Efficient Discriminative Local Learning for Object Recognition

Yen-Yu Lin[1,2] Jyun-Fan Tsai[1] Tyng-Luh Liu[1]
[1]Institute of Information Science, Academia Sinica, Taipei 115, Taiwan
[2]Department of CSIE, National Taiwan University, Taipei 106, Taiwan
{yylin, jftsai, liutyng}@iis.sinica.edu.tw

Abstract

Although object recognition methods based on local learning can reasonably resolve the difficulties caused by the large variations in images from the same category, the high risk of overfitting and the heavy computational cost in training numerous local models (classifiers or distance functions) often limit their applicability. To address these two unpleasant issues, we cast the multiple, independent training processes of local models as a correlative multi-task learning problem, and design a new boosting algorithm to accomplish it. Specifically, we establish a parametric space where these local models lie and spread as a manifold-like structure, and use boosting to perform local model training by completing the manifold embedding. Via sharing the common embedding space, the learning of each local model can be properly regularized by the extra knowledge from other models, while the training time is also significantly reduced. Experimental results on two benchmark datasets, Caltech-101 and VOC 2007, support that our approach not only achieves promising recognition rates but also gives a two order speed-up in realizing local learning.

1. Introduction

One of the major challenges in designing an object recognition system is to resolve the difficulty arising from the large intraclass variations depicted in images from the same object category. Such complications can be due to both intrinsic and extrinsic factors, and they significantly increase the complexity of separating data of one class from those of other classes or *the rest of the world*. In Figure 1, examples of images with substantial intraclass variations are shown to illustrate the scenario.

Local learning is effective in tackling large intraclass variations in data. Unlike the global approach, it considers multiple local models, each of which is learned to account for only a subset of data. For instance, to detect both frontal and profile faces, Schneiderman and Kanade [26] use multiple *view-based* detectors to specifically uncover faces within a certain range of poses. Similar strategies are also adopted in recent researches for image classification and recognition, *e.g.*, [12, 13, 19, 21]. However, except for certain cases such as the face images that can be reasonably aligned and divided into subsets by their pose, it is generally a difficult task to partition image data into meaningful subsets, each of which is covered by one local model. In the aforementioned approaches, the learning is performed for each sample or its neighborhood, and results in local models as many as the number of training data. In [12, 13, 21], the local model corresponds to a distance function, while in [14, 19] it relates to a kernel matrix.

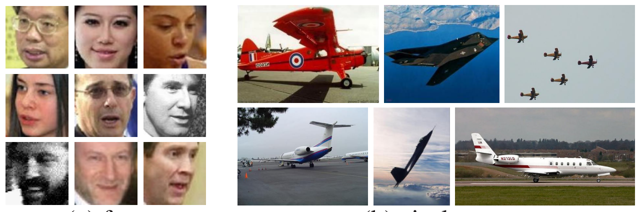

Figure 1. Images from categories (a) face and (b) airplane, with large intraclass variations in appearance, shape, color, scale, *etc*.

Even for a modest size of training data, learning the *sample-specific* local models may already not scale well. Take, for example, the two popular benchmark datasets for object recognition, Caltech-101 [10] and VOC 2007 [9]. Each comprises thousands of images, and learning all the local models often takes time in the order of hours or even days. It almost makes parameter tuning, a key step for obtaining satisfactory classification results, infeasible. Further, local learning may suffer from the risk of overfitting in that it often considers a relatively small group of training data. Addressing these two unfavorable issues of local learning will be among the main focuses of this work.

1.1. Related work

Local learning has been an active research topic in machine learning [2]. It draws on the idea that a local model should most likely better characterize the distinctive properties shared by a small subset of data than a global one does for the whole data. Consequently adopting multiple locally adaptive models would achieve better performances. For example, to enhance the nearest neighbor rule for classification, Domeniconi and Gunopulos [8] propose a *local flexible metric* by adaptively reweighting dimensions of the feature space according to the sample location.

An important issue in local learning is how to deploy the local models among data points. In [16], Kim and Kittler use k-means clustering to partition data into several clusters, and train a *linear discriminant analysis* (LDA) classifier for each cluster. Dai *et al*. [7], motivated by that more local models should be located near data with high risks of being misclassified, propose the *responsibility mixture model*, in which an EM algorithm is adopted to place local classifiers according to the distribution of data uncertainty. However, for approaches of this kind, it is hard to pre-determine the optimal number of local models. After all, the Gaussian assumptions for the data distribution or uncertainty distribution are not always valid.

Alternatively, in [12, 13, 21], a local model is specifically designed for each training sample. Thus, no assumptions about the data distribution are required. In addition, local learning in these works is often coupled with feature fusion. That is, not only the discriminant functions but also the discriminant visual features are jointly selected to enhance the power of each local model. Nevertheless, training the sample-specific models is time-consuming.

Pertaining to object recognition, the tasks of learning a local model from a given dataset are typically correlated. As is pointed out from the literature of multi-task learning, *e.g*., [1, 6, 29], investigating related tasks *jointly* in most cases can achieve a considerable performance improvement than *independently*, since the extra knowledge from other tasks may convey useful information to the completion of the underlying task.

