kernel weights is a gradient descent wrapping canonical SVM solvent process. Note that the proposed algorithm does not guarantee convergence to the global optimum, and the initial parameters a_m^g and b_m^g may affect the quality of the solution.

V. Grouping Samples for GS-MKL

By partitioning training corpus from the same category into groups, GS-MKL learns the group-sensitive multikernel combinations together with the associated GS-MKL classifier. Unlike the classifier ensemble method [46], GS-MKL partitions samples from the same category into "natural" groups (subclasses) rather than the entire training data set. Accordingly, similar samples from the same category are partitioned into the same group, which corresponds to the subcategory of the object category. Intraclass diversity can be represented by the set of groups of the

same category but with diverse appearance. Interclass correlation can be represented by the correlation between the groups from different categories.

Since prior knowledge about the optimal group (subclass) number and sample grouping for GS-MKL is not available, a clustering method is employed as a heuristic grouping strategy. One straightforward solution is the simple hybrid grouping method, which combines the clustering method and GS-MKL directly and empirically identifies the optimal group numbers over a validation data set. Such strategy is based on the assumption that ideal data partition for GS-MKL can be approximated by enumerating the group number of the clustering method. Rather than solving sample grouping and GS-MKL training independently, two more compact grouping strategies are presented for GS-MKL. Details of these two grouping strategies are discussed as follows.

A. Looping Hybrid Grouping Method for GS-MKL

A simple grouping strategy for GS-MKL is the hybrid grouping method, which combines the clustering method (e.g., KMeans, Meanshift [57], and probabilistic latent semantic analysis (pLSA) [25]; cf. Section VI-C1) with GS-MKL directly and solve sample grouping and GS-MKL training independently. Different from such simple hybrid grouping method, we propose a looping hybrid grouping method where sample grouping interacts with GS-MKL training iteratively via shared kernel metric. Given a predefined group number, the training corpus from the same category is first partitioned into groups by the kernel-based clustering method, where UMK serves as initial kernel metric. GS-MKL is then trained over the grouped training samples and their group assignments. The newly learnt group-sensitive multikernel combinations go on to serve as the updated kernel metric and engage in the refinement of group assignments by the kernel-based clustering method. Accordingly, GS-MKL is trained over the newly grouped training samples and their corresponding group assignments. Such looping process iterates until reaching a predefined looping count or the object function in (15) reaching a local maximum. The optimal group number can be identified by enumerating all candidate group numbers and choosing the one with the best recognition performance over the validation set. As shown in the experiments, such looping hybrid method gains advantage against simple hybrid grouping methods.

In the implementation, GK-KMeans [55] is employed to cluster training corpus from the same category into groups. Generally speaking, GK-KMeans provides three advantages as follows.

1) GK-KMeans, which makes use of kernel functions to map data from the input space to the feature space, is capable of identifying nonlinearly separable clusters in the input space.
2) GK-KMeans provides a near globally optimal clustering solution robust to the initialization and local minima.
3) To deal with the G-clustering problem, subproblems with $1, \ldots, G-1$ groups are incrementally solved, making it useful to seek the best group number.

Suppose that one has a positive training set $\mathbf{X} = \{x_i\}_{i=1}^{Nc}$ of object category c to be clustered into G groups (i.e., $C_1, \ldots, C_G$) and the kernel matrix $\mathbf{K}$ over $\mathbf{X}$. According to [55], the clustering error can be computed by

$$D(C_1, \ldots, C_G) = \sum_{n=1}^{Nc} \sum_{g=1}^{G} \delta(x_n \in C_g) \|x_n - C_g\|^2,$$

$$\text{where } \|x_n - C_g\|^2 = K(n,n) - \frac{2\sum_{i=1}^{Nc} \delta(x_i \in C_g) K(n,i)}{\sum_{i=1}^{N} \delta(x_i \in C_g)} + \frac{\sum_{j=1}^{Nc}\sum_{i=1}^{Nc} \delta(x_j \in C_g)\delta(x_i \in C_g) K(j,i)}{\sum_{j=1}^{Nc}\sum_{i=1}^{Nc} \delta(x_j \in C_g)\delta(x_i \in C_g)}. \quad (22)$$

Algorithm 2 GK-KMeans

Input: Kernel matrix, number of groups G

Output: Final grouping results over sample points $x_1, x_2, \ldots, x_N$

1: For subclustering problems $g = 2$ to G do

2: For all sample points xn, $n = 1, \ldots, N$ do

// suppose $x_n \in C_l^*$

3: Run kernel k-means with initial groups

$$C_1^*, C_2^*, \ldots, C_l^* - \{x_n\}, C_{g-1}^*, x_n$$

and output groups

$$C^n, C^n, \ldots, C_{g-1}^n, C_g^n$$

4: Evaluate the clustering error D in (22) for the output groups

5: End for

6: Find the grouping result

$$C_1^*, C_2^*, \ldots, C_{g-1}^*, C_g^* \text{ with the minimal } D$$

7: End for

The algorithm of GK-KMeans can be summarized in Alg. 2. In Alg. 2, GK-KMeans initiates a clustering procedure with only one group. A new group is iteratively added by globally searching the best initial sample with the lowest clustering error as the new group and starting kernel k-means with the initialization consisting of existing clusters and the newly added group.

B. DDG for GS-MKL

In the looping hybrid grouping method, the optimal group number with the best performance over the validation set is identified by brutal search over all candidate group numbers. To alleviate the cost of enumerating all candidate group numbers, we propose the DDG method, which integrates the search of group number, partition of training corpus, and GS-MKL training. In DDG, a hierarchy of the training samples is maintained for a

given object category by iteratively splitting the samples at a leaf group into two disjoint new leaf groups. Given the hierarchies of training corpus, GS-MKL is trained over all training samples with their leaf group assignments. The learnt group-sensitive multikernel combination then serves as kernel metric, which goes on to be utilized in GK-KMeans to split the corresponding leaf group.

