

# A Tree-Based Context Model for Object Recognition

Myung Jin Choi, *Student Member, IEEE*, Antonio Torralba, *Member, IEEE*, and Alan S. Willsky, *Fellow, IEEE*

**Abstract**—There has been a growing interest in exploiting contextual information in addition to local features to detect and localize multiple object categories in an image. A context model can rule out some unlikely combinations or locations of objects and guide detectors to produce a semantically coherent interpretation of a scene. However, the performance benefit of context models has been limited because most of the previous methods were tested on data sets with only a few object categories, in which most images contain one or two object categories. In this paper, we introduce a new data set with images that contain many instances of different object categories, and propose an efficient model that captures the contextual information among more than a hundred object categories using a tree structure. Our model incorporates global image features, dependencies between object categories, and outputs of local detectors into one probabilistic framework. We demonstrate that our context model improves object recognition performance and provides a coherent interpretation of a scene, which enables a reliable image querying system by multiple object categories. In addition, our model can be applied to scene understanding tasks that local detectors alone cannot solve, such as detecting objects out of context or querying for the most typical and the least typical scenes in a data set.

**Index Terms**—Object recognition, scene analysis, Markov random fields, structural models, image databases.

## 1 INTRODUCTION

IN this work, we use a probabilistic model to capture contextual information of a scene and apply it to object recognition and scene understanding problems. Standard single-object detectors [3], [8] focus on locally identifying a particular object category. In order to detect multiple object categories in an image, we need to run a separate detector for each object category at every spatial location and scale. Since each detector works independently from others, the outcome of these detectors may be semantically incorrect. In order to improve the accuracy of object recognition, we can exploit contextual information such as global features of an image (e.g., it is a street scene) and dependencies among object categories (e.g., a road and cars co-occur often) in addition to local features. An example is illustrated in Fig. 1b in which detector outputs for 107 object categories are shown. With so many categories, many false alarms appear on the image, providing an incoherent scene interpretation. The six most confident detections for the detector outputs, shown in Fig. 1c, are a mixture of indoor and outdoor objects, while the outcome of our

context model, shown in Fig. 1d, puts a lower probability for indoor objects like a desk and a floor.

Even if we have perfect local detectors that correctly identify all object instances in an image, some tasks in scene understanding require an explicit context model and cannot be solved with local detectors alone. One example is finding unexpected objects that are out of their normal context, which requires modeling expected scene configurations. Fig. 1e shows an image in which an object is out of context. These scenes attract a human's attention since they don't occur often in daily settings. Understanding how objects relate to each other is important to answer queries such as *find some funny pictures* or *which objects most typically co-occur with a car?*

Object dependencies in a typical scene can be represented parsimoniously in a hierarchy. For example, it is important to model that outdoor objects (e.g., sky, mountain) and indoor objects (e.g., desk, bed) typically do not co-occur in a scene. However, rather than encoding this negative relationship for all possible pairs of outdoor and indoor objects, it is more efficient to use a tree model in which all outdoor objects are in one subtree, all indoor objects are in another subtree, and the two trees are connected by an edge with a strong negative weight. Similarly, in order to capture the contextual information that kitchen-related objects such as a sink, a refrigerator, and a microwave co-occur often, all kitchen-related objects can be placed in one subtree with strong positive edge weights.

Motivated by such inherent structure among object categories, we model object co-occurrences and spatial relationships using a tree-structured graphical model. We show that even though no explicit hierarchy is imposed during the learning procedure (unlike in [4], [20]), a tree structure learned from a set of fully labeled images

• M.J. Choi is with Two Sigma Investments, 379 West Broadway, New York, NY 10012. E-mail: myungjin@mit.edu.

• A. Torralba is with the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 77 Massachusetts Avenue, 32-D432, Cambridge, MA 02139. E-mail: torralba@csail.mit.edu.

• A.S. Willsky is with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, 77 Massachusetts Avenue, 32-D582, Cambridge, MA, 02139. E-mail: willsky@mit.edu.

Manuscript received 6 Aug. 2010; revised 17 Feb. 2011; accepted 10 May 2011; published online 6 June 2011.

Recommended for acceptance by J. Winn.

For information on obtaining reprints of this article, please send e-mail to: [tpami@computer.org](mailto:tpami@computer.org), and reference IEEECS Log Number TPAMI-2010-08-0607.

Digital Object Identifier no. 10.1109/TPAMI.2011.119.

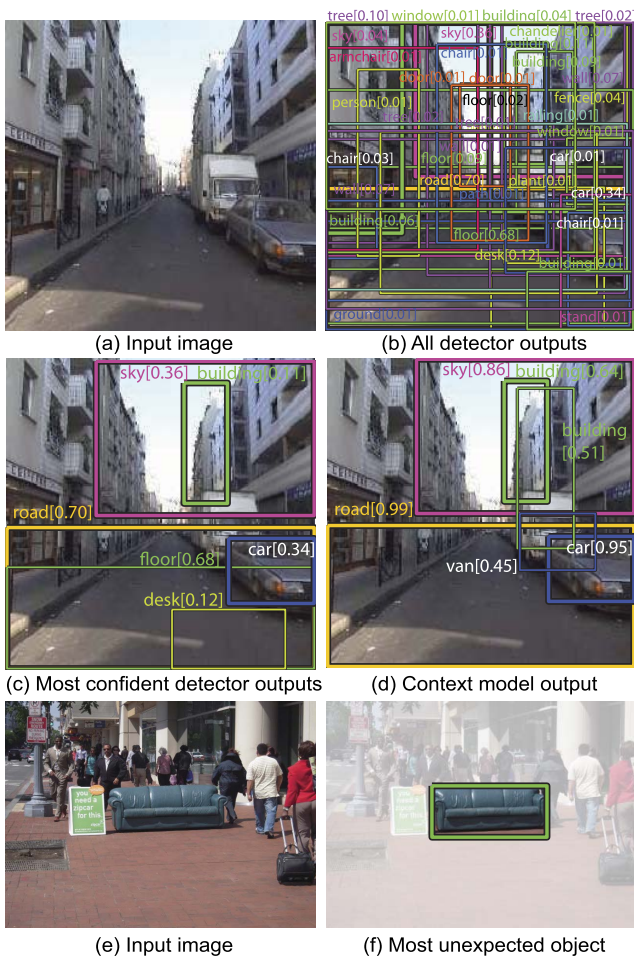


Fig. 1. Detecting objects in and out of context. (a) Input image. (b) Output of 107 class detectors. (c) Six most confident detections using the detector scores. (d) Six most confident detections using our context model. (e) Input image. (f) Most unexpected object in the image.

organizes objects in a natural hierarchy. Enforcing tree-structured dependencies among objects allows us to learn our model for more than a hundred object categories and apply it to images efficiently. We combine this prior model of object relationships with local detector outputs and global image features to detect and localize all instances of multiple object categories in an image.