1.2. Our approach

Our method focuses on efficiently learning sample-specific local models to improve the accuracies on object recognition. In our framework, each local model is a classifier derived by boosting, and has a special form such that the collection of them spreads as a manifold-like structure in the resulting classifier space. By respecting this global structure of local classifiers, we design a boosting algorithm to learn them jointly. The main contributions of our approach can be characterized as follows: 1) We propose to cast the independent training procedures of local classifiers as a multi-task learning problem. 2) We introduce a boosting algorithm that solves the underlying multi-task problem with good efficiency, scales well to the data size, and produces local classifiers with proper regularization and less redundancy. 3) We establish a new way of carrying out *multiple kernel learning* (MKL) [17, 25] when several image descriptors are used to account for the given data. It can yield recognition rates comparable to those reported by the state-of-the-art MKL tools, *e.g*., [25].

The proposed approach is related to *JointBoost* [29] in the sense that multiple boosted classifiers are simultaneously generated under a specific form of multi-task learning. However, our method can accomplish it more efficiently. We also note that our use of *dyadic hypercuts*, introduced in Moghaddam and Shakhnarovich [22], as the weak learners for boosting is pivotal to the formulation in that they can effectively capture useful information in a kernel matrix, and elegantly connect multiple kernel learning with boosting.

2. Multi-task local learning

We begin by specifying the notations used in this work, defining the problem of local learning for object recognition, and describing its link to the multi-task learning.

2.1. Notations

Since the multi-class object recognition can always be reduced to an array of two-class problems by adopting *one-against-one* or *one-against-all* rules, we assume a binary dataset $S = \{\mathbf{x}_n \in \mathcal{X}, y_n \in \pm 1\}_{n=1}^N$, where N is the data size and $\mathcal{X}$ denotes the input space.

In view of the complexity of visual object recognition, it is hard to find a universal descriptor to well characterize the whole dataset. Instead, we consider representing each $\mathbf{x}_n$ with totally M kinds of different descriptors, *i.e*., $\mathbf{x}_n = \{\mathbf{x}_{n,m} \in \mathcal{X}_m\}_{m=1}^M$, and each descriptor is associated with a distance measure $d_m : \mathcal{X}_m \times \mathcal{X}_m \to \mathbb{R}$.

The useful data representations are often of high dimensional and in diverse forms, such as vectors [23], histograms [5], bags of features [34], or pyramids [18]. To avoid the difficulties caused by working with these varieties, we represent data under each descriptor by a kernel matrix. And it leads to M kernel functions $\{k_m\}_{m=1}^M$ together with the corresponding kernel matrices $\{K_m\}_{m=1}^M$:

$$K_m(n, n') = k_m(\mathbf{x}_n, \mathbf{x}_{n'}) = \exp(-\gamma_m d_m^2(\mathbf{x}_n, \mathbf{x}_{n'})) \quad (1)$$

where γ_m is a positive constant.

2.2. Local learning

Our goal is to carry out local learning by deriving a classifier f_i for each training sample $\mathbf{x}_i$ such that f_i is expected to give good performances for testing samples falling around $\mathbf{x}_i$. To this end, we specify the *neighborhood* of $\mathbf{x}_i$ with a weight distribution $\mathbf{w}_i = \{w_{i,n}\}_{n=1}^N$ over S by

$$w_{i,n} = \begin{cases} 1/C, & \text{if } \mathbf{x}_n \in C\text{-NN of } \mathbf{x}_i, \\ 0, & \text{otherwise,} \end{cases} \quad (2)$$

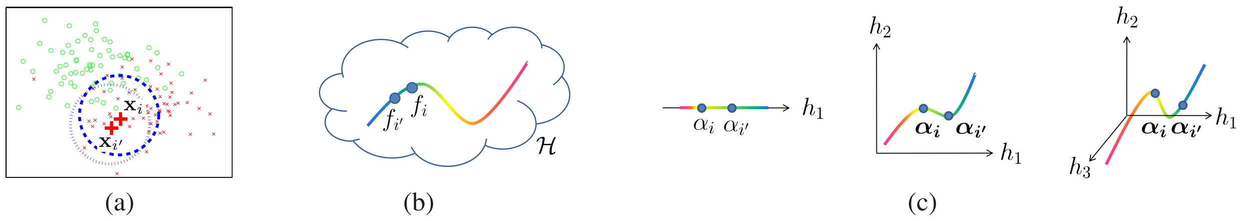

Figure 2. (a) Two training samples, $\mathbf{x}_i$ and $\mathbf{x}_{i'}$, and their respective neighborhoods (denoted by blue dotted circles). (b) All the local classifiers, including f_i and $f_{i'}$, spread as a manifold-like structure in the high-dimensional classifier space induced by $\mathcal{H}$. (c) These classifiers can be obtained by incrementally completing the manifold embedding. The embedding space is spanned by weak learners $\{h_t\}$, and the new coordinates of f_i and $f_{i'}$ in the embedding space are their respective ensemble coefficients $\boldsymbol{\alpha}_i$ and $\boldsymbol{\alpha}_{i'}$.

where C-NN of $\mathbf{x}_i$ denotes the C nearest neighbors of $\mathbf{x}_i$ (including $\mathbf{x}_i$ itself). In cases that multiple descriptors are used, we compute $\mathbf{w}_i$ for data under each representation and average the outcomes. Then each local classifier f_i can be obtained by coupling an optimal function with suitable features to best discriminate the weighted dataset $S_i = \{\mathbf{x}_n, y_n, w_{i,n}\}_{n=1}^N$. The purpose of our local learning is to learn such classifiers $\{f_i\}_{i=1}^N$.