Algorithm 3 DDG

Input: Multiple Basic Kernel matrix

Output: GS-MKL classifier and final clustering results over sample points $x_1, x_2, \ldots, x_N$

1: Assign training samples from the same object category to the same group. Initiate a_m^g and b_m^g with small random numbers for $g \in \{1, \ldots, L\}$ and $m \in \{1, \ldots, M\}$.

2: Optimize $\boldsymbol{\beta}$ and $\boldsymbol{\alpha}$ with Alg. 1. $J_{\max} \leftarrow J(\boldsymbol{\beta})$

3: **while** the termination criterion is not met **do**

4: Breath-firstly choose a leaf group g'

5: Partition samples within the leaf group g' into two new leaf groups with Alg. 2 using multikernel combination in (10)

6: Update group assignment over samples of group g'

7: Fix $\boldsymbol{\alpha}$ and optimize $J(\boldsymbol{\beta})$ in (15) by the method in Section IV-B3b.

8: if $J_{\max} \geq J(\boldsymbol{\beta})$

9: Undo current loop

10: else

11: Optimize $\boldsymbol{\beta}$ and $\boldsymbol{\alpha}$ with Alg. 1

12: $J_{\max} \leftarrow J(\boldsymbol{\beta})$

13: End if

14: End while

The algorithm of DDG is summarized in Alg. 3. In particular, all samples from the same category are first assigned to the same leaf group and feed for GS-MKL training. Then, the learnt group-sensitive multikernel combination is utilized in GK-KMeans to partition the unique leaf group into two new leaf groups. Iteratively, GS-MKL is trained over training samples with their leaf group assignments, and a leaf group is breath-firstly chosen to split into two new leaf groups with the newly updated group-sensitive multikernel combinations. Such procedure iterates until splitting any leaf group further results in a decreased value of the object function in (15). Two examples of hierarchies for bicycle and aero plane are illustrated in Fig. 4. It can be observed that leaves with diverse appearances can be viewed as the subclasses of the object category.

VI. Experiments

In the experiments, object recognition is treated as the multiclass classification problem in one-versus-all setting. Since prior knowledge is not available about the subcategories of the object category, we empirically evaluate the optimal number of groups.

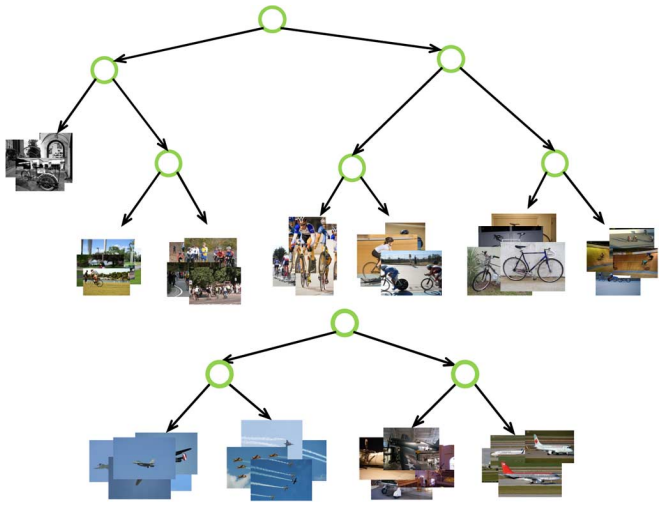


Fig. 4. Hierarchies of bicycle and aero plane from Pascal VOC2007. For better viewing, we demonstrate samples from the leaves only.

Then, the proposed GS-MKL is compared with several existing MKL methods on the four data sets. Finally, the performance of GS-MKL is compared with the results of the state-of-the-art methods.

A. Data Sets

Extensive experiments are performed on Caltech101 [6], Pascal VOC2007 [7], WikipediaMM [9], and Scene15 [58] data sets. Caltech101 involves 102 object categories, where each category contains 31 to 800 images. Pascal VOC2007 consists of 9963 images from 20 object categories, where 2501 images are provided for training, 2510 for validation, and 4952 for test, respectively. WikipediaMM data set contains 150 000 real-world web images from Wikipedia that cover 75 object categories. In our experiment, 33 categories, each of which contains more than 60 positive samples, are employed. Note that some categories not only share similar visual appearances but also produce semantic correlations, e.g., "house architecture" versus "gothic cathedral" and "military aircraft" versus "civil aircraft." Scene15 data set contains 15 scene categories. Each category has 200 to 400 positive samples with an average size of 300 by 250. Compared with the other three data sets, WikipediaMM exhibits higher intraclass diversity and interclass correlation with more background clutter and less alignment.

On Caltech101, WikipediaMM, and Scene data sets, we follow the experimental setups proposed by the designer, respectively [6], [9], [58], and adopt the recognition rate as the performance metric. Training and testing processes are repeated ten times, and the corresponding average performances are reported. On Caltech101 and WikipediaMM data sets, the numbers of randomly selected positive training samples are 10, 15, 20, 25, and 30 for each object category. The number of testing samples for each object category is fixed at 15. For object categories that have less than 45 images, training images are duplicated to maintain the balance of training samples. On Scene15 data set, 100 positive training samples are randomly selected for each category and the rest for testing. On Pascal VOC2007 data set, positive and negative training samples for each object category are provided by the data set designer [7]. For fair comparison with existing works [8], [28], [34],

[35], [48], [49], training and testing processes are conducted without repetition, and the official performance metric, average precision (AP) [7], is employed for performance evaluation.

B. Features and Kernels

Several feature descriptors are involved in the experiments. Two local appearance features (dense-color-SIFT (DCSIFT) and dense-SIFT (DSIFT) [3]) and two shape features (self-similarity (SS) [32] and PHOG [33]) are used. In particular, DCSIFT is computed in CIE-lab 3-channels over a square patch of radius with the spacing of r. We take $r = 4, 8,$ and 12 pixels to allow scalability. Likewise, DSIFT is calculated in gray channel. An SS descriptor is used to capture a correlation map of a 5×5 patch with its neighbors at every fifth pixel. The correlation map is quantized into ten orientations and three radial bins to form a 30 dim descriptor. We employ k-means to quantize these descriptors to obtain codebooks of size k (say, 400), respectively.

