Learning Class-to-Image Distance with Object Matchings

Guang-Tong Zhou, Tian Lan, Weilong Yang*, and Greg Mori
School of Computing Science, Simon Fraser University
Burnaby, BC, Canada
{gza11,tla58,wya16,mori}@cs.sfu.ca

Abstract

We conduct image classification by learning a class-to-image distance function that matches objects. The set of objects in training images for an image class are treated as a collage. When presented with a test image, the best matching between this collage of training image objects and those in the test image is found. We validate the efficacy of the proposed model on the PASCAL 07 and SUN 09 datasets, showing that our model is effective for object classification and scene classification tasks. State-of-the-art image classification results are obtained, and qualitative results demonstrate that objects can be accurately matched.

1. Introduction

We present a method for image classification that matches sets of objects. We aim to classify an input image into classes, such as those containing a specific object (PASCAL VOC [10]) or coming from a certain scene (SUN 09 [6]). Our representation focuses on the set of objects found in an image class. An image class is represented using the set of objects contained in its image instances. We formulate a class-to-image distance for matching to an unseen image that looks for a set of similar objects in similar spatial arrangements to those found in a set of training images. The distance between this collage of objects and a test image is used to classify the test image.

Image classification is a well-studied problem in computer vision. An important question is choosing an appropriate representation for classification. Standard approaches in the vision literature span a gamut of potential answers for this representation question. Purely statistical measures based on local features are common, *e.g*. Lazebnik *et al*. [17]. Direct exemplar matching methods are also well-studied, *e.g*. Berg *et al*. [1]. Detailed reasoning about object segmentation can also assist in image classification [3]. Higher-level semantic reasoning about object context is another important cue for image classification, *e.g*. [19]. The focus of our paper is on object-level representations, though a solution to image recognition likely requires integration of all these sources of information.

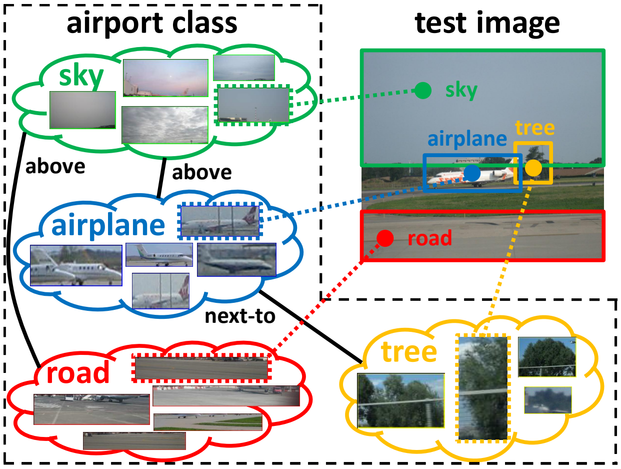

Figure 1. (Best viewed in color.) An example showing the object matchings between the *airport* class and a test image. There are four major object categories in the training *airport* images: “sky”, “airplane”, “road” and “tree”. We match the dashed objects from the training side to the objects in the test image, from which the class-to-image distance is calculated. Spatial relations, *e.g.* “sky-above-airplane”, “sky-above-road”, and “tree-nextto-airplane”, are also considered in measuring the distance.

In this paper we develop a method that matches the objects present in an image. We learn a distance from an image class to a given image that examines a higher-level semantic representation using objects. Figure 1 shows an example of the object matching. We are inspired by two recent lines of work – Object Bank [20], which takes a statistical view of object presence, and exemplar SVM [22] which considers matching individual exemplar objects. The Object Bank work of Li *et al*. [20] showed that a large bank of object detectors is an effective feature for image classification – building a feature vector that captures the statistics of object detector responses. Malisiewicz and Efros [22] advocate for an exemplar matching approach – each image is its own island of uniqueness. Our work bridges these two approaches, leveraging the strength of many objects as a representation

*Weilong Yang (weilongyang@google.com) is now working at Google Inc.

for images, but using a matching framework that considers collages of objects across an entire training class.

Our main contribution is the development of this image classification method. We present a novel latent variable distance function learning framework that considers matchings of objects between a test image and a set of training images from one class. We develop efficient representations for the relationships between objects in this latent variable framework. We show empirically that this method is effective, and that reasoning about objects and their relations in images can lead to high quality classification performance.

2. Related Work

Object-level representations: Image understanding with object-level representations is common in computer vision research. We divide the literature into three categories. First, object-level representations supply rich information to assist detection and recognition. Malisiewicz and Efros [22] learn per-exemplar distance functions for data association based object detection. Li *et al.* [20] tackle scene classification by representing an image as Object Bank – a feature vector that captures the statistics of object detectors. Second, object-level representations can be combined with other information sources. Wang and Mori [31] model object-tag correspondences in a latent variable framework. Wang and Forsyth [29] jointly learn object categories and visual attributes in a multiple instance learning framework. Third, groups of objects can provide useful contextual information. Rabinovich *et al.* [27] exploit contextual relevance of objects by modeling object co-occurrences. Lee and Grauman [18] encode the layout of object-level patterns by object-graph descriptors. Li *et al.* [19] model groups of objects as the basic elements for scene understanding. Lan *et al.* [16] retrieve images for structured object queries, and show that contextually-related objects are helpful even if they are not present in the given queries.

Distance function learning: There has been much work in recent years learning distance functions for image classification. An early representative work by Frome *et al.* [12, 13] builds image-to-image distance on top of local patch-based distances, where each patch is localized by a geometric blur descriptor. Boiman *et al.* [2] compute nearest-neighbor based image-to-class distance based on local SIFT descriptors. Wang *et al.* [33] also measure image-to-class distance by learning Mahalanobis distance metrics. A recent work by Wang *et al.* [32] regularizes class-to-image distance via L1-norm. Wang *et al.* [30] define a class-to-bag distance for multiple instance learning. Our method also learns class-to-image distance, but the key difference is that we focus on object-level representations and explicitly reason about objects and their relations in images. In contrast, existing methods always operate in the space of local descriptor features.