Specifically, each f_i is resulted from the boosting algorithm, and consists of a set of weak learners. Let $\mathcal{H}$ denote the domain of weak hypotheses. Our formulation assumes that each weak learner $h \in \mathcal{H}$ is generated by referencing only the M kernels defined in (1). The main advantage of so doing is that it uses these kernels as the unified information bottleneck, and enjoys the convenience of working with different descriptors and distance measures. In our discussions below, we will treat the training of each local classifier as a particular *task*. It follows that the numbers of tasks and training samples are both N. For sake of clarity, hereafter we will use subscript i as the index to tasks or classifiers, and n to the training samples.

2.3. Multi-task formulation

While local learning is effective for addressing complicated vision problems like object recognition, care must be taken to ensure that the learning procedure has been properly done for a given dataset. In particular, we pinpoint the following two critical issues related to learning local classifiers $\{f_i\}_{i=1}^N$. First, each f_i is learned with a small portion of training data. When the number of weak learner candidates $|\mathcal{H}|$ is large or infinite, f_i is at the high risk of overfitting. Second, learning a local classifier for each training sample is indeed an inefficient procedure. And the situation gets worse when dealing with a large dataset, but it is often the case for vision applications.

To alleviate the two above-mentioned unfavorable effects, we view completing the independent training processes of $\{f_i\}_{i=1}^N$ as a correlative multi-task learning problem. This is accomplished by assuming these local classifiers share the same weak learners, but have their respective ensemble coefficients, *i.e.*,

$$f_i(\mathbf{x}) = \sum_{t=1}^{T} \alpha_{i,t} h_t(\mathbf{x}), \text{ for } i = 1, 2, ..., N. \tag{3}$$

With (3), all the classifiers will be learned jointly. We will later describe a systematic way for constructing $\{f_i\}_{i=1}^N$ with multi-task learning in the next section. For now we give justifications on why the two unfavorable effects can be eased by setting local classifiers in the form of (3).

Proper regularization. It is instructive to think as if all the local classifiers lie in a parametric space induced by $\mathcal{H}$. The space dimension is $|\mathcal{H}|$, and the coordinates of a classifier (i.e., a point) in the space are its ensemble coefficients over the weak learners. Consider now two local classifiers f_i and $f_{i'}$ corresponding to two nearby samples $\mathbf{x}_i$ and $\mathbf{x}_{i'}$. According to (2), the highly overlapping neighborhoods, $\mathbf{w}_i$ and $\mathbf{w}_{i'}$, should lead to the high similarity between f_i and $f_{i'}$. Hence, f_i and $f_{i'}$ are expected to be close in the classifier space. Extending the concept to all the classifiers, it suggests that they would spread as a manifold-like structure. We instill this property into regularizing the training process of each classifier. In other words, we learn all the local classifiers simultaneously by respecting the underlying manifold structure. This could lessen the instability (overfitting) problem when otherwise independently learning each classifier with insufficient training data would cause. Observe that constructing the local classifiers of the form in (3) can capture the idea faithfully. As we will show later, while the ensemble coefficients $\{\alpha_{i,t}\}_{t=1}^T$ are to represent the embedding coordinates of f_i, the shared weak learners $\{h_t\}_{t=1}^T$ can be optimized to span the embedding space. An illustration of the regularization is given in Figure 2.

Redundancy elimination. If $\{f_i\}_{i=1}^N$ are learned independently, redundancy can easily become an unpleasant concern. The phenomenon can be understood from both the aspects of weak learners and training data. Since a weak learner generally yields similar performances in the related tasks, learning each of them separately is to overlook such relatedness. On the other hand, as a training sample can be accessed by multiple tasks, inefficiency can occur when measuring the loss induced by the sample is evaluated independently throughout the relevant tasks. Our strategy is to learn all local classifiers of (3) jointly so that information redundancy among them can be reasonably circumvented with a substantial speed-up in the training process.

3. A multi-task boosting algorithm

The main theme of this section is to detail the steps of the proposed multi-task boosting algorithm, and to discuss its justifications and useful properties.

3.1. Design of weak learners

We consider dyadic hypercuts [22] as the weak learners in that they can achieve good classification performances, and be generated by referencing only the kernel functions $\{k_m\}_{m=1}^M$ or matrices $\{K_m\}_{m=1}^M$ of (1). A dyadic hypercut h is specified by a kernel and a pair of training samples of opposite labels. Specifically, h is parameterized by positive sample $\mathbf{x}_n$, negative sample $\mathbf{x}_{n'}$, and kernel function k_m, and can be expressed by

$$h(\mathbf{x}) = sign(k_m(\mathbf{x}_n, \mathbf{x}) - k_m(\mathbf{x}_{n'}, \mathbf{x}) - \theta), \tag{4}$$

where θ is for thresholding. The size of the resulting pool of weak learners is $|\mathcal{H}| = N^+ \times N^- \times M$, where N^+ and N^- are the numbers of positive and negative training data. To have an efficient boosting process, we randomly sample a subset of weak learners from the pool at each iteration.