For PHOG, two SPKs of gradient orientation are calculated to measure the image similarity in shape. PHOG-180° employs 20 orientation bins, and PHOG-360° uses 40 orientation bins. For the other feature descriptors, we implement two kernel functions (i.e., SPK [3] and PDK [16]). For SPK, an image is divided into cells, and the features from the spatially corresponding cells are matched across two images. The resulting kernel is a weighted combination of histogram intersections from coarse to fine cells. A four-level pyramid is used with the cell number of 8×8, 4×4, 2×2, and 1×1, respectively. For PDK, local feature distributions of the K-nearest neighbors are matched across two images. The resulting kernel combines the local feature distributions at multiple scales, e.g., $K = 1, \ldots, k$, where k is set to (8, 16, 32) ranging from the finest to the coarsest neighborhood.

C. Experimental Results

Three set of experiments are carried out. First, effects of different grouping strategies are evaluated over the validation sets. Second, several existing MKL methods are implemented as baselines and compared with the proposed method. Finally, we conduct comparison of the proposed method to the state-of-the-art methods.

1) Comparison of Different Grouping Strategies: A validation set is utilized during the evaluation of grouping strategies. On Caltech101 and WikipediaMM data sets, 20 positive samples are randomly selected for training and 10 positive samples for validation to find out the optimal grouping results for each object category. On Pascal VOC2007 data set, 2501 training samples and 2510 validation samples are employed to find out the optimal grouping results as official setting in [7]. On Scene15 data set, 60 positive samples are randomly selected for training and 40 positive samples for validation. To evaluate the effects of different grouping strategies on GS-MKL, five sample grouping strategies for GS-MKL are involved in the experiments over the validation set, including

- *Hybrid_K-Means*: Hybrid grouping method (Hybrid_K-Means) combines K-Means and GS-MKL directly;
- *Hybrid_pLSA*: Hybrid grouping method (Hybrid_pLSA) combines pLSA [25] and GS-MKL directly;
- *Hybrid_GK-Kmeans*: Hybrid grouping method (Hybrid_GK-KMeans) combines GK-KMeans with GS-MKL directly;
- *Looping-Hybrid*: Looping hybrid grouping method (Looping-Hybrid) combines GK-KMeans with GS-MKL and iteratively conducts two processes with the shared group-sensitive multikernel combination (cf. Section V-A);
- *DDG*: Dynamic divisive grouping (DDG) method maintains a hierarchy of training samples and integrates GS-MKL training with the grouping procedure (cf. Section V-B).

In particular, K-Means and GK-KMeans, which are two widely used clustering methods based on explicit distance/kernel metric, are utilized to partition training samples into groups in Hybrid_K-Means and Hybrid_GK-KMeans, respectively. Compared with K-Means, GK-KMeans provides three main advantages (details can be found in Section V-A). In Hybrid_K-Means and Hybrid_GK-KMeans, UMK is utilized as distance/kernel metric. Hybrid_pLSA employs PLSA to investigate data distribution in the latent topic space. As a generative model, PLSA does not rely on any explicit distance/kernel metric. Bag-of-words representations [6], [13] over different types of low-level features are employed, and a sample is assigned to the most prominent topic as its group assignment. In Looping-Hybrid and DDG, group-sensitive multikernel combination is employed as kernel metric and updated during the iteration. The first four grouping strategies identify the optimal group number by enumerating. The potential group number of each object category ranges from 2 to 5, and the optimal group number is identified by empirical evaluation over the validation set. DDG does not need to tune the number of group, since the optimal group number and the corresponding grouping result are obtained together with the GS-MKL classifier over the same training data set.

In the first three methods, GS-MKL is trained directly over the partitioned training corpus obtained by different clustering methods. Hence, these methods are referred to as simple hybrid grouping methods where sample grouping and GS-MKL are independently solved. In the last two methods, sample grouping is integrated with GS-MKL training, providing a more compact solution. Three simple hybrid grouping strategies mentioned above serve as baselines to compare with two proposed grouping strategies. The best recognition results for five grouping strategies are listed in Table I. From the table, several observations have to be emphasized.

1) Five grouping strategies produce comparable performances, showing that different grouping strategies do not affect the performance of GS-MKL significantly.
2) PLSA in Hybrid_pLSA, which does not need explicit distance measure, seems more robust against K-Means using UMK.
3) Two hybrid grouping methods based on GK-KMeans obtain better performances against Hybrid_K-Means and Hybrid_pLSA across four data sets, which substantiates the advantage of GK-KMeans.
4) Two proposed grouping methods (i.e., Looping-Hybrid and DDG) consistently outperform three baseline

TABLE I
COMPARISON OF FIVE GROUPING STRATEGIES OVER THE VALIDATION SETS OF FOUR DATA SETS

	Caltech101 (Rec. in %)	PascalVOC2007 (AP. in %)	WikipediaMM (Rec. in %)	Scene15 (Rec. in %)
Hybrid_K-Means	78.6	54.2	60	81.2
Hybrid_pLSA	80.4	56.8	61.2	81.8
Hybrid_GK-KMeans	81.4	57.8	61.5	82.6
Looping-Hybrid	**82.1**	**58.2**	**62.1**	83.2
DDG	80.9	57.2	61.8	**83.3**

TABLE II
COMPARISON OF SVMS WITH INDIVIDUAL FEATURE/KERNEL AND GS-MKL WITH MULTIPLE FEATURES/KERNELS OVER THE VALIDATION SETS OF FOUR DATA SETS

Method	Kernel		Dataset			
	SPK	PDK	Caltech101 (Rec. in %)	Pascal VOC2007 (AP. in %)	WikipediaMM (Rec. in %)	Scene15 (Rec. in %)
SVM(DSIFT)	√		63.6	41.7	54.2	72.8
SVM(DSIFT)		√	60.7	43.3	51.4	67.5
SVM(DCSIFT)	√		62.8	44.5	55.8	74.6
SVM(DCSIFT)		√	61.3	43.4	52.9	68.2
SVM(SS)	√		58.5	39.5	47.7	67.6
SVM(SS)		√	56.7	40.3	46.8	64.3
SVM(PHOG)			57.5	36.7	44.3	59.4
GS-MKL(all features & kernels)			**82.1**	**58.2**	**62.1**	**83.2**

methods, showing that a sample grouping strategy where sample grouping interacts with GS-MKL training is more effective against that based on the common clustering method.