3. The Object Matching Based Distance Model

Our goal is to learn a class-to-image distance function that jointly capture object matchings, the pairwise interactions among objects, as well as the global image appearance. We start with an example (Figure 1) that illustrates calculating the class-to-image distance from the *airport* class to a test image. The airport class is represented as a collage of object sets (*i.e.* “sky”, “airplane”, “road” and “tree”) from training images, arranged in certain spatial layout, such as “sky-above-airplane”. In essence, our distance model matches to a test image with a set of similar objects in similar spatial arrangements from training images.

Our model consists of three components: the **unary object distance**, the **pairwise object distance**, and the **global image appearance distance**. The unary object distance measures the object-level distance from an image class to a test image. In our example, we match one object from each of the four object sets (“sky”, “airplane”, “road” and “tree”) to the test image. We calculate the distance between the matched pair of objects. The unary object distance is a summation over the four distances calculated from the four object matchings. The pairwise object distance measures the distance of spatial arrangements of objects from an image class to a test image. In our example, the matched objects in the test image meet the three popular spatial relations in the training airport images. Thus, we further pull the test image close to the airport scene. Finally, our distance model takes the global image features into account and calculates the global image appearance distance accordingly.

3.1. Model Formulation

We first introduce the notations used in this paper before defining our distance model. We assume the ground-truth object bounding boxes are available in the training images. Note that this assumption is easy to satisfy because object annotation is becoming more and more prevalent with the help of online annotation tools such as LabelMe [28] and Amazon Mechanical Turk. Our two experimental datasets, PASCAL 07 and SUN 09, are both fully annotated.

For an image class C, we gather together all the objects in the training images belonging to this class to make up the object sets $\mathcal{O} = \{\mathcal{O}_i\}_{i \in \mathcal{V}}$, where $\mathcal{V}$ denotes all the object categories in $\mathcal{O}$, and $\mathcal{O}_i$ is the set of objects annotated with category $i \in \mathcal{V}$. We use $\mathcal{O}_i^u$ to represent the u-th object in $\mathcal{O}_i$. Given an image $\mathbf{x}$, our model is a distance function $D_\theta(C, \mathbf{x})$ (here θ are the parameters of this function) that measures the class-to-image distance from C to $\mathbf{x}$ based on object matchings. Ideally, $D_\theta(C, \mathbf{x})$ will have a small value if the image $\mathbf{x}$ belongs to the class C, and a large value if $\mathbf{x}$ comes from a class other than C.

There are two major challenges in defining $D_\theta(C, \mathbf{x})$. First, even though the ground-truth object bounding boxes are readily available in the training images, we do not have annotated objects on the test image set. To resolve this problem, we assume $\mathbf{x}$ is associated with a set of "hypothesized" objects. We model the location/scale configurations of the "hypothesized" objects as latent variables and infer them implicitly in our model. The latent variables are denoted as $\mathcal{H} = \{\mathcal{H}_i\}_{i \in \mathcal{V}}$, where $\mathcal{H}_i$ is the set of "hypothesized" object configurations in category i. We use $\mathcal{H}_i^v$ to denote the v-th configuration in $\mathcal{H}_i$ and the corresponding "hypothesized" object interchangeably. Note that $\mathcal{H}$ is normally smaller than $\mathcal{O}$ in size because $\mathcal{O}$ gathers all the objects in class-C images and $\mathcal{H}$ only includes the objects in the image $\mathbf{x}$.

A second challenge lies in finding the optimal object matchings from $\mathcal{O}$ to $\mathcal{H}$. If we only consider the unary object distance, we can find the optimal object matching separately within each object category by choosing the closest pair over the bipartite matchings between $\mathcal{O}_i$ and $\mathcal{H}_i$. However, we believe that the pairwise spatial relations can also deliver useful information for measuring distance (as shown in Figure 1). Therefore, we need to jointly consider the unary object distance as well as the pairwise interactions. To address the problem, we model the object matchings as a set of latent variables $\mathcal{M} = \{(u_i, v_i)\}_{i \in \mathcal{V}}$, where u_i and v_i are both object indices, and the pair (u_i, v_i) indicates that object $\mathcal{O}_i^{u_i}$ is matched to object $\mathcal{H}_i^{v_i}$ for category i.

Given the class C and the image $\mathbf{x}$, we can find the optimal settings of $\mathcal{H}$ and $\mathcal{M}$ by minimizing the distance over all possible object configurations and all possible object matchings. Then the minimum distance is treated as the class-to-image distance $D_\theta(C, \mathbf{x})$. Formally, we have

$$D_\theta(C, \mathbf{x}) = \min_{\{\mathcal{H}, \mathcal{M}\}} \theta^\top \Phi(\mathcal{O}, \mathcal{H}, \mathcal{M}, \mathbf{x}), \tag{1}$$

where $\theta^\top \Phi(\mathcal{O}, \mathcal{H}, \mathcal{M}, \mathbf{x})$ is a linear function measuring the distance from C to $\mathbf{x}$ accordingly to putative object configurations $\mathcal{H}$ and putative object matchings $\mathcal{M}$. We define

$$\begin{aligned} &\theta^\top \Phi(\mathcal{O}, \mathcal{H}, \mathcal{M}, \mathbf{x}) = \\ &\alpha^\top \psi(\mathcal{O}, \mathcal{H}, \mathcal{M}) + \beta^\top \rho(\mathcal{H}, \mathcal{M}) + \gamma^\top \phi(\mathbf{x}), \end{aligned} \tag{2}$$

where $\theta = \{\alpha, \beta, \gamma\}$ are the model parameters, and $\Phi = \{\psi, \rho, \phi\}$ is the feature vector defined on $(\mathcal{O}, \mathcal{H}, \mathcal{M}, \mathbf{x})$. Next we describe in detail each component in Eq. 2.