3.2. The boosting algorithm

The multi-task setting of local learning is accomplished via a boosting algorithm, in which task i is to learn classifier f_i as in (3) with the weighted dataset $S_i = \{\mathbf{x}_n, y_n, w_{i,n}\}_{n=1}^N$. Steps of the proposed multi-task boosting are listed in Algorithm 1. Note that the algorithm maintains a two-dimensional weight array $\{w_{i,n}^{(t)}\}_{i,n=1}^N$ to link successive iterations, where $w_{i,n}^{(t)}$ denotes the weight of $\mathbf{x}_n$ in task i at iteration t. In what follows, we describe the details of running iteration t.

We start by defining the loss function of task i. At iteration t, $f_i = \sum_{\tau=1}^{t-1} \alpha_{i,\tau} h_\tau$ is a linear combination of the $(t-1)$ selected weak learners. With the *exponential loss* model [11], the loss of f_i with respect to S_i is

$$\mathcal{L}(f_i, S_i) = \sum_{n=1}^{N} w_{i,n} \exp(-y_n f_i(\mathbf{x}_n)). \tag{5}$$

Since all the classifiers are trained jointly, a reasonable choice of the joint objective function is the total loss induced by these classifiers in all the tasks, *i.e.*,

$$\text{Loss} = \sum_{i=1}^{N} \mathcal{L}(f_i, S_i). \tag{6}$$

Algorithm 1: *Multi-task Boosting for Local Learning*

```
Input  : $N$ tasks: task $i$ involves weighted dataset
             $S_i = \{\mathbf{x}_n, y_n, w_{i,n}\}_{n=1}^N$.
Output: Local classifiers $\{f_i\}_{i=1}^N$, where
             $f_i(\mathbf{x}) = \sum_{t=1}^T \alpha_{i,t} h_t(\mathbf{x})$.
Initialize: $w_{i,n}^{(1)} = w_{i,n}$, for $i, n = 1, 2, ..., N$.
for $t \leftarrow 1, 2, \ldots, T$ do
    1. Compute the cross-task data weights $\{\tilde{w}_n^{(t)}\}_{n=1}^N$:
           $\tilde{w}_n^{(t)} = \sum_{i=1}^N w_{i,n}^{(t)}$.
    2. Select the optimal dyadic hypercut $h_t$:
           $h_t = \arg\min_h \sum_{n=1}^N \tilde{w}_n^{(t)} \cdot 1_{[h(\mathbf{x}_n) \neq y_n]}$.
    3. Compute task-wise weighted errors $\{\epsilon_{i,t}\}_{i=1}^N$:
           $\epsilon_{i,t} = \sum_{n=1}^N w_{i,n}^{(t)} \cdot 1_{[h_t(\mathbf{x}_n) \neq y_n]}$.
    4. Compute task-wise weighted accuracies $\{c_{i,t}\}_{i=1}^N$
           $c_{i,t} = \sum_{n=1}^N w_{i,n}^{(t)} \cdot 1_{[h_t(\mathbf{x}_n) = y_n]}$.
    5. Set task-wise ensemble coefficients $\{\alpha_{i,t}\}_{i=1}^N$:
           $\alpha_{i,t} = \max(0, \frac{1}{2} \ln \frac{c_{i,t}}{\epsilon_{i,t}})$.
    6. Update data weights $\{w_{i,n}^{(t+1)}\}_{i,n=1}^N$:
           $w_{i,n}^{(t+1)} = w_{i,n}^{(t)} \exp(-y_n \alpha_{i,t} h_t(\mathbf{x}_n))$.
```

To decide the best weak learner h_t shared by all classifiers at iteration t, we minimize the total loss as in (6):

$$h_t = \arg\min_h \sum_{i=1}^{N} \mathcal{L}(f_i + h, S_i) \tag{7}$$

$$= \arg\min_h \sum_{i=1}^{N} \sum_{n=1}^{N} w_{i,n} \exp(-y_n(f_i(\mathbf{x}_n) + h(\mathbf{x}_n))) \tag{8}$$

$$= \arg\min_h \sum_{i=1}^{N} \sum_{n=1}^{N} w_{i,n}^{(t)} \exp(-y_n h(\mathbf{x}_n)) \tag{9}$$

$$= \arg\min_h \sum_{n=1}^{N} \tilde{w}_n^{(t)} \exp(-y_n h(\mathbf{x}_n)) \tag{10}$$

In (9), we have $w_{i,n}^{(t)} = w_{i,n} \exp(-y_n f_i(\mathbf{x}_n))$ from the initialization and step 6 of Algorithm 1. The *cross-task* data weight of $\mathbf{x}_n$ is defined to be its total weight in all tasks,and is denoted by $\tilde{w}_n^{(t)} = \sum_{i=1}^N w_{i,n}^{(t)}$. Thus, (10) implies that the optimal discrete h is the one with the minimal cross-task weighted error, *i.e.*,

$$h_t = \arg\min_h \sum_{n=1}^{N} \tilde{w}_n^{(t)} \cdot 1_{[h(\mathbf{x}_n) \neq y_n]}. \tag{11}$$