An important application of object recognition is image interpretation such as querying for images that contain certain object categories. We demonstrate that our context model performs significantly better in querying images with multiple object categories than using only local detectors. We also present the performance of our context model on finding objects/images out of context.

Contextual information is most beneficial when many different object categories are present simultaneously in an image. Current studies that incorporate contextual information for object recognition have been evaluated on standard data sets such as PASCAL 07 [7]. However, those data sets were originally designed to evaluate single-object detectors, and most of the images have no co-occurring instances. We introduce a new data set, SUN 09, with more than 200 object categories in a wide range of scene categories. Each image contains instances of multiple object categories with a wide range of difficulties due to

variations in shape, sizes, and frequencies. As shown in Sections 2 and 6, SUN 09 contains richer contextual information and is more suitable to train and evaluate context models than PASCAL 07.

## 1.1 Related Work

A simple form of contextual information is a co-occurrence frequency of a pair of objects. Rabinovich et al. [24] use local detectors to first assign an object label to each image segment, and then adjust these labels using a conditional random field (CRF). This approach is extended in [10] and [11] to encode spatial relationships between a pair of objects. In [10], spatial relationships are quantized to four prototypical relationships—above, below, inside, and around, whereas in [11], a nonparametric map of spatial priors is learned for each pair of objects. Torralba et al. [27] combine boosting and CRFs to first detect easy objects (e.g., a monitor) and pass the contextual information to detect other more difficult objects (e.g., a keyboard). Tu [29] uses both image patches and their probability maps estimated from classifiers to learn a contextual model, and iteratively refines the classification results by propagating the contextual information. Desai et al. [5] combine individual classifiers by using spatial interactions between object detections in a discriminative manner.

Contextual information may be obtained from coarser, global features as well. Torralba [28] demonstrates that a global image feature called “gist” can predict the presence or absence of objects and their locations without running an object detector. This is extended in [21] to combine patch-based local features and the gist feature. Heitz and Koller [13] combine a sliding window method and unsupervised image region clustering to leverage “stuff” such as the sea, the sky, or a road to improve object detection. A cascaded classification model in [14] links scene categorization, multiclass image segmentation, object detection, and 3D reconstruction.

Hierarchical models can be used to incorporate both local and global image features. He et al. [12] use multiscale conditional random fields to combine local classifiers with regional and global features. Sudderth et al. [26] model the hierarchy of scenes, objects, and parts using hierarchical Dirichlet processes, which encourage scenes to share objects, objects to share parts, and parts to share features. Parikh and Chen [22] learn a hierarchy of objects in an unsupervised manner, under the assumption that each object appears exactly once in all images. Hierarchical models are also common within grammar models for scenes [16], [23], which have been shown to be very flexible to represent complex relationships. Bayesian hierarchical models also provide a powerful mechanism to build generative scene models [18].

In this work, we use a tree-structure graphical model to capture dependencies among object categories. A fully connected model as in [24] is computationally expensive for modeling relationships among many object categories and may overfit with limited number of samples. In the scene-object model [21], objects are assumed to be independent conditioned on the scene type, which may not capture direct dependencies among objects. A tree-structured model provides a richer representation of object dependencies while maintaining a number of connections (i.e., parameters)

that grows linearly with the number of object categories. In addition, it allows efficient integration of different sources of contextual information, which results in improved object detection performances [6]. Note that the tree-structure of our model is learned from the co-occurrence statistics of object categories, and is different from the semantic hierarchies used in [4], [20].

The rest of the paper is organized as follows: In Section 2, we introduce the new SUN 09 data set and compare its statistics with PASCAL 07. In Section 3, we describe our context model that incorporates global image features, object dependencies, and local detector outputs in one probabilistic framework. We use tree-structured dependencies among objects, a framework that admits efficient learning and inference algorithms, described in Sections 4 and 5. We evaluate object recognition and scene understanding performances of our context model in Section 6, and conclude the paper in Section 7.

## 2 THE SUN 09 DATA SET

We introduce a new data set (SUN 09) suitable for leveraging contextual information. The data set contains 12,000 annotated images covering a large number of indoor and outdoor scenes with more than 200 object categories and 152,000 annotated object instances. The images were collected from multiple sources (Google, Flickr, Altavista, LabelMe), and any closeup of an object or images with white backgrounds were removed to keep only images corresponding to scenes in the collection. The annotation procedure was carried out by a single annotator over one year using LabelMe [25]. The labeled images were carefully verified for consistency and synonymous labels were consolidated. The resulting annotations have a higher quality than that by LabelMe or Amazon Mechanical Turk. Therefore, this data set can be used both for training and performance evaluation.

Fig. 2 shows statistics of our data set and compares them with PASCAL 07 [7]. The PASCAL data set provides an excellent framework for evaluating object detection algorithms. However, this data set, as shown in Fig. 2, is not suitable to test context-based object recognition algorithms. The PASCAL data set contains 20 object classes, but more than 50 percent of the images contain only a single object class. MSRC [30] provides more co-occurring objects, but it only contains 23 object classes. The cascaded classification models (DS1) data set [14] is designed for evaluating scene understanding methods, but it has only 14 object classes in outdoor scenes.

Contextual information is most useful when many object categories are present simultaneously in an image, with some object instances that are easy to detect (i.e., large objects) and some instances that are hard to detect (i.e., small objects). The average PASCAL bounding box occupies 20 percent of the image. On the other hand, in our data set, the average object size is 5 percent of the image size, and a typical image contains seven different object categories. Figs. 2c and 2d show typical images from each data set.