5) Looping-Hybrid obtains the best performance among five grouping strategies. This indicates that the performance of GS-MKL can be further improved if the grouping results are iteratively refined according to the newly learnt group-sensitive multikernel combinations.
6) The performance of DDG is a little lower than that of Looping-Hybrid. Although DDG adopts a more compact integration of sample grouping and GS-MKL training, deficiency can be drawn from the top-down divisive grouping strategy.

According to the results over the validation sets, Looping-Hybrid is employed as the only grouping strategy for GS-MKL in the following experiments, and the optimal group numbers derived from the validation sets are employed without further optimization.

To further demonstrate the effectiveness of GS-MKL, performances of GS-MKL over the validation sets are also compared with those of SVMs using individual features/kernels. In particular, the parameter of C is tuned over the validation set taking the values with 0.01, 0.1, 1, 10, and 100. As shown in Table II, GS-MKL consistently outperforms SVM by using group-sensitive multikernel combination.

2) Comparison with Other MKL Methods: We conduct comparisons between the proposed method and four baselines multiple kernel methods, including:

- *UMK*: Unweighted multikernel combination (UMK) adopts flat distribution for multikernel weights, and an SVM classifier is learnt for each object category;
- *MKL*: Canonical MKL (MKL) trains single classifier together with class-specific multikernel combination (implemented as [30]);
- *SS-MKL*: Sample-specific MKL (SS-MKL) trains single classifier for each object category with sample-specific multikernel combination;
- *MKL-ES*: MKL Ensemble (MKL-ES) partitions training image corpus into disjoint subsets and trains the MKL classifier over each subset as base classifier independently.

Under the same experimental setting, these four MKL methods are implemented as baselines. For fair comparison, we also compare the performance of the proposed method with the reported results of existing methods in Section VI-C3. Tables III–V and Fig. 5 list the comparison results over four data sets, respectively. Several observations can be drawn as follows.

1) On four data sets, GS-MKL outperforms four baseline methods. This demonstrates the advantage of GS-MKL against existing MKL methods by taking into account group-sensitive multikernel combination. However, on Caltech101, the advantage of GS-MKL against SS-MKL is less substantial when the number of the positive training sample is 10. This may be caused by the inefficacy of sample grouping strategy when training samples are too sparse. Performances of MKL, UMK, and MKL-ES are successively descent.
2) On Caltech101, WikipediaMM, and Scene15 data sets, GS-MKL obtains different improvements over the other four methods when the positive training sample number is larger than 10. It can be expected that GS-MKL is more effective in adapting multikernel combination to local data distribution when more training samples are available.
3) Two methods (i.e., GS-MKL and SS-MKL), taking into account intraclass variance, consistently outperform the methods using the global constant multikernel combination (i.e., MKL and UMK). This indicates that multikernel combination, which is adaptive to local data distribution, is more discriminative than the global constant multikernel combination.

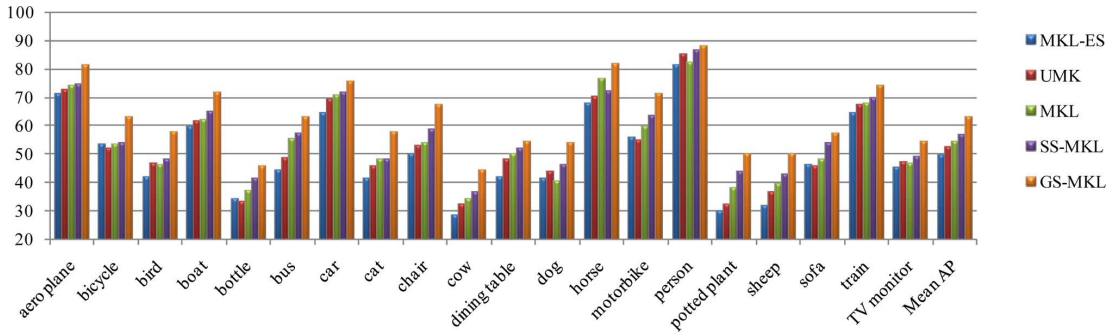


Fig. 5. Performance comparison of different MKL methods (i.e., MKL-ES, UMK, MKL, SS-MKL, and GS-MKL) on Pascal VOC2007. The resulting MAPs are 50.0, 52.5, 54.5, 57, and 63.4, respectively.

TABLE III
PERFORMANCE (REG. IN %) COMPARISON OF DIFFERENT MKL METHODS (I.E., MKL-ES, UMK, MKL, SS-MKL, AND GS-MKL) ON CALTECH101

Methods	Number of positive training samples per category				
	10	15	20	25	30
MKL-ES	58.5±1.3	64.6±1.2	67.8±1.0	68.8±0.9	69.5±0.9
UMK	65.6±0.8	68.8±0.7	70.4±0.6	71.7±0.6	73.2±0.5
MKL	66.4±1.2	70.6±1.1	73.9±1.0	75.1±0.8	75.8±0.7
SS-MKL	**69.1±1.5**	75.2±1.3	77.8±1.1	79.6±1.0	80.3±1.0
GS-MKL	66.2±1.3	**75.1±1.2**	**81.5±1.0**	**83.7±0.9**	**84.6±0.9**

TABLE IV
PERFORMANCE (REG. IN %) COMPARISON OF DIFFERENT MKL METHODS (I.E., MKL-ES, UMK, MKL, SS-MKL, AND GS-MKL) ON WIKIPEDIAMM