Unary object distance $\alpha^\top \psi(\mathcal{O}, \mathcal{H}, \mathcal{M})$: This function measures the unary object distance between $\mathcal{O}$ and $\mathcal{H}$ based on the object matchings $\mathcal{M}$. To compute the distance between a pair of matched objects, we consider five base distance measures calculated from five local object features including color histograms, HoG [7], LBP [24], Texton [21], and location histograms (more details in Section 6). The unary object distance is then calculated as a weighted summation over all base distances. Formally, we parameterize this function as:

$$\alpha^\top \psi(\mathcal{O}, \mathcal{H}, \mathcal{M}) = \sum_{i \in \mathcal{V}} \sum_t \alpha_{it} \cdot \psi_t(\mathcal{O}_i^{u_i}, \mathcal{H}_i^{v_i}), \tag{3}$$

where $\psi_t(\mathcal{O}_i^{u_i}, \mathcal{H}_i^{v_i})$ is a scalar distance between $\mathcal{O}_i^{u_i}$ and $\mathcal{H}_i^{v_i}$ measured by the type-t features. Note that α_{it} is a scalar parameter that weights the t-th distance measure for all the category-i objects – high weights indicate discriminative object categories. Similar to [12, 13, 22], we restrict α_{it} to be non-negative.

Pairwise object distance $\beta^\top \rho(\mathcal{H}, \mathcal{M})$: This function captures the pairwise spatial relations among certain object categories. Here we follow [8] to define four spatial relations including "ontop", "above", "below" and "next-to". Given two object categories (i, j) and the matched objects $(\mathcal{H}_i^{v_i}, \mathcal{H}_j^{v_j})$ in the image $\mathbf{x}$, we define $\rho_k(\mathcal{H}_i^{v_i}, \mathcal{H}_j^{v_j}) = -1$ if the spatial relation between $\mathcal{H}_i^{v_i}$ and $\mathcal{H}_j^{v_j}$ is consistent with a spatial relation k, and $\rho_k(\mathcal{H}_i^{v_i}, \mathcal{H}_j^{v_j}) = 0$ otherwise. The pairwise object distance is parameterized as:

$$\beta^\top \rho(\mathcal{H}, \mathcal{M}) = \sum_{i \in \mathcal{V}} \sum_{j \in \mathcal{V}} \sum_k \beta_{ijk} \cdot \rho_k(\mathcal{H}_i^{v_i}, \mathcal{H}_j^{v_j}), \tag{4}$$

where β_{ijk} is a scalar parameter that weights the spatial relation k between object categories i and j – high weights indicate discriminative spatial relations. We also require β_{ijk} to be non-negative. This function implements the idea that we should pull the image $\mathbf{x}$ close to the class C if the spatial relations between the matched objects in the image $\mathbf{x}$ are discriminative for the class C.

Global image appearance distance $\gamma^\top \phi(\mathbf{x})$: This function models the distance based on the global image features $\phi(\mathbf{x})$. It is parameterized as:

$$\gamma^\top \phi(\mathbf{x}) = \sum_g \gamma_g \cdot \phi_g(\mathbf{x}), \tag{5}$$

where γ_g is a scalar parameter that weights the g-th global feature $\phi_g(\mathbf{x})$. In fact, the choice of $\phi(\mathbf{x})$ is task-dependent and any robust features can be flexibly encoded in the model. In our experiments, we use the bag-of-word features [4] for object classification on PASCAL 07, and the GIST descriptors [25] for scene classification on SUN 09.

4. Inference

During testing, we are given the model parameters $\theta = \{\alpha, \beta, \gamma\}$ as well as a collection of unannotated test images. For each test image $\mathbf{x}$, we need to compute the class-to-image distance $D_\theta(C, \mathbf{x})$. The final decision is made by classifying images with small distances as positive, and images with large distances as negative. Here the key computational issue is to solve the inference problem in Eq. 1.

The inference problem is hard because we need to examine all the possible configurations (*i.e.* locations and scales) for each object category, search over all the possible object matchings, and find the complete configurations and object matchings that jointly minimize the objective function. If we only consider the unary object distance, this results in inferring the optimal object configuration and object matching within each object category independently. We can try each object's configuration in a sliding window manner, and then examine all the possible object matchings. With our full model defined in Eq. 2, the inference problem in Eq. 1 is computationally infeasible.

To resolve this problem, we employ several approximation strategies. First, we reduce the search space of location/scale configurations for the objects in an object category. This is achieved by running an object detector [11] on all locations/scales in $\mathbf{x}$ in a standard sliding window manner, followed by non-maximum suppression to obtain the candidate configurations. In our experiments, we use respectively 5 and 10 candidate configurations for each object category per PASCAL 07 and SUN 09 image. We keep using the notation $\mathcal{H}_i$ to denote the candidate configurations of object category i. When solving the inference problem in Eq. 1, we restrict the selected object for object category i to one of its corresponding candidate configurations in $\mathcal{H}_i$.