## 3 TREE-BASED CONTEXT MODEL

In Section 3.1, we describe a prior model that captures co-occurrence statistics and spatial relationships among objects,

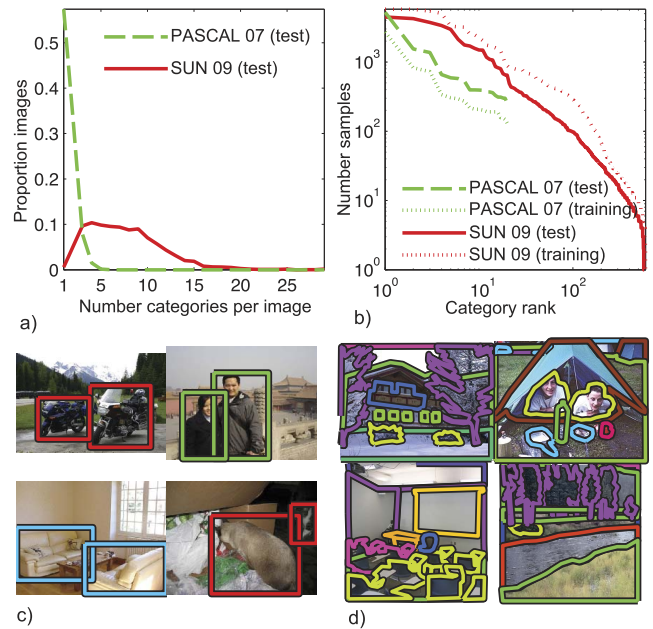


Fig. 2. Comparison of PASCAL 07 and SUN 09. (a) Histogram of the number of object categories in each image. (b) Distribution of train and test samples per object category. (c) Examples of PASCAL images. A typical PASCAL image contains two instances of a single object category, and an object occupies 20 percent of the image. (d) Examples of SUN images. A typical SUN image has seven object categories (with around 14 annotated object instances) with a wide range of sizes (average 5 percent).

and in Section 3.2, we explain how global image features and local detector outputs can be integrated as measurements.

### 3.1 Prior Model

Each object category in our prior model is associated with a binary variable, representing whether the object is present or not in the image, and a Gaussian variable, representing its location.

#### 3.1.1 Co-occurrences Prior

A simple yet effective contextual information is the co-occurrence of object pairs. We encode the co-occurrence statistics using a binary tree model. Each node  $b_i$  in a tree represents whether the corresponding object  $i$  is present or not in an image. The joint probability of all binary variables are factored according to the tree structure

$$p(b) = p(b_{root}) \prod_i p(b_i | b_{pa(i)}), \quad (1)$$

where  $pa(i)$  is the parent of node  $i$ . Throughout the paper, we use a subscript  $i$  to denote a variable (or a vector) corresponding to object  $i$ , and a symbol without a subscript denotes a collection of all corresponding variables:  $b \equiv \{b_i\}$ . A parent-child pair may have either a positive relationship (e.g., a floor and a wall co-occur often) or a negative relationship (e.g., a floor seldom appears with the sky).

#### 3.1.2 Spatial Prior

**Spatial location representation.** Objects often appear at specific relative positions to one another. For example, a computer screen typically appears above a keyboard and a mouse. We capture such spatial relationships by adding location variables to the tree model. Instead of using the



segmentation of an object, we use a bounding box, which is the minimum enclosing box for all the points in the segmentation, to represent the location of an object instance. Let  $\ell_x, \ell_y$  be the horizontal and vertical coordinates of the center of the bounding box, respectively, and  $\ell_w, \ell_h$  be the width and height of the box, respectively. We assume that the image height is normalized to one and that  $\ell_x = 0, \ell_y = 0$  is the center of the image. The expected distance between centers of objects depends on the size of the objects—if a keyboard and a mouse are small, the distance between the centers should be small as well. The constellation model [9] achieves scale invariance by transforming the position information to a scale invariant space. Hoiem et al. [15] relate scale changes to an explicit 3D information. We take the approach in [15] and apply the following coordinate transformations to represent object locations in the 3D-world coordinates:<sup>1</sup>

$$L_x = \frac{\ell_x}{\ell_h} H_i, L_y = \frac{\ell_y}{\ell_h} H_i, L_z = \frac{1}{\ell_h} H_i, \quad (2)$$

where  $L_z$  is the distance between the observer and the object, and  $H_i$  is the physical height of an object  $i$ . The height of each object category could be inferred from the annotated data using the algorithm in [17], but instead, we manually encode real-object sizes (e.g., person = 1.7 m, car = 1.5 m).

**Prior on spatial locations.** The horizontal relative locations of objects vary considerably from one image to another due to different viewpoints, and it has been shown that horizontal locations generally have weak contextual information [28]. Thus, we ignore  $L_x$  and only consider  $L_y$  and  $L_z$  to capture vertical location and scale relationships. We assume that  $L_y$ s and  $L_z$ s are independent, i.e., the vertical location of an object is independent of its distances from the image plane. While we model  $L_y$ s as jointly Gaussian, we model  $L_z$ s using log-normal distributions since they are always positive and are more heavily distributed around small values. We redefine a location variable for object category  $i$  as  $L_i = (L_y, \log L_z)$  and assume that  $L_i$ s are jointly Gaussian. For simplicity, we model spatial relationships among object categories, not between individual instances, so if there are multiple instances of object category  $i$  in an image,  $L_i$  represents the median location of all instances.

We assume that when conditioned on the presence variable  $b$ , the dependency structure of the  $L_i$ s has the same tree structure as the binary tree:

$$p(L|b) = p(L_{root}|b_{root}) \prod_i p(L_i|L_{pa(i)}, b_i, b_{pa(i)}), \quad (3)$$

where each edge potential  $p(L_i|L_{pa(i)}, b_i, b_{pa(i)})$  encodes the distribution of a child location conditioned on its parent location and the presence/absence of both child and parent objects.