Methods	Number of positive training samples per category				
	10	15	20	25	30
MKL-ES	38.4±1.2	43.0±1.0	45.7±0.9	47.3±1.0	48.7±0.9
UMK	38.8±0.8	42.1±0.6	44.3±0.6	46.8±0.5	49.2±0.5
MKL	45.0±1.1	50.1±0.8	54.3±0.9	56.1±0.7	58.2±0.6
SS-MKL	47.1±1.6	53.4±1.2	57.3±1.0	59.8±1.1	61.4±1.0
GS-MKL	**49.3±1.3**	**56.9±1.2**	**61.9±1.0**	**64.8±0.9**	**66.9±0.9**

TABLE V
PERFORMANCE (REG. IN %) COMPARISON OF DIFFERENT MKL METHODS (I.E., MKL-ES, UMK, MKL, SS-MKL, AND GS-MKL) ON SCENE15

Methods	Number of positive training samples per category				
	10	20	30	50	100
MKL-ES	51.7±1.3	58.8±1.1	64.3±1.0	70.2±0.9	76.4±0.6
UMK	58.2±0.9	64.5±0.8	68.3±0.8	74.6±0.6	81.7±0.4
MKL	59.6±1.2	65.8±1.0	70.9±0.9	77.1±0.8	82.8±0.6
SS-MKL	**62.5±1.5**	67.2±1.2	72.6±1.1	78.1±1.0	84.2±0.9
GS-MKL	61.4±1.4	**68.3±1.3**	**74.2±1.0**	**80.9±0.9**	**86.5±0.8**

4) On Scene15, the relative improvements of GS-MKL and SS-MKL against the other three methods are less than those on Caltech101, Pascal VOC2007, and WikipediaMM data sets. This shows that different MKL methods yield comparable performances, when intraclass diversity and interclass correlation are not significant in the image corpus.
5) MKL yields higher performances than UMK on four data sets. This can be caused by the effectiveness of MKL in depressing noisy basic kernels.
6) MKL-ES yields performances comparable to UMK on WikipediaMM data set and the lowest performances on the other three data sets. Compared with the other four methods, MKL-ES does not gain any advantages. This can be attributed to the fact that learning base MKL classifiers over data partitions independently causes the insufficiency of training samples. GS-MKL learns group-sensitive multikernel combination and the classifier over the corpus of training data, obtaining a much higher recognition performance against MKL-ES.

To better understand the effectiveness of different MKL methods, we illustrate in Fig. 6 the mean probabilities of the top-k nearest neighborhoods for a testing sample that are from the same category on Pascal VOC2007 training set. Distance of two samples is computed by the learnt multikernel combination. From the figure, it can be observed that performance of GS-MKL is almost comparable to that of SS-MKL when the size of neighborhood is smaller than five. As the neighborhood becomes larger, GS-MKL obtains the highest probability for the sample neighborhood being the same class. This shows that GS-MKL is more effective against the other MKL methods in local adaptive multikernel combination learning when more training samples are available.

3) Comparison With Methods in the Literature: In Fig. 7, the performance of GS-MKL on Caltech101 is compared with the state-of-the-art results published in the literature [2], [3], [6], [11], [12], [18], [26], [27], [33], [49], [51], [52]. As shown in the figure, GS-MKL has achieved promising results comparable to the top performances of the state-of-the-art methods [26], [33]. When the number of positive training samples for each object category (N_{train}) is equal to 10, GS-MKL obtains the performance of 66.5%, which is a bit lower than the best reported one (69.5%) [26]. When $N_{\text{train}} > 10$, GS-MKL has obtained better performance against other reported results. When N_{train} is set to 30, the average recognition rate of GS-MKL reaches up to 84.4%, which is 3.56% higher than the best reported performance (81.5%) [26].

Table VI compares the performances of GS-MKL on Pascal VOC2007 to some other recently published methods [8], [28], [34], [35], [48], [49], and [50]. It is worthy to note that the approach INRIA_genetic [8] obtained the best performance in the Pascal VOC2007 challenge using nonlinear SVMs. The mean

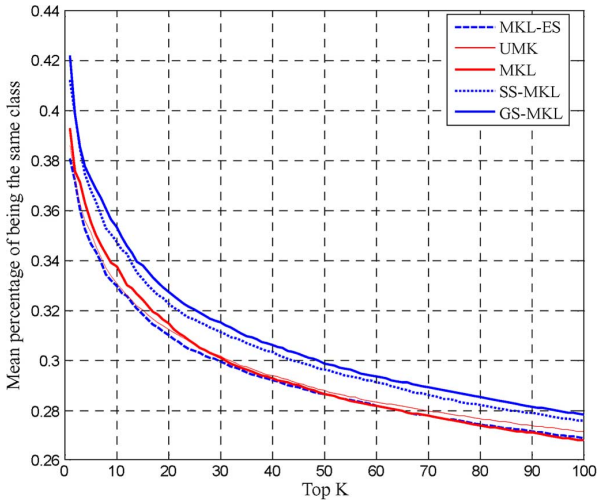


Fig. 6. Comparison of different multikernel combinations. x-Axis corresponds to the top-K nearest neighbor of a query sample. y-Axis denotes the mean percentage of the corresponding nearest neighbor being the same class of the query sample.

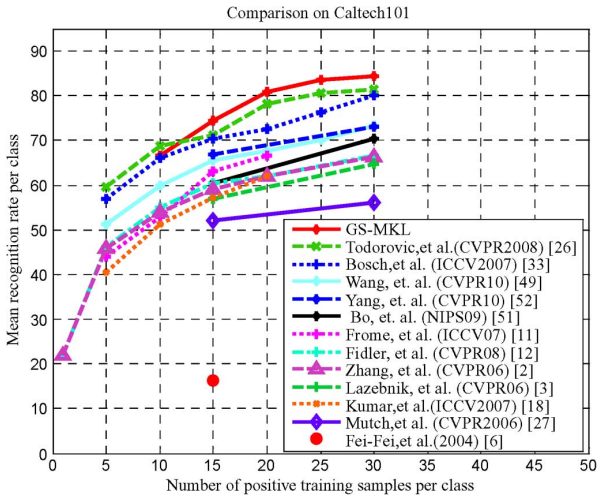


Fig. 7. Performance of GS-MKL and other recent methods on Caltech101 data set. GS-MKL: number of training samples (average recognition rate), 10 (66.5), 15 (74.5), 20 (81.0), 25 (83.5), and 30 (84.4).