The second approximation strategy is for object matchings. Given the candidate configurations $\mathcal{H}_i$, there are $|\mathcal{O}_i| \times |\mathcal{H}_i|$ possible object matchings for the object category i. It is costly to consider all of them, especially since we need to jointly regard all the object categories in finding the optimal set of object matchings. Here we reduce the search space for category i by only considering $|\mathcal{H}_i|$ candidate object matchings. In detail, for each candidate object configuration $\mathcal{H}_i^v \in \mathcal{H}_i$, we compute the distance from all the objects in $\mathcal{O}_i$ to it. We then assign a candidate object matching by pairing $\mathcal{H}_i^v$ to its closest object $\mathcal{O}_i^{u^*}$ in $\mathcal{O}_i$. Formally, we identify the candidate ojbect matching by solving the following optimization problem:

$$u^* = \operatorname*{argmin}_u \sum_t \alpha_{it} \cdot \psi_t(\mathcal{O}_i^u, \mathcal{H}_i^v). \tag{6}$$

Note that the candidate object matchings are still latent (*i.e.* not observed in the original data) because they change with the model parameters α during learning. When solving the inference problem in Eq. 1, we require each object category to select one object matching from the candidate set.

Provided the above approximations, it is easy to show that the inference problem in Eq. 1 is now equivalent to the energy minimization problem [15] in a Markov Random Field (MRF) with $|\mathcal{V}|$ nodes. Each node in the MRF corresponds to an object category. The node i has $|\mathcal{H}_i|$ possible states, where the unary energy for each state is the distance calculated by Eq. 6 for the corresponding candidate object matching. An edge (i, j) in the MRF corresponds to the relation between object categories i and j.

The optimization problem in Eq. 1 is still hard if we have to consider the relation between all pairs of object categories, *i.e.* when the relation between object categories is represented by a complete graph. For further speed-up, we prune the graph into a tree structure by considering only frequent spatial relations in the class-C images. In detail, we first assume that only one spatial relation matters for a given pair of object categories, and we choose it as the most frequent spatial relation. The selected spatial relations are then used to construct an undirected weighted (by frequency) graph. We take the maximum spanning tree of this graph as our pruned tree structure for class C. Putting everything together, we can now solve the inference problem in Eq. 1 efficiently with Belief Propagation.

5. Learning

We now describe how to learn the distance function for the class C. Given a set of positive training images $\{\mathbf{x}_p\}_{p=1}^P$ and a set of negative images $\{\mathbf{x}_n\}_{n=1}^N$ of class C, we would like to train the model parameters $\theta = \{\alpha, \beta, \gamma\}$ that tend to associate a small distance to a new test image $\mathbf{x}$ if $\mathbf{x}$ belongs to class C, and a large distance otherwise. A natural way of learning the model is to adopt the latent SVM formulation [11, 9] as follows:

$$\begin{aligned} &\min_{\{\alpha,\beta,\xi\}\geq 0} \frac{1}{2}||\theta||^2 + \frac{c}{P}\sum_{pn} \xi_{pn} \\ &\text{s.t.} \quad D_\theta(C, \mathbf{x}_n) - D_\theta(C, \mathbf{x}_p) \geq 1 - \xi_{pn}, \forall p, n. \end{aligned} \tag{7}$$

Note that each constraint in Eq. 7 constrains that the class-to-image distance from class C to a negative image $\mathbf{x}_n$ should be larger than the distance to a positive image $\mathbf{x}_p$ by a large margin. ξ_{pn} is a slack variable to allow soft-margin. With the constraints, the learned model can discriminate positive and negative images for the class C.

The constrained optimization problem in Eq. 7 can be equivalently written as an unconstrained problem:

$$\min_{\{\alpha,\beta\}\geq 0} \frac{1}{2}||\theta||^2 + \frac{c}{P}\sum_{pn}(1 + D_\theta(C, \mathbf{x}_p) - D_\theta(C, \mathbf{x}_n)). \tag{8}$$

We use the non-convex bundle optimization (NRBM) in [9] to solve Eq. 8. The key issue is to compute the subgradient $\partial_\theta D_\theta(C, \mathbf{x})$ for a particular θ. Let $(\mathcal{H}^*, \mathcal{M}^*)$ be the optimal solution to the inference problem we have solved in Section 4: $\min_{\{\mathcal{H},\mathcal{M}\}} \theta^\top \Phi(\mathcal{O}, \mathcal{H}, \mathcal{M}, \mathbf{x})$. Then it can be shown that the subgradient can be calculated as $\partial_\theta D_\theta(C, \mathbf{x}) = \Phi(\mathcal{O}, \mathcal{H}^*, \mathcal{M}^*, \mathbf{x})$. Note that to keep α and β non-negative, we project the negative values in α and β to zeros after each iteration of the NRBM learning.

It is also possible to learn our distance model by using the ground-truth object bounding boxes annotated in the training images without inferring the latent "hypothesized" configurations. However, our experiments suggest that it does not perform as well as the learning method defined in Eq. 7. This is because the learning of Eq. 7 simulates the testing process when unannotated test images are provided for distance calculation.

6. Experiments

We evaluate the performance of our method on two image datasets: PASCAL 07 [10] and SUN 09 [6]. We briefly describe our experimental setup before reporting the experimental results in Section 6.1.

PASCAL 07 dataset [10]: The PASCAL Visual Object Challenge provides a standard platform for image classification. We use the PASCAL 07 dataset for a comparison with previous work. This dataset contains 9,963 annotated images, 5,011 for training and 4,052 for testing. There are 20 image classes, each corresponds to an object category, *e.g.* *bus*, *table*, *person*, *etc*. The goal is to predict the presence of an object category in a test image. A typical image has around 3 object instances in 2 object categories. On average, an object category contains 783 object instances in the training image set.

SUN 09 dataset [6]: This dataset consists of 12,000 annotated scene images. Similar to [6], we use 4,367 images for training and 4,317 images for testing. There are 111 object categories each containing at least 5 object instances. We filter out small object instances sized less than 20 by 20 pixels, and finally, we have a training set of 4,356 images and a testing set of 4,305 images. A typical image has around 11 object instances in 5 object categories. On average, there are 417 object instances per object category in the training image set. We perform classification tasks on 58 scene classes each containing at least 10 training and 10 test images[1]. The other small scene classes are only considered as negative data in the experiments.