1. In order to simplify the coordinate transformations, we make the following assumptions: 1) The image plane is perpendicular to the ground plane, 2) the distance from the observer to the image plane is 1, and 3) all objects have fixed aspect ratios. While these assumptions are unrealistic for recovering the actual 3D-world coordinates, our goal here is to capture relative vertical locations (see the next paragraph) of object categories, so it is not required to get precise coordinate values.

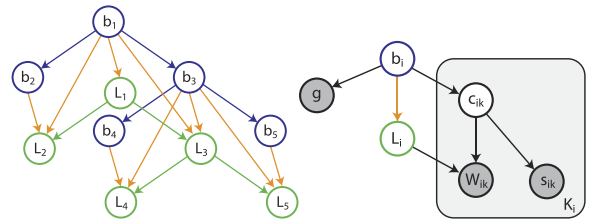


Fig. 3. Graphical model representations for parts of our context model. All nodes are observed during training, and only the shaded nodes are observed during testing. (Left) Prior model relating object presence variables  $b_i$ s and location variables  $L_i$ s. (Right) Measurement model for object  $i$ . The gist descriptor  $g$  represents global image features, and local detector provides candidate window locations  $W_{ik}$  and scores  $S_{ik}$ . The binary variable  $c_{ik}$  indicates whether the window is a correct detection or not.

Fig. 3 shows the graphical model relating the presence variables  $b_i$ s and the location variables  $L_i$ s. Combining (1) and (3), the joint distribution of all binary and Gaussian variables can be represented as follows:

$$p(b, L) = p(b)p(L|b) = p(b_{root})p(L_{root}|b_{root}) \times \prod_i p(b_i|b_{pa(i)})p(L_i|L_{pa(i)}, b_i, b_{pa(i)}). \quad (4)$$

## 3.2 Measurement Model

### 3.2.1 Incorporating Global Image Features

The gist descriptor [28] is a low-dimensional representation of an image, capturing coarse texture and spatial layout of a scene. We introduce the gist as a measurement for each presence variable  $b_i$  to incorporate global image features into our model. This allows the context model to implicitly infer a scene category, which is particularly helpful in predicting whether indoor objects or outdoor objects should be present in the image.

### 3.2.2 Integrating Local Detector Outputs

In order to detect and localize object instances in an image, we first apply off-the-shelf single-object detectors and obtain a set of candidate windows for each object category. Let  $i$  denote an object category and  $k$  index candidate windows generated by baseline detectors. Each detector output provides a score  $s_{ik}$  and a bounding box, to which we apply the coordinate transformation in (2) to get the location variable  $W_{ik} = (L_y, \log L_z)$ . We assign a binary variable  $c_{ik}$  to each window to represent whether it is a correct detection ( $c_{ik} = 1$ ) or a false positive ( $c_{ik} = 0$ ). Fig. 3 shows the measurement model for object  $i$  to integrate gist and baseline detector outputs into our prior model, where we used plate notations to represent  $K_i$  different candidate windows.

If a candidate window is a correct detection of object  $i$  ( $c_{ik} = 1$ ), then its location  $W_{ik}$  is a Gaussian vector with mean  $L_i$ , the location of object  $i$ , and if the window is a false positive ( $c_{ik} = 0$ ),  $W_{ik}$  is independent of  $L_i$  and has a uniform distribution.

## 4 LEARNING

### 4.1 Learning Object Dependency Structure

We learn the dependency structure among objects from a set of fully labeled images. The Chow-Liu algorithm [2] is a

simple and efficient way to learn a tree model that maximizes the likelihood of the data: The algorithm first computes empirical mutual information of all pairs of variables using their sample values. Then, it finds the maximum weight spanning tree with edge weights equal to the mutual information between the variables connected by the edge. We learn the tree structure using the samples of  $b_i$ s in a set of labeled images. Even with more than 100 objects and thousands of training images, a tree model can be learned in a few seconds in Matlab.

Fig. 6 shows a tree structure learned from the SUN 09 data set. Since the output of the Chow-Liu algorithm is an undirected tree, we have selected sky to be the root of the tree to obtain a directed tree structure.<sup>2</sup> Note that we do not use any information regarding the inherent hierarchical structure among object categories during the learning procedure; the Chow-Liu algorithm is simply selecting strong pairwise dependencies. However, the learned tree structure organizes objects in a natural hierarchy. For example, a subtree rooted at building has many objects that appear in street scenes, and the subtree rooted at sink contains objects that commonly appear in a kitchen. Thus, many nonleaf nodes act as if they are representing coarser scale meta-objects or scene categories. In other words, the learned tree structure captures the inherent hierarchy among objects and scenes, resulting in better object recognition and scene understanding performances as demonstrated in Section 6.

## 4.2 Learning Model Parameters

We use the ground-truth labels of training images to learn parameters for the prior model.  $p(b_i|b_{pa(i)})$  can be learned simply by counting the co-occurrences of parent-child object pairs. For each parent-child object pair, we use three different Gaussian distributions for  $p(L_i|L_{pa(i)}, b_i, b_{pa(i)})$ : When both objects are present ( $b_i = 1, b_{pa(i)} = 1$ ), the location of the child object  $L_i$  depends on its parent location  $L_{pa(i)}$ . When the object is present but its parent object is not ( $b_i = 1, b_{pa(i)} = 0$ ), then  $L_i$  is independent of  $L_{pa(i)}$ . When an object is not present ( $b_i = 0$ ), we assume that  $L_i$  is independent of all other object locations and that its mean is equal to the average location object  $i$  across all images.

In the measurement model,  $p(g|b_i)$  can be trained using the gist descriptors computed from each training image. Since the gist is a vector, to avoid overfitting, we use logistic regression to fit  $p(b_i|g)$  for each object category [21], from which we estimate  $p(g|b_i)$  indirectly using  $p(g|b_i) = p(b_i|g)p(g)/p(b_i)$ .

In order to learn the rest of the parameters in the measurement model, we run local detectors for each object category in the training images. The local detector scores are sorted so that  $s_{ik}$  is the  $k$ th highest score for category  $i$ , and  $p(c_{ik}|s_{ik})$  is trained using logistic regression, from which we can compute the likelihoods  $p(s_{ik}|c_{ik}) = p(c_{ik}|s_{ik})p(s_{ik})/p(c_{ik})$ . The probability of correct detection  $p(c_{ik}|b_i)$  is trained using the ground-truth labels and correct detections in the training set.