Once we have h_t, the remaining work at iteration t is to determine its task-specific ensemble coefficients $\{\alpha_{i,t}\}_{i=1}^{N}$. Analogously, the optimal value of $\alpha_{i,t}$ is set to minimize (6). We observe that $\alpha_{i,t}$ has influence only on $\mathcal{L}(f_i, S_i)$, and is irrelevant to the loss of other tasks. Thus the optimal value of $\alpha_{i,t}$ can be obtained by setting the first derivative of $\mathcal{L}$ with respect to $\alpha_{i,t}$ to zero. It follows that

$$\alpha_{i,t} = \frac{1}{2}\ln\frac{c_{i,t}}{\epsilon_{i,t}}, \text{ where} \tag{12}$$

$$\epsilon_{i,t} = \sum_{n=1}^{N} w_{i,n}^{(t)} \cdot 1_{[h_t(\mathbf{x}_n)\neq y_n]}, \quad c_{i,t} = \sum_{n=1}^{N} w_{i,n}^{(t)} \cdot 1_{[h_t(\mathbf{x}_n)= y_n]}.$$

Note that the resulting $\alpha_{i,t}$ is not guaranteed to be positive. However, we allow only nonnegative combinations of weak learners. Thus $\alpha_{i,t}$ will be set to zero if it is negative. Finally, iteration t is completed by updating the data weights as in step 6 of Algorithm 1.

Novel sample prediction. In the testing phase, given a new sample $\mathbf{z}$, the proposed technique will first find its nearest training sample, say, $\mathbf{x}_i$, and then use local classifier f_i to predict the label of $\mathbf{z}$. Since $\mathbf{z}$ belongs to the neighborhood of $\mathbf{x}_i$, such a tactic is justified by that f_i is optimized to give good classification performances around $\mathbf{x}_i$.

3.3. Useful Properties

The proposed multi-task boosting is simple, easy to implement, and has theoretic merit. At each iteration, it picks the optimal weak learner and computes its ensemble coefficients by directly minimizing the total exponential loss as in (6). Although the selected weak learners are shared cross tasks, it can be readily verified that the exponential loss of each task is monotonically decreased in training. This property guarantees the convergence of our algorithm.

Regarding the gain efficiency of leaning all the local classifiers, we elucidate this point with two aspects of considerations. First, observe that according to (11), no matter how many tasks sample $\mathbf{x}_n$ is associated with, evaluating whether $\mathbf{x}_n$ is misclassified by some weak learner h is performed only once. Second, each selected weak learner is shared cross all the tasks, and the degree of sharing in these tasks is adaptively controlled by the ensemble coefficients. It is in this aspect that the proposed approach has the advantage over other related work, *e.g.*, *JointBoost* [29], where jointly training would quickly become infeasible due to an exponential growth in the number of search hypotheses as the size of training dataset increases.

As is noted before, the algorithm maintains a 2-D weight array $\{w_{i,n}\}$ throughout the boosting iterations. It records not only the difficulties of training data but also those of the tasks. With iterative re-weighting, high weights will be gradually distributed to the difficult data and tasks, and results in more emphases on them. The strategy ensures that all learning tasks will be appropriately addressed.

One final remark about the multi-task boosting is that it iteratively learns classifiers by fusing information from multiple kernels and thus inherently leads to an on-line and incremental way to perform multiple kernel learning (MKL). In addition, compared with conventional optimization techniques like *semi-definite programming* (SDP) [32], it does scale better when the size of data is increased.

4. Experimental results

To evaluate the performances of the proposed method, we carry out experiments of object recognition on two benchmark databases, Caltech-101 [10] and VOC 2007 [9]. Both the two datasets contain images with multiple class labels, and are complicated by large intraclass variations. Therefore, they serve as good test beds for justifying the effectiveness of our approach to local learning.

4.1. Caltech-101

We implement the *one-against-all* rule for our approach to handle multi-class recognition on Caltech-101. In the following, we respectively describe the image data, the adopted descriptors, the experiment setting and the results.

4.1.1 Dataset

The Caltech-101 dataset contains 101 object categories and one additional category of background images. Each category consists of $31 \sim 800$ images. Although the object is positioned near the central region of each image, the large class number and intraclass variations still result in a challenging recognition task. Since the resolutions of images in the dataset are different, we resize them to around $60,000$ pixels before performing feature extraction.

4.1.2 Descriptors and kernels

To enrich the set of weak learners, several image descriptors and their associated distance function are used to extract diverse image features. From (1), the resulting kernels are

- **GB-Dist**: For a given image, the *geometric blur* descriptor [3] is applied to each of 400 randomly sampled edge pixel. Coupling with the distance function suggested in (2) of [34], the kernel is constructed.
- **GB**: The same as GB-Dist, except the geometric distortion term is excluded in evaluating the distance.
- **SIFT-Dist**: The same as GB-Dist, except *SIFT* descriptor [20] is used instead.