Note that, as a superset of SUN 09, the SUN dataset [34] also provides a standard benchmark for image classification. However, we choose SUN 09 for two reasons. First, the number of object instances per category in SUN 09 is significantly larger than that in SUN (417 as compared to around 65). Second, our method requires ground-truth object bounding boxes on the training set, but only one tenth of the SUN images are annotated.

Local object features: We select or design several state-of-the-art features that are potentially useful for representing object categories. We build color histograms in RGB space. Our histograms have 11 bins in each channel. HoG descriptors [7] provide excellent performance for object recognition. We resize each object instance to 80×100 pixels (which is the average object size), and extract HoG on a regular grid at steps of 8 pixels. In order to characterize image textures, we further use two powerful texture features: Texton [21] and LBP [24]. We construct a 128 entry texton dictionary by clustering the responses of a filter bank with 8 orientations, 2 scales, and 2 elongations. A 128-dimensional texton histogram is built for each object instance. LBP are computed using 8 sampling points on a circle of radius 1 together with a uniform mapping of 59 patterns. In this way, we produce a 59-dimensional LBP histogram for each object instance. To represent an object's absolute location in an image, we partition the image into 5×5 cells, and compute the area of the object instance in each cell. We normalize all the histograms by l_1 norm, and use the histogram intersection distance (*i.e.* one minus the histogram intersection) to measure the base distance on each feature type.

Global image features: For PASCAL 07, dense SIFT with improved Fisher encoding [26] are shown to outperform the other encoding methods in a fair comparison [4]. We use the implementation of [4] with suggested parameters to extract a 327,680-dimensional feature vector for each image. To improve the learning efficiency, we pre-train 20 SVM classifiers for the 20 image classes based on a kernel calculated from the high-dimensional feature vectors. For an image, the output scores of the 20 SVM classifiers are used to construct a 20-dimensional global appearance feature vector. For SUN 09, we simply extract the 512-dimensional GIST descriptors [25] with filters tuned to 8 orientations at 4 different scales.

Baselines: We design five baselines by considering different components of our *Full* model. The first one is the *Global* model using Eq. 5 only. The second one is our *Unary* model with Eq. 3. The third one is the *Unary+Pair* model that incorporates Eqs. 3 and 4. We further develop two unary models based on Eqs. 5 and 3: *Global+Unary*, where object matchings are infered using Eq. 6; and *Global+Unary-Latent*, where object matchings are fixed by setting $\alpha_{it} = 1$ in Eq. 6. The two unary models are designed to test the efficacy of latent object matchings.

For a fair comparison, we use the same solver for learning all these methods. The learning parameter c in Eq. 7 is selected as the best from the range $\{10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}\}$. We perform one-vs-all classification for each image class. Following the PASCAL challenge criterion, the classification performance on both datasets is measured by average precision (AP) and mean average precision over all classes (mAP).

6.1. Results

PASCAL 07: The classification results on PASCAL 07 are listed in Table 1. We first compare *Full* with several state-of-the-art approaches. Our model has significant performance gains over various methods, including similar approaches that operate on object-level representations and explore contextual information in groups of objects [27], a latent SVM model for region-based classification [35], the winner of PASCAL VOC 2007 using multiple kernel learning on bag-of-word features [23], and the "dense SIFT + Fisher encoding" approach which is shown to outperform the other encoding methods [4]. *Full* is comparable with

[1] We manually extract the scene labels for the SUN 09 images as they are not included in the original release. The scene labels are available on our website.

	plane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	motor	person	plant	sheep	sofa	train	tv	mAP	t-test
Rabinovich *et al.* [27]	63.0	22.0	18.0	28.0	43.0	46.0	62.0	32.0	37.0	19.0	30.0	32.0	12.0	31.0	43.0	33.0	41.0	37.0	29.0	62.0	36.0	4.1E-6
Yakhnenko *et al.* [35]	66.9	43.3	32.4	59.5	16.0	39.2	68.9	38.0	38.5	27.7	27.6	31.7	66.7	45.8	77.0	12.5	28.8	28.5	61.1	35.0	42.3	3.2E-10
OB+SVM	67.4	**79.5**	28.7	49.9	**47.5**	69.4	**88.0**	51.0	8.6	37.2	19.8	8.5	78.2	70.5	41.3	33.9	42.4	45.9	75.2	63.6	50.3	2.4E-3
Marszalek *et al.* [23]	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	59.4	1.5E-3
Chatfield *et al.* [4]	79.0	67.4	51.9	70.9	30.8	72.2	79.9	61.4	56.0	49.6	58.4	44.8	78.8	70.8	85.0	31.7	51.0	56.4	80.2	57.5	61.7	7.1E-4
Harzallah *et al.* [14]	77.2	69.3	**56.2**	66.6	45.5	68.1	83.4	53.6	**58.3**	51.1	62.2	45.2	78.4	69.7	86.1	**52.4**	54.4	54.3	75.8	62.1	63.5	**7.2E-1**
Chen *et al.* [5]	76.7	74.7	53.8	72.1	40.4	71.7	83.6	**66.5**	52.5	**57.5**	62.8	**51.1**	81.4	71.5	**86.5**	36.4	**55.3**	60.6	80.6	57.8	**64.7**	**5.1E-1**
Full	**79.2**	69.9	48.9	**73.2**	36.0	**75.6**	83.7	63.8	55.4	50.0	**64.7**	43.7	**82.9**	**74.2**	86.3	31.5	52.1	**62.2**	**83.6**	**64.4**	64.1	N/A
Global	77.8	64.8	47.9	71.0	27.9	70.3	81.2	61.0	54.3	46.2	59.5	41.9	80.3	70.1	85.4	28.3	45.0	53.5	82.1	54.8	60.2	3.7E-6
Unary	62.6	74.9	17.4	34.2	44.3	68.0	86.8	45.6	46.7	39.4	47.3	22.0	77.0	66.8	90.3	31.0	43.8	43.8	68.9	57.8	53.4	8.7E-4
Unary+Pair	62.9	75.3	17.6	34.4	44.9	66.7	87.6	45.6	46.6	39.4	48.4	22.8	77.1	68.3	90.3	30.2	40.7	44.9	68.4	59.4	53.6	9.7E-4
Global+Unary-Latent	78.9	69.6	48.7	73.0	35.4	75.7	83.7	63.3	55.5	48.9	64.5	43.3	82.9	74.2	86.2	31.6	50.4	62.3	83.7	64.7	63.8	4.1E-2
Global+Unary	78.9	69.7	48.9	72.7	35.8	75.5	83.7	63.3	55.7	49.1	64.7	43.3	82.9	74.2	86.3	31.7	50.7	62.4	83.7	64.5	63.9	**5.5E-2**