2. For our location model, the selection of the root node affects the joint probability distribution since the location of a child object is defined relative to its parent object. A rule of thumb for choosing the root node is selecting an object that is large and appears often.

## 5 USING THE MODEL: ALTERNATING INFERENCE ON TREES

Given the gist  $g$ , candidate window locations  $W \equiv \{W_{ik}\}$  and their scores  $s \equiv \{s_{ik}\}$ , we infer the presence of objects  $b \equiv \{b_i\}$ , correct detections  $c \equiv \{c_{ik}\}$ , and the expected locations of all objects  $L \equiv \{L_i\}$  by solving the following optimization problem:

$$\hat{b}, \hat{c}, \hat{L} = \arg \max_{b, c, L} p(b, c, L | g, W, s). \quad (5)$$

Exact inference is complicated since there are both binary and Gaussian variables in the model, so we leverage the tree structures embedded in the model for efficient inference. Specifically, conditioned on  $b$  and  $c$ , the location variables  $L$  form a Gaussian tree. On the other hand, conditioned on  $L$ , the presence variables  $b$  and the detector variables  $c$  together form a binary tree. For each of these trees, efficient inference algorithms are available [1], so we infer  $b$ ,  $c$ , and  $L$  in an alternating manner.

In the first iteration, we ignore the location information  $W$  and compute the MAP estimates of  $b$  and  $c$  conditioned only on the gist of the image  $g$  and the candidate windows scores of detector outputs  $s$ :  $\hat{b}, \hat{c} = \arg \max_{b, c} p(b, c | s, g)$ . Conditioned on these estimates, we infer the expected locations of objects  $\hat{L} = \arg \max_L p(L | \hat{b}, \hat{c}, W)$  using the Gaussian tree. Then, conditioned on the estimates of locations  $\hat{L}$ , we reestimate the presence and detector variables:

$$\begin{aligned} \hat{b}, \hat{c} &= \arg \max_{b, c} p(b, c | s, g, \hat{L}, W) \\ &= \arg \max_{b, c} p(b, c | s, g) p(\hat{L}, W | b, c), \end{aligned}$$

inferring on a binary tree with node and edge potentials modified by the likelihoods  $p(\hat{L}, W | b, c)$ . In this step, we encourage pairs of objects or windows in likely spatial arrangements to be present in the image. In our experiments, we use three iterations alternating between inference in the binary tree and inference in the Gaussian tree.

In the final step, the marginal probability of each presence variable  $p(b_i = 1 | s, g, \hat{L}, W)$  is computed for presence prediction and the marginal probability of each detector variable  $p(c_{ik} = 1 | s, g, \hat{L}, W)$  is computed for object localization. We do not apply a nonmaximum suppression [3] as a postprocessing step, so it is possible that multiple detections of the same object instance reinforce each other through the co-occurrence tree. To avoid this problem, when computing  $p(c_{ik} = 1 | s, g, \hat{L}, W)$  for each detector variable, we cancel out the effect of other detections of the same object. This involves a simple modification of the message from each  $b_i$  to  $c_{ik}$ ,<sup>3</sup> which does not increase the computational complexity of inference. Our inference procedure is efficient even for models with hundreds of objects categories and thousands of candidate windows. For the SUN 09 data set, it takes about 0.5 seconds in Matlab to produce estimates from one image (given the local detector outputs).

3. For example, when computing the message from  $b_i$  to  $c_{i1}$ , the messages from  $c_{i2}, \dots, c_{iK}$  are set to 1.

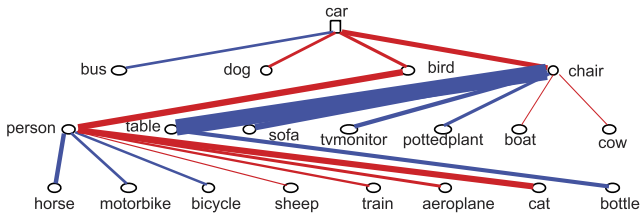


Fig. 4. Object dependency structure learned from PASCAL 07. Red edges correspond to negative correlations between categories. The thickness of each edge represents the strength of the link.

## 6 RESULTS

In this section, we apply the tree-based context model (TreeContext) to object localization (i.e., detecting the correct bounding box), presence predication (i.e., is the object present in the scene?), image query by object categories, and estimation of out-of-context objects.

We use the discriminative part-based models described in [8] as the baseline local detectors, and apply logistic regression to normalize scores across different categories. We also show the performance of the context rescoring method introduced in [8], denoted here as SVM-Context, which trains an SVM for each object category to incorporate contextual information. For each candidate window, a feature vector consists of the score and location of the window, and the maximum scores of all object categories in the image, so for  $M$  object categories, it requires  $M$  different SVMs with an  $(M + 5)$ -dimensional feature vector for each candidate window.

For presence prediction, the most confident detection for each object category is used for the baseline detector. For SVM-context, we extended the approach in [8] by training an SVM for predicting presence of each object category using the maximum scores of all other object categories as feature vectors (which performed much better than simply selecting the most confident detection using the SVMs trained for localization).

### 6.1 Recognition Performance on PASCAL 07

We train the context model for PASCAL 07 using 5,011 images in the training and the validation sets. Fig. 4 shows the dependency structure of 20 object categories learned from the training set. Table 1 provides the average precision (AP)<sup>4</sup> for object localization. Note that the best achievable performance of any context model is limited by the baseline detectors since context models are only used to enhance the scores of the bounding boxes proposed by the baseline detectors. We compare the performance of the tree-based context model with other state-of-the-art methods that also incorporate contextual information [5], [8]. All context models perform better than the baseline detectors, but the performance differences of these methods are relatively small. As discussed in Section 2, the PASCAL 07 data set contains very little contextual information and the performance benefit of incorporating contextual information is small for most of the object categories. We show in the next section that when many object categories are

4. Precision =  $100 \times$  Number of correct detections/Number of detections estimated as correct by the model; Recall =  $100 \cdot$  Number of correct detections/Number of ground-truth object instances; average precision can be computed by varying thresholds. The AP ranges from 0 to 100, and a higher AP indicates better performance.