AP of GS-MKL is 63.4%, which is better than that of [8], [28], [34], [35], [48], and [49] and slightly lower than that (63.5%) of [50]. To our best knowledge, [50] obtained the best ever reported result on Pascal VOC2007 by combining the classification system [8] and the costly sliding-window-based object localization. GS-MKL has obtained the best results for 10 out of 20 categories. Particularly, in the category of "chair," which exhibits significant intraclass diversity, GS-MKL has achieved over 10% relative improvements against the other methods.

On Scene15 data set, an average recognition rate of 81.0% is reported in [3] using the SIFT feature and SPK. The hybrid method [59] obtains an average recognition rate of 83.7% by fusing the pLSA model [25] and SVM. On the same data set, GS-MKL achieves an average recognition rate of 86.5%, which is slightly better than that in [3] and [59].

In summary, compared with the state-of-the-art results, GS-MKL obtains comparable or even better performances on Caltech101, Pascal VOC2007, and Scene15 data sets. This shows that exploring the contribution of multiple kernels over the groups of training data in a local adaptive manner is promising for object recognition.

TABLE VI
AP (IN %) OF GS-MKL AND OTHER METHODS ON THE PASCAL VOC2007 DATA SET

categories	[8]	[34]	[28]	[35]	[48]	[49]	[50]	GS-MKL
aero plane	77.5	63.0	65.0	65.0	72.7	74.8	77.2	**81.4**
bicycle	63.6	22.0	44.3	48.0	53	65.2	**69.3**	63.0
bird	56.1	14.0	48.6	44.0	49.1	50.7	56.2	**58.0**
boat	**71.9**	42.0	58.4	60.0	66.8	70.9	66.6	**71.9**
bottle	33.1	43.0	17.8	20.0	25.6	28.7	45.5	**46.0**
bus	60.6	50.0	46.4	49.0	52.4	**68.8**	68.1	63.0
car	78.0	62.0	63.2	70.0	69.9	78.5	**83.4**	75.7
cat	58.8	32.0	46.8	49.0	50	**61.7**	53.6	58.0
chair	53.5	37.0	42.2	50.0	46	54.3	58.3	**67.6**
cow	42.6	19.0	29.6	32.0	36.4	**48.6**	51.1	44.8
dining table	**54.9**	30.0	20.8	39.0	43.3	51.8	62.2	54.6
dog	45.8	29.0	37.7	40.0	43.9	44.1	45.2	**54.0**
horse	77.5	15.0	66.6	72.0	74.7	76.6	78.4	**81.7**
motorbike	64.0	31.0	50.3	59.0	59.5	66.9	69.7	**71.5**
person	85.9	43.0	78.1	81.0	83.4	83.5	86.1	**88.0**
potted plant	36.3	33.0	27.2	32.0	39	30.8	**52.4**	50.5
sheep	44.7	41.0	32.1	35.0	39.5	44.6	**54.4**	50.1
sofa	50.6	37.0	26.8	42.0	39.9	53.4	54.3	**57.7**
train	**79.2**	29.0	62.8	68.0	74.3	78.2	75.8	74.4
TV monitor	53.2	**62.0**	33.3	49.0	42.8	53.5	62.1	54.8
Mean AP	59.4	36.7	44.9	50.2	53.1	59.3	**63.5**	63.4

D. Time Complexity

We implemented GS-MKL in C++. In each iteration of Algorithm 1, we need to solve a canonical SVM problem with the group-sensitive kernel weights optimized by a gradient descent method. The time complexity of the gradient calculation is ignorable compared with the SVM solver. As those in canonical SVM solvers, using hot-start (i.e., providing previous $\boldsymbol{\alpha}$ as input) may accelerate the training process. Given the convergence termination criteria, the number of iterations before convergence depends on the training data and the step sizes. During training each category over 5000 image samples on Pascal VOC2007, the canonical MKL needs about 20 min, and GS-MKL needs 40–60 min to converge on a PC server (8 Corel 3.0 GHz, 8-GB RAM).

VII. CONCLUSION

In this paper, we consider the problem of object recognition and argue that both intraclass diversity and interclass correlation among images are crucial to improve the discriminative power of an object recognition method. To this end, "group" is introduced into the MKL framework as an intermediate representation between the object category and individual samples. GS-MKL is proposed to learn both the parameters of group-sensitive multikernel combinations and the classifier in a joint manner.

Rather than using simple hybrid grouping strategies that solve sample grouping and GS-MKL training independently,

two sample grouping strategies are proposed to integrate the two processes. It has been shown that the performance of GS-MKL does not significantly vary with different grouping strategies. A simple hybrid grouping strategy can boost GS-MKL against other multiple kernel methods. Furthermore, the performance of GS-MKL can be further improved using two proposed grouping strategies, respectively. On four benchmark data sets, promising results, which are comparable to the state-of-the-art, have been obtained by GS-MKL using existing visual feature/kernels, and significant improvements have been achieved over several existing MKL methods.

REFERENCES

[1] L. Fei-Fei and P. Perona, "A Bayesian hierarchical model for learning natural scene categories," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2005, pp. 524–531.

[2] H. Zhang, A. Berg, M. Maire, and J. Malik, "SVM-KNN: Discriminative nearest neighbor classification for visual category recognition," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2006, pp. 2126–2136.

[3] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2006, pp. 2169–2178.

[4] A. Bosch, A. Zisserman, and X. Muoz, "Image classification using random forests and ferns," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2007, pp. 1–8.

[5] M. Varma and D. Ray, "Learning the discriminative power-invariance trade-off," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2007, pp. 1–8.

[6] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental Bayesian approach testing on 101 object categories," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. Workshop Generative-Model Based Vis.*, 2004, p. 178.