	1-NN	SVM	AdaBoost	Ours
GB-Dist	42.4 ± 1.3	61.4 ± 0.8	61.5 ± 0.7	**63.2 ± 0.8**
GB	37.4 ± 0.9	57.6 ± 1.0	58.5 ± 0.7	**59.7 ± 1.2**
SIFT-Dist	49.6 ± 0.8	**58.7 ± 0.9**	57.5 ± 1.0	57.8 ± 0.7
SIFT-SPM	48.8 ± 0.7	**56.1 ± 0.9**	53.3 ± 0.8	55.2 ± 0.7
SS-Dist	31.7 ± 1.4	53.4 ± 1.0	56.1 ± 0.7	**56.3 ± 0.9**
SS-SPM	41.7 ± 0.9	53.9 ± 0.9	55.5 ± 0.7	**56.9 ± 1.3**
C2-SWP	22.0 ± 0.8	**28.3 ± 0.8**	26.0 ± 0.9	26.9 ± 1.0
C2-ML	37.7 ± 0.7	46.3 ± 1.0	44.8 ± 0.9	**46.6 ± 0.6**
PHOG	27.7 ± 1.0	**43.9 ± 0.8**	41.3 ± 1.0	42.9 ± 0.8
GIST	36.8 ± 1.1	48.7 ± 0.8	49.1 ± 1.0	**51.2 ± 0.8**

(a)

	SimpleMKL	AdaBoost (local)	Ours
All	74.3 ± 1.2	74.6 ± 1.3	**75.8 ± 1.1**
	3.26×10^2 sec.	1.87×10^5 sec.	3.92×10^3 sec.

(b)

Table 1. Recognition rates [mean $\pm$ std %] of several approaches on the Caltech-101 dataset. (a) A kernel is taken into account at a time. (b) All kernels are jointly considered.

- **SIFT-SPM**: With a densely sampled grid, SIFT features are extracted and quantized into visual words. The kernel is built by matching *spatial pyramids* [18].
- **SS-SPM / SS-Dist**: The same as SIFT-SPM and SIFT-Dist respectively, except the *self-similarity* descriptor [28] is used to replace the SIFT descriptor.
- **C2-SWP / C2-ML**: We adopt biologically inspired features to depict images. Both the *C2* features suggested by Serre *et al*. [27] and by Mutch and Lowe [23] are respectively used to establish the two RBF kernels.
- **PHOG**: The *PHOG* descriptor [5] is used to summarize the distributions of edge orientations. Together with the χ^2 distance, the kernel is established.
- **GIST**: The images are resized to 128×128 pixels prior to applying the *gist* descriptor [24]. The RBF kernel is then constructed with the L^2 distance.

The distance matrices result from different parameter values of each descriptor are combined in advance. It is used for ensuring that the resulting kernels individually reach their best performances.

4.1.3 Quantitative results

Like in [3, 33, 34], we randomly pick 30 images from each of the 102 categories, and split them into two disjoint subsets: one contains N_{train} images per category, and the other consists of the rest. The two subsets are respectively used for training and testing. The whole evaluation process are repeated 20 times by using different splits between the training and testing subsets. Recognition rates are measured in cases that N_{train} is set to 5, 10, 15, 20, and 25.

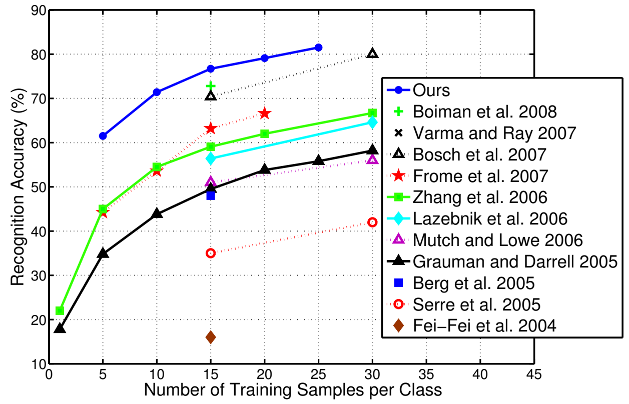

Figure 3. The recognition rates of several approaches on Caltech-101 dataset with different numbers of training data per class.

We first investigate the effectiveness of each kernel by comparing our method with the *one-nearest-neighbor* (1-NN) classifier, SVM, and AdaBoost. When the value of N_{train} is 15, the recognition rates are reported in Table 1a. The 1-NN serves as the baseline for the experiment. By transforming a kernel to a set of dyadic hypercuts (weak learners), AdaBoost and our approach can fuse these hypercuts into a global classifier and a set of local classifiers, respectively. Observe that AdaBoost achieves comparable performance to that of SVM. It implies that the hypercuts can capture useful information in a kernel. Overall, the proposed local learning consistently outperforms AdaBoost, and gives the best recognition rates in most cases.

Focusing now on the case that multiple kernels are considered simultaneously, we evaluate the performances of *SimpleMKL* [25], AdaBoost for local learning, and our method. SimpleMKL is a well-known software for multiple kernel learning, and can learn an optimal ensemble kernel by searching the convex combinations of kernels. With SimpleMKL, a global classifier is obtained with training time 326 seconds, and achieves recognition rate 74.3%. We then compare AdaBoost and our approach in realizing local learning. While AdaBoost carries out local learning for a total of $N = 1530$ classifiers one by one, our method constructs them via a joint training. The recognition rates and computational costs are shown in Table 1b. The results demonstrate that our approach achieves not only a better performance but also gives a two order speed-up.

In Figure 3, the accuracy rates of several recent techniques, including ours, on Caltech-101 are plotted with respect to different numbers of training data. The outcomes of ours are $60.1 \pm 1.1\%$, $70.5 \pm 0.7\%$, $77.5 \pm 0.7\%$, and $79.1 \pm 2.1\%$ when N_{train} is set to 5, 10, 20, and 25 respectively. For the case that N_{train} is 15, the recognition rate 75.8% by our method is either better or comparable to those by other published systems, e.g., [4, 5, 13, 15, 18, 34].