Table 1. Classification results (AP and mAP in %) on PASCAL 07. The figures boldfaced are the best performance among *Full* and state-of-the-art methods. Paired t-tests are also conducted on the AP values to examine *Full* agnist all the other methods. We list the returned p-values in the last column, where the boldfaced figures indicate no significance between *Full* and the compared methods under 5% significance level.

	bedroom	skyscraper	street	building	snowy mtn.	kitchen	highway	field	bathroom	livingroom	forest	coast	mountain	office	airport	mAP	t-test
OB+SVM	**41.6**	50.6	59.1	24.5	**55.3**	**46.1**	63.2	40.7	**51.7**	**19.7**	60.0	27.6	9.7	**10.0**	3.8	13.9	1.4E-6
GIST+SVM	24.9	**71.9**	74.7	30.1	43.5	17.8	78.0	39.3	22.0	3.9	**76.4**	17.0	11.6	6.0	**10.8**	14.2	3.6E-4
Full	38.0	67.8	**82.3**	**42.9**	54.8	44.8	**78.9**	**54.4**	50.2	18.9	74.1	**31.5**	**15.7**	7.9	9.1	**19.2**	N/A
Global	26.7	71.8	76.7	29.0	46.5	23.6	73.3	43.7	23.1	4.1	78.4	28.7	20.4	6.5	7.6	15.3	2.9E-3
Unary	30.8	12.0	51.8	23.6	43.1	28.3	66.3	40.2	29.4	11.8	17.9	28.0	17.5	6.4	4.3	11.5	1.3E-5
Unary+Pair	31.4	15.2	53.3	30.8	46.2	34.6	64.5	50.8	34.9	15.2	20.7	31.1	17.3	6.6	4.4	13.0	2.0E-4
Global+Unary-Latent	38.0	65.4	73.4	27.4	50.0	40.3	74.9	47.2	48.0	13.7	69.1	30.6	19.6	7.0	6.6	17.1	2.7E-4
Global+Unary	37.0	64.6	73.6	32.1	47.7	41.1	74.5	47.3	46.4	19.4	70.2	32.3	22.4	7.5	8.1	17.6	3.6E-3

Table 2. Classification results (AP and mAP in %) on SUN 09. We only report AP on the 15 largest scene classes due to space limitations. The mAP results are averaged over all 58 classes. See the caption of Table 1 for more details.

[14] which combines detection and classification into a unified learning framework, and [5] which is a recent top result on PASCAL 07. We also build our own object bank representations for PASCAL 07. For an image, the representation is a 20-dimensional feature vector, where each dimension corresponds to an object category in PASCAL 07, and its value is the maximum response of an object detector. We train linear SVMs based on the object bank features, leading to OB+SVM in Table 1. Our model significantly improves (by 14% mAP) over this method. These results validate the effectiveness of the proposed method.

We compare *Full* with *Global*, *Unary* and *Unary+Pair*. Table 1 shows that, as a simple combination of these models, *Full* significantly outperforms *Global*, *Unary* and *Unary+Pair* by 4% mAP, 10% mAP and 10% mAP, respectively. This demonstrates that the object matchings learned by local object models (*i.e. Unary* and *Unary+Pair*) provide complementary information to the global image features, and our full model can effectively combine these two sources to build stronger classifiers.

Now we consider *Global+Unary-Latent*, *Global+Unary* and *Full* to evaluate the efficacy of latent object matchings. As shown in Table 1, the two latent models (*i.e. Full* and *Global+Unary*) only perform slightly better than the non-latent model *Global+Unary-Latent*, indicating that the latent object matching method does not contribute much to classification, when the latent variables are inferred by either the unary object distance or the combination of unary and pairwise object distance. This is reasonable since the goal of PASCAL 07 classification is to decide the presence of an object category in a given test image. Once the object detector fires on the test image, matching the detected object to a particular object in the class does not significantly affect the overall classification performance. The next dataset, SUN 09, has scenes with multiple objects, for which this ambiguity is more important.

SUN 09: We summarize the classification results on SUN 09 in Table 2. For comparison, we implement two state-of-the-art scene classification methods. The first is *OB+SVM*, which is the exactly same as the one designed for PASCAL 07. The only difference is that here we employ a 111-dimensional object bank representation, where each dimension corresponds to an object category in SUN 09. We also extract 512-dimensional GSIT descriptors [25] and train a linear SVM for each scene class, *i.e. GIST+SVM*. Our *Full* model significantly outperforms the two methods, and is effective for scene classification. It is worth noting that our *Global* model operates on the same GIST features as *GIST+SVM*, but achieves better performance by targeting on distance function learning.