TABLE 1  
Average Precision for Localization

Category	Baseline	Tree-Context	SVM-Context	Desai et al.		Bound
				Baseline	Context	
aeroplane	28.12	<b>33.13</b>	31.12	27.80	28.80	50.88
bicycle	51.52	50.34	52.22	55.90	<b>56.20</b>	58.76
bird	1.93	1.76	<b>3.26</b>	1.40	3.20	27.45
boat	13.85	<b>15.28</b>	14.87	14.60	14.20	28.14
bottle	23.44	22.64	23.20	25.70	<b>29.40</b>	40.51
bus	38.87	39.08	<b>39.46</b>	38.10	38.70	47.89
car	47.01	45.95	47.26	47.00	<b>48.70</b>	65.95
cat	14.73	<b>18.43</b>	17.69	15.10	12.40	48.60
chair	16.01	<b>17.34</b>	17.04	16.30	16.00	49.08
cow	18.24	<b>19.07</b>	18.08	16.70	17.70	36.89
diningtable	21.01	23.68	20.84	22.80	<b>24.00</b>	30.58
dog	10.73	<b>12.71</b>	11.70	11.10	11.70	46.22
horse	43.22	<b>48.04</b>	46.37	43.80	45.00	69.54
motorbike	40.27	41.64	<b>42.73</b>	37.30	39.40	59.69
person	35.46	36.03	<b>36.21</b>	35.20	35.50	58.92
pottedplant	14.90	15.22	<b>16.48</b>	14.00	15.20	43.75
sheep	19.37	<b>22.48</b>	18.07	16.90	16.10	35.13
sofa	20.56	20.04	<b>23.57</b>	19.30	20.10	42.67
train	37.74	39.48	<b>42.86</b>	31.90	34.20	61.35
tvmonitor	37.00	35.67	<b>38.01</b>	37.30	35.40	54.87
MEAN AP	26.70	27.90	<b>28.05</b>	26.41	27.10	47.84

**Baseline:** baseline detectors without context [8]; **TreeContext:** our context model; **SVM-Context:** context rescoring method from [8]; **Desai et al.:** results from [5] (the baseline in [5] is the same as our baseline, but performances slightly differ); **Bound:** Maximal AP that can be achieved by any context model given the baseline detector outputs.

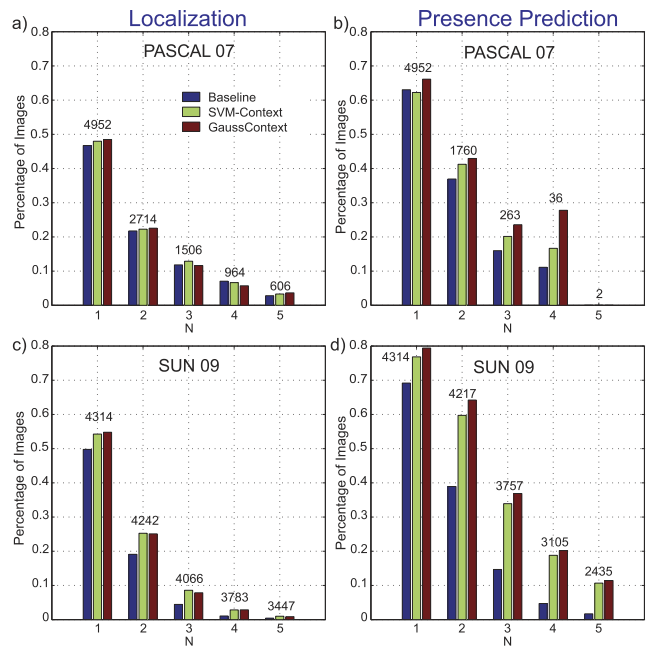


Fig. 5. Object localization and presence prediction performances on PASCAL 07 and SUN 09. (a), (c) Percentage of images in which the top  $N$  most confident detections are all correct. The numbers on top of the bars indicate the number of images that contain at least  $N$  ground-truth object instances. (b), (d) Percentage of images in which the top  $N$  most confident object presence predictions are all correct. The numbers on top of the bars indicate the number of images that contain at least  $N$  different ground-truth object categories.

present simultaneously in an image and when some of them cannot be detected well using only local information, contextual information is crucial in object recognition, and that our tree-based context model does improve the performance significantly in the new data set SUN 09.

Fig. 5a shows the object localization performances based on images. We look at the  $N$  most confident detections in each image and check whether they are all correct. The

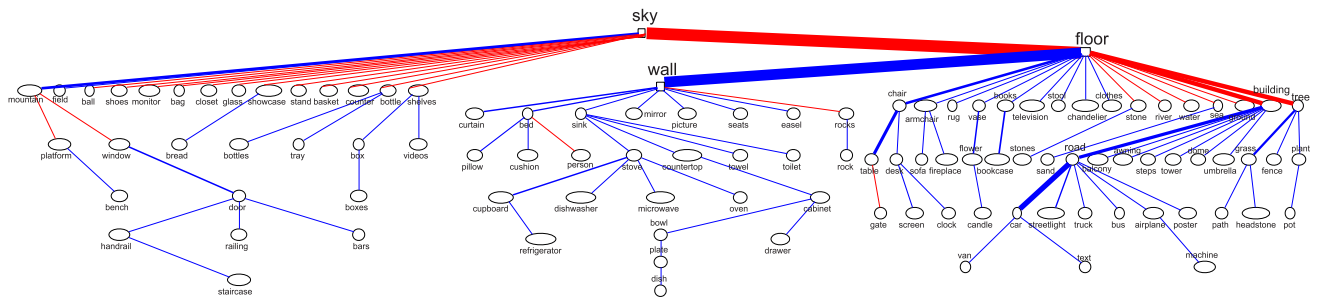


Fig. 6. Object dependency structure learned from SUN 09. Red edges denote negative correlation between categories. The thickness of each edge represents the strength of the link.

numbers on top of the bars indicate the number of images that contain at least  $N$  ground-truth object instances. Fig. 5b shows the performances in presence predication. We compute the probability of each object category being present in the image, and check whether the top  $N$  object categories are all correct. Predicting which objects are present in an image is crucial in understanding its content (e.g., whether it is an indoor or outdoor scene) and can be applied to query images by objects as shown in Section 6.3.1. The numbers on top of the bars indicate the number of images that contain at least  $N$  different ground-truth object categories. Note that the number of images drops significantly as  $N$  gets larger since most images in PASCAL contain only one or two object categories.