4.2. VOC 2007

The second set of our experiments is performed on the dataset provided by the VOC 2007 classification challenge, which serves as a benchmark for comparing image classification methods in dealing with realistic scenes.

4.2.1 Dataset

The VOC 2007 dataset contains 20 object categories, including indoor objects, vehicles, animals, and people. Each category consists of $195 \sim 4192$ images, and most of them are of similar sizes. Unlike Caltech-101, objects in the images are neither centrally located nor with similar scales. Besides, more than one kind of objects can be present in an image. The overall learning task contains 20 binary classification problems, each of which is to predict the presence/abcense of objects from a certain category.

4.2.2 Descriptors and kernels

We use 18 distance matrices generated from the combinations of six kinds of descriptors and three kinds of spatial pyramids. The image descriptors are

- **SIFT / GB / SS**: We extract the SIFT features from a densely sampled grid over multiple scales, and quantize them into 4000 visual words. The χ^2 distance is adopted as the distance function. Following the same procedure, we build the other two using the geometric blur and self-similarity features.
- **TC-SIFT**: The same as SIFT, except that SIFT features are computed over three normalized RGB channels separately [30].
- **GIST**: All images are resized to resolution 128×128 before performing feature extraction. Then the L^2 distance is used as the distance function.
- **C2-ML**: This is constructed with the C2 feature by Mutch and Lowe [23], and the L^2 distance function.

Three kinds of spatial pyramids, 1×1 (whole image), 2×2 (image quarters) [18], and 1×3 (horizontal bars), are considered to exploit the spatial information. Coupling the descriptors and the spatial pyramids yields 18 distance matrices. However, using all the distance matrices will lead to large memory consumption, and it can be resolved by using only one average distance matrix for each descriptor. To address the possibly large variations due to using different distance measures, the distance matrices are normalized by dividing their respective standard deviation. After generating the distance matrices, the hyper-parameters γ_m in (1) are tuned to form kernels that achieve best performances.

	Train		Train+Val	
	SVM	Ours	SVM	Ours
SIFT	46.3	**47.5**	50.7	**51.4**
TC-SIFT	46.8	**46.9**	51.1	51.1
SS	48.4	**48.8**	**52.4**	52.1
GB	**45.4**	42.8	**47.5**	46.7
Gist	38.6	**39.4**	43.1	**44.3**
C2-ML	34.6	**36.2**	**39.8**	39.6
	SimpleMKL	Ours	SimpleMKL	Ours
All	51.4	**55.2**	57.3	**59.3**

Table 2. Average APs (%) for the VOC 2007 dataset.

4.2.3 Quantitative results

The VOC 2007 dataset provides two training sets: the larger one (Train+Val) contains 5011 images, and the other (Train) has 2501 images. We experiment on both training sets and use the same test set, which consists of 4952 images, for performance evaluation.

For each category, one needs to predict if there exists at least one object of that class in a test image. Each prediction should be a real value to reflect the confidence of object presence. The precision-recall curve is built on the prediction values of all test images, and the performance is measured by the *average precision* (AP), which is proportional to the area under the precision-recall curve. Finally the average of 20 APs is used for the comparison.

In Table 2, we report the results of comparing our method with SVM and SimpleMKL. When training with one individual kernel at a time, our method performs slightly better than SVM on the Train set, and comparably as SVM on the Train+Val set. It shows that the learned local models give the same or better results than a global one does. When learning with multiple kernels, our method achieves average AP 59.3% on the Train+Val set, and yields significant improvements to those learned on each individual kernel. The performance gain in the classification accuracy supports that the use of various distances can complement each other and the concept of local models can better capture the structures of complex data. When SimpleMKL is applied to the same kernels to learn a global model, the resulting average AP is 57.3%.

Table 3 summarizes the per-class results on the Train+Val set of our method, SimpleMKL, the top three (denoted respectively as INRIA, XRCE, and TKK) in VOC Challenge 2007 [9], and van Gemert *et al.* [31], which has produced by far the best results for the dataset. Our method achieves average AP 59.3%, and performs the best in 4 out of 20 categories. The performance by the proposed approach is consistently better than that of SimpleMKL, while it is also comparable with that by INRIA and meanwhile falls slightly behind the average AP 60.5% reported in [31].

	avg.	Aero.	Bicy.	Bird	Boat	Bott.	Bus	Car	Cat	Chair	Cow	Table	Dog	Horse	Moto.	Pers.	Plant	Sheep	Sofa	Train	Tv
INRIA	59.4	77.5	63.6	56.1	**71.9**	**33.1**	60.6	78.0	**58.8**	53.5	42.6	54.9	45.8	77.5	64.0	85.9	36.3	44.7	50.6	79.2	53.2
XRCE	57.5	72.3	57.5	53.2	68.9	28.5	57.5	75.4	50.3	52.2	39.0	46.8	45.3	75.7	58.5	84.0	32.6	39.7	50.9	75.1	49.5
TKK	51.7	71.4	51.7	48.5	63.4	27.3	49.9	70.1	51.2	51.7	32.3	46.3	41.5	72.6	60.2	82.2	31.7	30.1	39.2	71.1	41.0
van Gemert *et al.* [31]	**60.5**	**80.4**	**64.9**	**57.0**	69.1	24.6	**65.8**	**78.2**	54.3	**56.9**	42.4	53.7	**47.0**	**81.5**	65.6	**87.9**	**38.3**	**52.3**	53.9	**83.2**	53.3
SimpleMKL	57.3	74.1	62.7	48.7	66.9	29.1	62.6	75.0	56.9	54.5	42.7	54.8	44.2	76.3	**65.8**	83.6	28.7	42.5	51.5	74.7	50.9
Ours	59.3	76.5	64.6	51.8	68.3	32.2	61.3	77.5	57.8	56.3	**43.5**	**58.8**	44.8	78.4	65.2	85.4	30.4	47.7	**54.6**	76.4	**54.6**

Table 3. Average APs (%) of several approaches on Train+Val set of the VOC 2007 dataset.