Similar to PASCAL 07, our *Full* model significantly outperforms *Global*, *Unary* and *Unary+Pair*, by 4%, 8% and 6% respectively. This result again validates that we can build a strong *Full* model by taking advantage of both global image appearance and local object matchings.

Now we evaluate the efficacy of latent object matchings. Recall that *Global+Unary-Latent* uses fixed object matchings, *Global+Unary* uses latent object matchings based on the unary object distance, and our *Full* model uses latent object matchings inferred by the combination of unary and pairwise object distance. Although we do not see a big performance leap from *Global+Unary-Latent*

	airport	highway	bedroom	kitchen
Color	airplane, sky, person, truck, streetlight	sky, road, sign, car, tree	bed, wall, curtain, drawer, television	cupboard, stove, cabinet, oven, microwave
HoG	sky, airplane, road, van, door	sky, road, car, sign, tree	wall, bed, floor, curtain, table	wall, stove, cupboard, floor, oven
Texton	tree, door, streetlight, truck, van	sign, car, tree, road, building	bed, drawer, curtain, television, flowers	stove, oven, cabinet, countertop, refrigerator
LBP	door, truck, streetlight, window, van	sign, car, building, bus, fence	drawer, bed, television, flowers, bottle	stove, oven, cabinet, countertop, microwave
Location	tree, truck, van, window, person	sign, car, tree, sky, building	bed, wall, drawer, television, microwave	stove, cupboard, oven, countertop, cabinet
Spatial relations	airplane-below-sky person-ontop-road truck-ontop-road van-ontop-road tree-ontop-sky	tree-ontop-car car-ontop-building car-ontop-fence bus-ontop-car sky-above-road	bottle-nextto-bed television-ontop-wall bed-ontop-wall table-ontop-wall microwave-ontop-floor	cupboard-above-floor stove-ontop-wall wall-above-floor cabinet-ontop-wall refrigerator-ontop-wall

Table 3. We list the five most discriminative object categories (*i.e.* highly weighed by α) with respect to each local object feature on sample scene classes. We also provide the five most discriminative spatial relations (*i.e.* highly weighed by β) among these object categories.

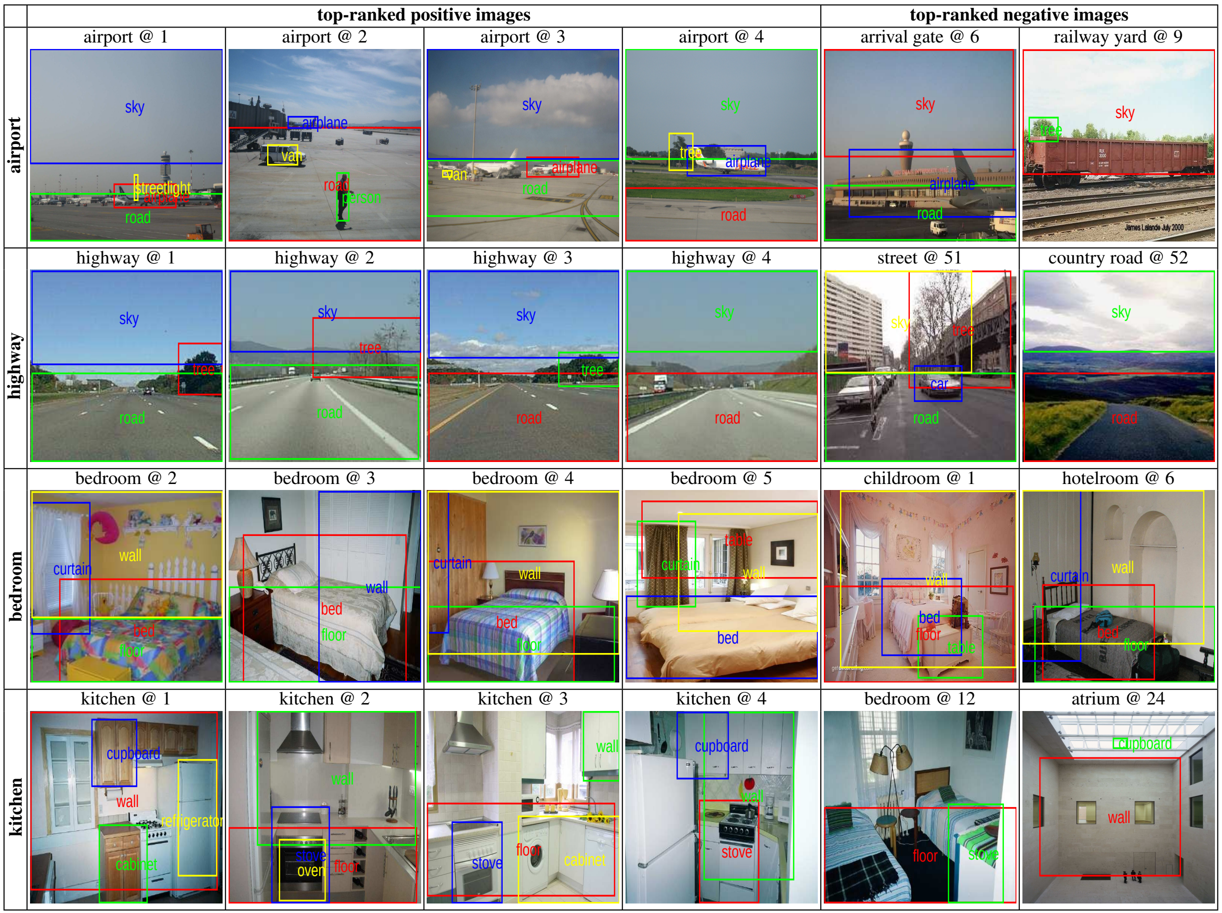

Figure 2. (Best viewed in color.) Sample classification results using our *Full* model. Each row corresponds to a scene class, and we show the top four ranked positive images and the top two ranked negative images. The title of an image includes the scene class label and a figure indicating the rank of the image according to our learned distance: the smaller the rank, the smaller the distance. For an image, we plot up to four discriminative objects (as listed in Table 3) together with the predicted locations. The color of the bounding box shows the relative importance of the objects in distance calculation (sorted by the unary object distance): red > blue > green > yellow.

to *Global+Unary*, our *Full* model does perform significantly better than *Global+Unary-Latent*. This shows the efficacy of our latent object matching method on scene classification. Moreover, *Full* also significantly outperforms *Global+Unary*, by exploiting pairwise spatial relations.