[7] M. Everingham, L. VanGool, C. K. I. Williams, J. Winn, and A. Zisserman, The PASCAL visual object classes challenge 2007 (VOC2007) results [Online]. Available: http://www.Pascal-network.org/challenges/VOC/voc2007/workshop/index.html

[8] M. Marszałek, C. Schmid, H. Harzallah, and J. Weijer, "Learning object representations for visual object class recognition," in *Proc. Workshop Visual Recognit. Challenge, IEEE Int. Conf. Comput. Vis.*, 2007, pp. 1–8.

[9] [Online]. Available: www.imageclef.org/2008/wikipedia

[10] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan, "Multiple kernel learning, conic duality, and the SMO algorithm," in *Proc. Int. Conf. Mach. Learn.*, 2004, p. 6.

[11] A. Frome, Y. Singer, F. Sha, and J. Malik, "Learning globally-consistent Local distance functions for shape-based image retrieval and classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2007, pp. 1–8.

[12] S. Fidler, M. Boben, and A. Leonardis, "Similarity-based cross-layered hierarchical representation for object categorization," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.

[13] J. Sivic, B. Russell, A. A. Efros, and A. Zisserman, "Discovering objects and their location in images," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2005, pp. 370–377.

[14] G. Wang, Y. Zhang, and L. Fei-Fei, "Using dependent regions for object categorization in a generative framework," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2006, pp. 1597–1604.

[15] K. Grauman and T. Darrell, Pyramid match kernels: Discriminative classification with sets of image features MIT, Cambridge, MA, Tech. Rep. MIT CSAIL TR 2006-020, Mar. 2006.

[16] L. Haibin and S. Soatto, "Proximity distribution kernels for geometric context in category recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2007, pp. 1–8.

[17] K. Q. Weinberger, J. Blitzer, and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," in *Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2005.

[18] A. Kumar and C. Sminchisescu, "Support kernel machines for object recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2007, pp. 1–8.

[19] O. Chum and A. Zisserman, "An exemplar model for learning object classes," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2007, pp. 1–8.

[20] T. Malisiewicz and A. A. Efros, "Recognition by association via learning per-exemplar distances," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.

[21] J. Platt, *Advances in Kernel Methods-Support Vector Learning, chapter Fast Training of Support Vector Machines using Sequential Minimal Optimization*. Cambridge, MA: MIT Press, 1998, pp. 185–208.

[22] S. Sonnenburg, G. Raetsch, C. Schaefer, and B. Scholkopf, "Large scale multiple kernel learning," *J. Mach. Learn. Res.*, vol. 7, pp. 1531–1565, Jul. 2006.

[23] M. Gonen and E. Alpaydin, "Localized multiple kernel learning," in *Proc. IEEE Int. Conf. Mach. Learn.*, 2008, pp. 352–359.

[24] D. Lowe, "Object recognition from local scale-invariant features," in *Proc. IEEE Int. Conf. Comput. Vis.*, 1999, pp. 1150–1157.

[25] T. Hofmann, "Probabilistic latent semantic indexing," in *Proc. ACM Special Interest Group Inform. Retrieval*, 1999, pp. 50–57.

[26] S. Todorovic and N. Ahuja, "Learning subcategory relevancies for category recognition," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.

[27] J. Mutch and D. G. Lowe, "Multiclass object recognition with sparse, localized features," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2006, pp. 11–18.

[28] G. Wang, D. Hoiem, and D. Forsyth, "Learning image similarity from flickr groups using stochastic intersection kernel machines," in *Proc. Multimedia Inform. Retrieval*, 2008, pp. 428–435.

[29] Y. Lin, T. Liu, and C. Fuh, "Local ensemble kernel learning for object category recognition," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2007, pp. 1–8.

[30] A. Rakotomamonjy, F. Bach, Y. Grandvalet, and S. Canu, "SimpleMKL," *J. Mach. Learn. Res.*, vol. 9, pp. 2491–2521, Nov. 2008.

[31] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee, "Choosing multiple parameters for support vector machines," *Mach. Learn.*, vol. 29, no. 1–3, pp. 131–159, Jan. 2002.

[32] E. Shechtman and M. Irani, "Matching local self-similarities across images and videos," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2007, pp. 1–8.

[33] A. Bosch, A. Zisserman, and X. Munoz, "Representing shape with a spatial pyramid kernel," in *Proc. ACM Int. Conf. Image Video Retrieval*, 2007, pp. 401–408.

[34] C. Galleguillos, A. Rabinovich, and S. Belongie, "Object categorization using co-occurrence, location and appearance," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.

[35] F. Khan, J. Weijer, and M. Vanrell, "Top-down color attention for object recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2009, pp. 979–986.

[36] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Proc. 9th Eur. Conf. Comput. Vis.*, Graz, Austria, May 2006, pp. 404–417.

[37] B. Babenko, S. Branson, and S. Belongie, "Similarity metrics for categorization: From monolithic to category specific," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2009, pp. 293–300.

[38] P. Gehler and S. Nowozin, "On feature combination for multiclass object classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2009, pp. 221–228.

[39] M. Gonen and E. Alpaydin, "Localized multiple kernel machines for image recognition," in *Neural Information Processing Systems—Workshop on Understanding Multiple Kernel Learning Method*. Cambridge, MA: MIT Press, 2009.

[40] I. Catalin, B. Liefeng, and S. Cristian, "Structural SVM for visual localization and continuous state estimation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2009, pp. 1157–1164.

[41] C. Yang and M. Dong, "Region-based image annotation using asymmetrical support vector," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2006, pp. 2057–2063.

[42] J. R. Smith, M. Naphade, and A. Natsev, "Multimedia semantic indexing using model vectors," in *Proc. IEEE Int. Conf. Multimedia Expo.*, 2003, pp. 445–448.

[43] O. Duchenne, I. Laptev, J. Sivic, F. Bach, and J. Ponce, "Automatic annotation of human actions in video," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2009, pp. 1491–1498.