## 6.2 Recognition Performance on SUN 09

We divide the SUN 09 data set into the training and the test set so that each set has the same number of images per scene category. The training set has 4,367 images and the test set has 4,317 images. In order to have enough training samples for the baseline detectors [8], we annotated an additional set of 26,000 images using Amazon Mechanical Turk. This set consists of images with a single annotated object, and it was used only for training the baseline detectors and not for learning the context model.

The SUN 09 data set contains over 200 object categories, but the baseline detectors for some objects have poor quality even with additional set of annotations. Since the context model takes baseline detector outputs as measurements and computes the probability of correct detection for each candidate window, it cannot detect an object instance if there is no candidate window produced by the baseline detector. Thus, we remove object categories for which the baseline detector failed to produce at least four correct candidate windows in the entire training set,<sup>5</sup> and use the remaining 107 object categories. These categories span from regions (e.g., road, sky, building) to well-defined objects (e.g., car, sofa, refrigerator, sink, bowl, bed) and highly deformable objects (e.g., river, towel, curtain). The distribution of objects in the test set follows a power law (the number of instances for object  $k$  is roughly  $1/k$ ) as shown in Fig. 2.

### 6.2.1 Context Learned from the Training Set

Fig. 6 shows the dependency structure relating the 107 objects. A notable difference from the tree learned from PASCAL 07 (Fig. 4) is that the proportion of positive correlations is larger. In the tree learned from PASCAL 07,

10 out of 19 edges and 4 out of the top 10 strongest edges have negative relationships. In contrast, 25 out of 106 edges and 7 out of 53 ( $\approx 13$  percent) strongest edges in the SUN 09 tree model have negative relationships. In PASCAL 07, most objects are related by repulsion because most images contain only few categories. In SUN 09, there are many more opportunities to learn positive correlations between objects. Some examples of strong positive relationships are  $p(\text{wall present} \mid \text{floor present}) = 0.94$  and  $p(\text{car not present} \mid \text{road not present}) = 0.98$ . From the learned tree structure, we can see that some objects take the role of dividing the tree according to the scene category as described in Section 4. For instance, *floor* separates indoor and outdoor objects.

Given an image and ground-truth object labels, we can quantify how the labels fit well into our context model by computing the log likelihood of the given labels and their bounding box locations. Fig. 7 shows images in the test set with the highest log likelihood (most typical scenes) and the lowest log likelihood (most unusual scenes). Only objects that are outlined are included in the 107 object categories, and all other objects are ignored. The three most common scenes among the entire test set consist only of floors and walls. The least common scenes have unlikely combinations of labels (e.g., the first image has a label “platform,” which appears in train platform scenes in many of the training images, the second image has a floor, the sea, the sky, and a table all in the same scene, and the last image shows a scene inside a closet). Fig. 7 also shows the most and least common scenes that include grass, a desk, and the sea, respectively. Images with the high likelihood have common object configurations and locations, while images with the low likelihood score have uncommon objects (headstones) or unlikely combinations (sea and table, car and floor).

### 6.2.2 Object Recognition Performance

Figs. 5c and 5d show localization and presence prediction results on SUN 09. We see bigger improvements from incorporating contextual information than on the PASCAL 07 data set for both localization and presence prediction. Note that the tree-based context model improves the presence prediction results significantly: As shown in Fig. 5d, among the 3,757 images that contain at least three different object categories, the three most confident objects are all correct in 38 percent of images (and only 15 percent without context). Fig. 8 shows the improvement in average precision for each object category sorted by the AP improvement over the baseline. Due to the large number of objects in our database, there are