5. Conclusion

We have introduced an efficient local learning approach to resolving the difficulties in object recognition caused by large intraclass variations. We cast multiple, independent training processes of local classifiers to a correlative multi-task learning problem, and develop a boosting-based algorithm to accomplish it. The proposed technique is comprehensively evaluated with two benchmark datasets. The recognition rates in both datasets are comparable to those yielded by the respective state-of-the-art approaches. Our method can be considered as a general multi-task learning tool for vision applications where multiple correlative classifiers are required, such as multi-view face detection, multi-part object tracking, or user-dependent media ranking. The framework also provides a new way to carry out multiple kernel learning in an incremental manner.

Acknowledgements. The work is supported in part by NSC grants 95-2221-E-001-031-MY3 and 97-2221-E-001-019-MY3.

References

[1] R. Ando and T. Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *JMLR*, 2005. 2

[2] C. Atkeson, A. Moore, and S. Schaal. Locally weighted learning. *Artificial Intelligence Review*, 1997. 1

[3] A. Berg, T. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondences. In *CVPR*, 2005. 5, 6

[4] O. Boiman, E. Shechtman, and M. Irani. In defense of nearest-neighbor based image classification. In *CVPR*, 2008. 6

[5] A. Bosch, A. Zisserman, and X. Muñoz. Representing shape with a spatial pyramid kernel. In *CIVR*, 2007. 2, 6

[6] R. Caruana. Multitask learning. *ML*, 1997. 2

[7] J. Dai, S. Yan, X. Tang, and J. Kwok. Locally adaptive classification piloted by uncertainty. In *ICML*, 2006. 2

[8] C. Domeniconi and D. Gunopulos. Adaptive nearest neighbor classification using support vector machines. In *NIPS*, 2001. 2

[9] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. 1, 5, 7

[10] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *CVPR Workshop on Generative-Model Based Vision*, 2004. 1, 5

[11] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Computational Learning Theory*, 1995. 4

[12] A. Frome, Y. Singer, and J. Malik. Image retrieval and classification using local distance functions. In *NIPS*, 2006. 1, 2

[13] A. Frome, Y. Singer, F. Sha, and J. Malik. Learning globally-consistent local distance functions for shape-based image retrieval and classification. In *ICCV*, 2007. 1, 2, 6

[14] M. Gönen and E. Alpaydin. Localized multiple kernel learning. In *ICML*, 2008. 1

[15] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *ICCV*, 2005. 6

[16] T.-K. Kim and J. Kittler. Locally linear discriminant analysis for multimodally distributed classes for face recognition with a single model image. *PAMI*, 2005. 2

[17] G. Lanckriet, N. Cristianini, P. Bartlett, L. Ghaoui, and M. Jordan. Learning the kernel matrix with semidefinite programming. *JMLR*, 2004. 2

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

[19] Y.-Y. Lin, T.-L. Liu, and C.-S. Fuh. Local ensemble kernel learning for object category recognition. In *CVPR*, 2007. 1

[20] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 5

[21] T. Malisiewicz and A. Efros. Recognition by association via learning per-exemplar distances. In *CVPR*, 2008. 1, 2

[22] B. Moghaddam and G. Shakhnarovich. Boosted dyadic kernel discriminants. In *NIPS*, 2002. 2, 4

[23] J. Mutch and D. Lowe. Multiclass object recognition with sparse, localized features. In *CVPR*, 2006. 2, 6, 7

[24] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 2001. 6

[25] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet. More efficiency in multiple kernel learning. In *ICML*, 2007. 2, 6

[26] H. Schneiderman and T. Kanade. A statistical method for 3d object detection applied to faces and cars. In *CVPR*, 2000. 1

[27] T. Serre, L. Wolf, and T. Poggio. Object recognition with features inspired by visual cortex. In *CVPR*, 2005. 6

[28] E. Shechtman and M. Irani. Matching local self-similarities across images and videos. In *CVPR*, 2007. 6

[29] A. Torralba, K. Murphy, and W. Freeman. Sharing visual features for multiclass and multiview object detection. *PAMI*, 2007. 2, 5

[30] K. van de Sande, T. Gevers, and C. Snoek. Evaluation of color descriptors for object and scene recognition. In *CVPR*, 2008. 7

[31] J. C. van Gemert, C. J. Veenman, A. W. M. Smeulders, and J. M. Geusebroek. Visual word ambiguity. *PAMI*, 2009. 7, 8

[32] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 1996. 5

[33] M. Varma and D. Ray. Learning the discriminative power-invariance trade-off. In *ICCV*, 2007. 6

[34] H. Zhang, A. Berg, M. Maire, and J. Malik. Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In *CVPR*, 2006. 2, 5, 6