As compared to object classification on PASCAL 07, where the class label is purely determined by one object in the image, scene classification on SUN 09 is more complicated because we need to consider a collection of objects and their correlations to correctly classify a test image. To this end, our model explores object-level representations and various contextual information among objects, and the experimental results show that our model is highly effective.

Visualization: We select four scene classes in SUN 09, and view the learned *Full* model in Table 3. Sample classification results are visualized in Figure 2. Please refer to the captions for more details.

7. Conclusion

We have presented a discriminative model to learn class-to-image distances for image classification by considering the object matchings between a test image and a set of training images from one class. The model integrates three types of complementary distance including the unary object distance, the pairwise object distance and the global image appearance distance. We formulate a latent variable framework and have proposed efficient inference and effective learning methods. Our experiments validates the efficacy of our model in object classification and scene classification tasks. We believe our solution is general enough to be applied in other applications with elementary "object"-level representations, *e.g.* image retrieval with object matchings or video classification/retrieval with action matchings.

References

[1] A. C. Berg, T. L. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondences. In *CVPR*, 2005. 1

[2] O. Boiman, E. Shechtman, and M. Irani. In defense of nearest-neighbor based image classification. In *CVPR*, 2008. 2

[3] Y. Chai, V. S. Lempitsky, and A. Zisserman. BiCoS: A bi-level co-segmentation method for image classification. In *ICCV*, 2011. 1

[4] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In *BMVC*, 2011. 3, 5, 6

[5] Q. Chen, Z. Song, Y. Hua, Z. Huang, and S. Yan. Hierarchical matching with side information for image classification. In *CVPR*, 2012. 6

[6] M. J. Choi, J. J. Lim, A. Torralba, and A. S. Willsky. Exploiting hierarchical context on a large database of object categories. In *CVPR*, 2010. 1, 5

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 3, 5

[8] C. Desai, D. Ramanan, and C. Fowlkes. Discriminative models for multi-class object layout. In *ICCV*, 2009. 3

[9] T. M. T. Do and T. Artières. Large margin training for hidden markov models with partially observed states. In *ICML*, 2009. 4

[10] 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

[11] P. F. Felzenszwalb, D. A. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008. 4

[12] A. Frome, Y. Singer, and J. Malik. Image retrieval and classification using local distance functions. In *NIPS*, 2006. 2, 3

[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. 2, 3

[14] H. Harzallah, F. Jurie, and C. Schmid. Combining efficient object localization and image classification. In *ICCV*, 2009. 6

[15] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *T-PAMI*, 28(10):1568–1583, 2006. 4

[16] T. Lan, W. Yang, Y. Wang, and G. Mori. Image retrieval with structured object queries using latent ranking svm. In *ECCV*, 2012. 2

[17] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006. 1

[18] Y. J. Lee and K. Grauman. Object-graphs for context-aware category discovery. In *CVPR*, 2010. 2

[19] C. Li, D. Parikh, and T. Chen. Automatic discovery of groups of objects for scene understanding. In *CVPR*, 2012. 1, 2

[20] L.-J. Li, H. Su, E. P. Xing, and F.-F. Li. Object bank: A high-level image representation for scene classification & semantic feature sparsification. In *NIPS*, 2010. 1, 2

[21] J. Malik, S. Belongie, T. K. Leung, and J. Shi. Contour and texture analysis for image segmentation. *IJCV*, 43(1):7–27, 2001. 3, 5

[22] T. Malisiewicz and A. A. Efros. Recognition by association via learning per-exemplar distances. In *CVPR*, 2008. 1, 2, 3

[23] M. Marszalek, C. Schmid, H. Harzallah, and J. van de Weijer. Learning object representations for visual object class recognition. In *Visual Recognition Challange*, 2007. 5, 6

[24] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *T-PAMI*, 24(7):971–987, 2002. 3, 5

[25] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 42(3):145–175, 2001. 3, 5, 6

[26] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *ECCV*, 2010. 5

[27] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie. Objects in context. In *ICCV*, 2007. 2, 5, 6

[28] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. LabelMe: A database and web-based tool for image annotation. *IJCV*, 77(1-3):157–173, 2008. 2

[29] G. Wang and D. A. Forsyth. Joint learning of visual attributes, object classes and visual saliency. In *ICCV*, 2009. 2

[30] H. Wang, H. Huang, F. Kamangar, F. Nie, and C. H. Q. Ding. Maximum margin multi-instance learning. In *NIPS*, 2011. 2

[31] Y. Wang and G. Mori. A discriminative latent model of image region and object tag correspondence. In *NIPS*, 2010. 2

[32] Z. Wang, S. Gao, and L.-T. Chia. Learning class-to-image distance via large margin and l1-norm regularization. In *ECCV*, 2012. 2

[33] Z. Wang, Y. Hu, and L.-T. Chia. Image-to-class distance metric learning for image classification. In *ECCV*, 2010. 2

[34] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. SUN database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010. 5

[35] O. Yakhnenko, J. Verbeek, and C. Schmid. Region-based image classification with a latent svm model. Technical report, INRIA, 2011. 5, 6