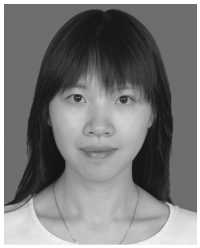
[44] K. Motoaki, N. Shinichi, and N. Alexander, "A procedure of adaptive kernel combination with kernel-target alignment for object classification," in *Proc. ACM Int. Conf. Image Video Retrieval*, 2009, p. 23.

[45] A. Bosch, A. Zisserman, and X. Munoz, "Image classification using ROIs and multiple kernel learning," in *Proc. Int. J. Comput. Vis.*, 2008.

[46] L. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*. Hoboken, NJ: Wiley, 2004.

[47] T. Tuytelaars, "Dense interest points," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 2281–2288.

[48] M. Guillaumin, J. Verbeek, and C. Schmid, "Multimodal semi-supervised learning for image classification," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 902–909.
[49] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 3360–3367.
[50] Y. Boureau, F. Bach, L. Yann, and J. Ponce, "Learning midlevel features for recognition," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 2559–2566.
[51] L. Bo and C. Sminchisescu, "Efficient match kernels between sets of features for visual recognition," in *Proc. Neural Inform. Process. Syst.*, 2009, pp. 135–143.
[52] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 1794–1801.
[53] J. J. Yang, Y. Li, Y. Tian, L. Y. Duan, and W. Gao, "Group-sensitive multiple kernel learning for object categorization," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2009, pp. 436–443.
[54] J. J. Yang, Y. Li, Y. Tian, L. Y. Duan, and W. Gao, "Per-sample multiple kernel approach for visual concept learning," *EURASIP J. Image Video Process.*, vol. 2010, pp. 461 450-1–461 450-13, Jan. 2010, DOI:10.1155/2010/461450.
[55] G. F. Tzortzis and A. C. Likas, "The global kernel k-means algorithm for clustering in feature space," *IEEE Trans. Neural Netw.*, vol. 20, no. 7, pp. 1181–1194, Jul. 2009.
[56] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.
[57] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.
[58] L. Fei-Fei and P. Perona, "A Bayesian hierarchical model for learning natural scene categories," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2005, pp. 524–531.
[59] A. Bosch, A. Zisserman, and X. Muñoz, "Scene classification using a hybrid generative/discriminative approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 4, pp. 712–727, Apr. 2008.



Jingjing Yang (S'11) received the B.S. degree in automation from Civil Aviation University of China, Tianjin, China, the M.S. degree in automation from Tianjin University, Tianjin, and the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2003, 2006, and 2011, respectively.

She is currently a Data Analyst with Guangdong Power Grid Corporation, China Southern Power Grid Co., Ltd., Guangzhou, China. Her research interests include machine learning, data mining, object recognition, and computer vision.

Yonghong Tian (M'05–SM'10) received the M.S. degree from the School of Computer Science, University of Electronic Science and Technology of China, Chengdu, China, in 2000 and the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2005.

He is currently an Associate Professor with the National Engineering Laboratory for Video Technology, School of Electronics Engineering and Computer Science, Peking University, Beijing. His research interests include machine learning, multimedia analysis, retrieval, interaction, and copyright protection.

Dr. Tian is a member of the Association for Computing Machinery.

Ling-Yu Duan (M'06) received the M.Sc. degree in automation from the University of Science and Technology of China, Hefei, China, in 1999, the M.Sc. degree in computer science from National University of Singapore, in 2002, and the Ph.D. degree in information technology from The University of Newcastle, Newcastle, Australia, in 2007.

Since 2008, he has been with Peking University, Beijing, China, where he is currently an Associate Professor with the School of Electronics Engineering and Computer Science. Before that, he was a Research Scientist with the Institute for Infocomm Research, Singapore, from 2003 to 2008. His current interests are in the areas of multimedia content analysis; computer vision; large-scale multimedia information mining and retrieval; and mobile media computing for multimedia authoring, sharing, and advertising. He has authored more than 60 publications in these areas and has five filed U.S. patents or pending applications.

Dr. Duan is a member of the Association for Computing Machinery.

Tiejun Huang (M'01) received the B.S. and M.S. degrees from the Department of Automation, Wuhan University of Technology, Wuhan, China, in 1992 and the Ph.D. degree from the School of Information Technology and Engineering, Huazhong University of Science and Technology, Wuhan, in 1999.

He was a Postdoctoral Researcher from 1999 to 2001 and a Research Faculty Member with the Institute of Computing Technology, Chinese Academy of Sciences. He was also the Associated Director (from 2001 to 2003) and the Director (from 2003 to 2006) of the Research Center for Digital Media in Graduate School at the Chinese Academy of Sciences. He is currently a Professor with the National Engineering Laboratory for Video Technology, School of Electronics Engineering and Computer Science, Peking University, Beijing, China. His research interests include digital media technology, digital library, and digital rights management.

Dr. Huang is a member of the Association for Computing Machinery.

Wen Gao (M'92–SM'05–F'09) received the M.S. degree in computer science from Harbin Institute of Technology, Harbin, China, in 1985 and the Ph.D. degree in electronics engineering from the University of Tokyo, Tokyo, Japan, in 1991.

He was a Professor in computer science with Harbin Institute of Technology from 1991 to 1995 and a Professor in computer science with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, from 1996 to 2005. He is currently a Professor with the Institute of Digital Media, School of Electronics Engineering and Computer Science, Peking University, Beijing. He has been leading research efforts to develop systems and technologies for video coding, face recognition, sign language recognition and synthesis, and multimedia retrieval. He has published four books and over 500 technical articles in refereed journals and proceedings in the areas of signal processing, image and video communication, computer vision, multimodal interface, pattern recognition, and bioinformatics. His current research interests include signal processing, image and video communication, computer vision, and artificial intelligence.

Dr. Gao received many awards, including five national awards for research achievements and activities. He did many services to the academic society, such as the General Co-chair of the IEEE International Conference on Multimedia and Expo in 2007, and the ACM International Conference on Multimedia in 2009, and the Head of Chinese delegation to the Moving Picture Expert Group of International Standard Organization since 1997. He is also the Chairman of the working group responsible for setting a national Audio Video Coding Standard for China.