5. Some examples of such categories are attic, candle, chimney, and snow.



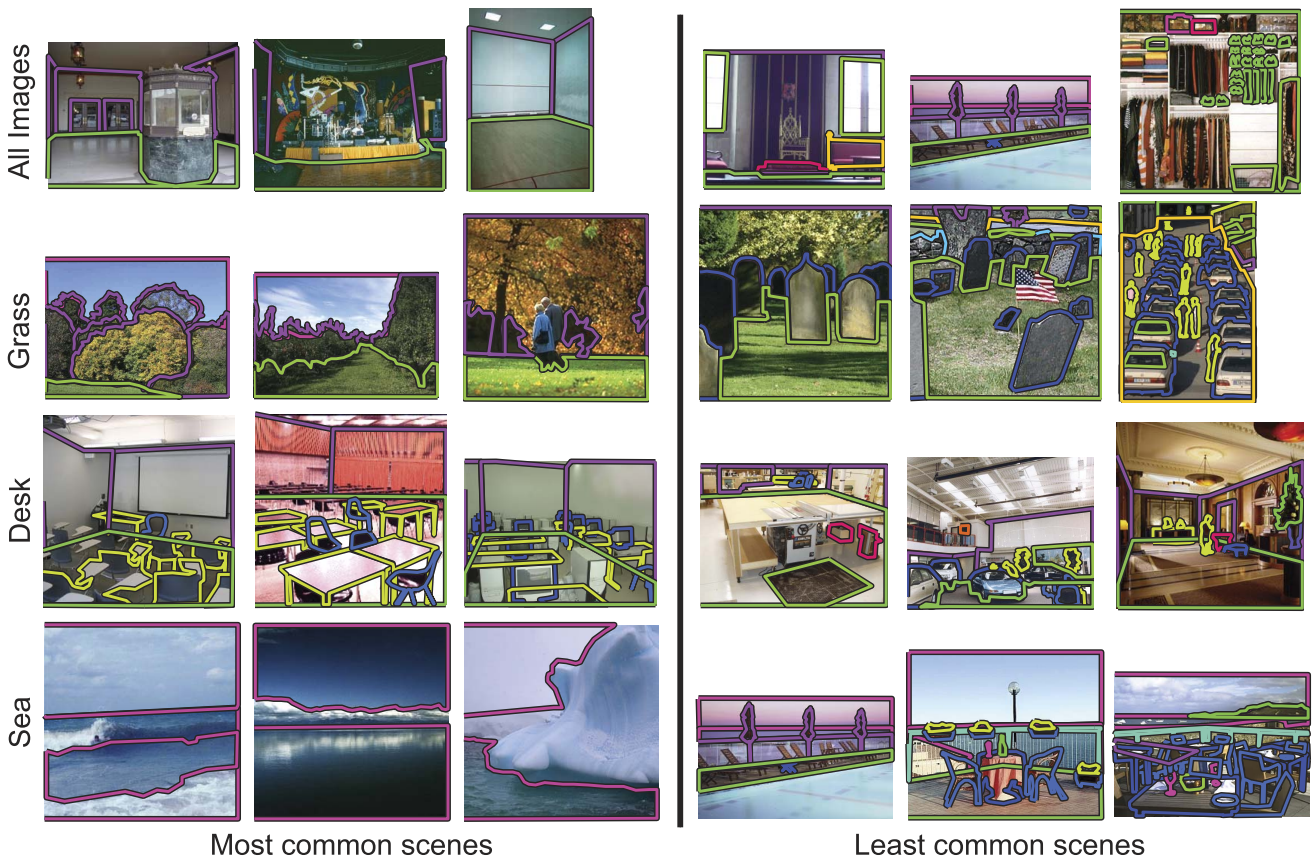


Fig. 7. The most typical scenes and the least typical scenes in the SUN 09 test set estimated using the context model. The first row shows scenes selected from all images, and the remaining rows show scenes that contain grass, desk, and sea, respectively. Only the outlined objects (ground-truth labels) are used to evaluate the likelihood score (e.g., an iceberg is ignored since it is not among the 107 object categories recognized by the model).

many objects that benefit in different degrees from context. Six objects with the largest improvement with TreeContext for object localization are floor (+11.7 over the baseline), refrigerator (+11.0), bed (+8.0), seats(+7.3), road (+6.9), and chandelier (+6.7).

Table 2 shows the mean APs averaged across all object categories for both localization and presence prediction. To investigate the contribution of each component, we show

the mean APs using some parts of the context model as well as the full context model integrating the co-occurrence tree, the location model, and the gist descriptors. Note that using many sources of contextual information results in increased performances and that the global contextual information (i.e., gist) contributes significantly both in localization and presence prediction.

Fig. 9 shows example images with object localization results. For each image, only the six most confident detections are shown. Note that the tree-based context model generally enforces stronger contextual coherency than SVM-Context, which may result in improvement (e.g., removing a truck in a kitchen scene) or may lead to incorrect detections (e.g., hallucinating a truck because of a strong detection of a road in the second image).

### 6.3 Scene Understanding Performance on SUN 09

The SUN 09 data set contains a wide range of scene categories and is suitable for evaluating scene understanding performances. In this section, we show the results of applying our context model for querying images that are most likely to contain certain object categories, and detecting objects in unusual settings.

#### 6.3.1 Querying Images with Multiple Object Categories

A reliable object recognition system enables querying images using objects (e.g., *Search images with a sofa and a table*), rather than relying on captions to guess the content of

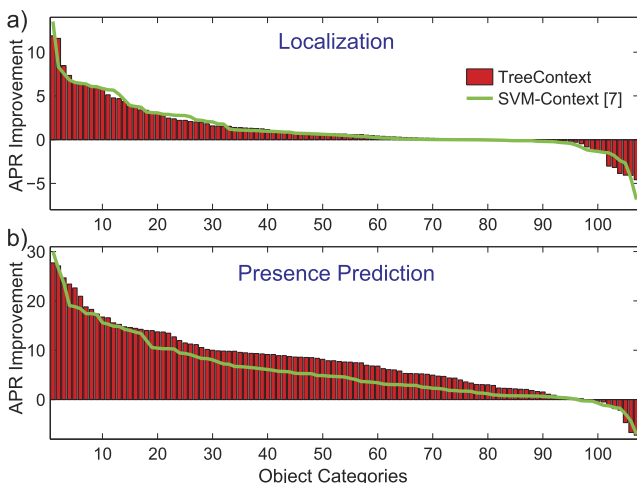


Fig. 8. Improvement of the context models over the baseline detectors. Object categories are sorted by the improvement in AP.





Fig. 9. Examples of scenes showing the six most confident detections using each method.

an image. Our context model performs significantly better than the baseline detectors in predicting whether an object is present or not, as shown in Figs. 5 and 8. Moreover, since the tree-based context model use the detector outputs of all objects as well as the gist descriptor to implicitly infer the scene, it is more reliable in predicting the presence of multiple object categories.

Fig. 10 shows precision-recall curves for image query results using different sets of object categories. We approximated the joint probability of all objects in the set simultaneously present in the image as the product of each object present in the image,<sup>6</sup> and classified a query result as correct only when the image contains all objects in the query set. The tree-based context model shows a clear

6. If the objects in the query set are neighbors in the tree (e.g., bookcase and books), we can compute the joint probability without much additional computation for our context model, but for three or more objects that are far apart in the tree, computing the joint probability can be computationally expensive, even for a tree model. For simplicity, we approximate the joint probability as products of marginal probabilities.

advantage over the baseline detectors, and in four of the five query sets, performs better than SVM-Context as well. Fig. 11 shows examples of top queries. Note that even when the query result of TreeContext is incorrect, the content of the image strongly resembles that of a correct query result.

TABLE 2  
Mean AP (Averaged across All Object Categories)  
for Localization and Presence Prediction on SUN 09

MEAN AP		Localization	Presence Prediction
Baseline		7.06	17.93
Tree Context	Co-occur tree	7.33	19.93
	Co-occur tree + loc	7.76	21.69
	Co-occur tree + gist	8.13	25.28
	<b>Co-occur tree + loc + gist</b>	<b>8.55</b>	<b>26.08</b>
SVM-Context		8.34	23.79

**Baseline:** baseline detectors without contextual information [8]; **Tree-Context:** our context model. We show the results using parts of the context model; **SVM-Context:** Context rescoring method [8].

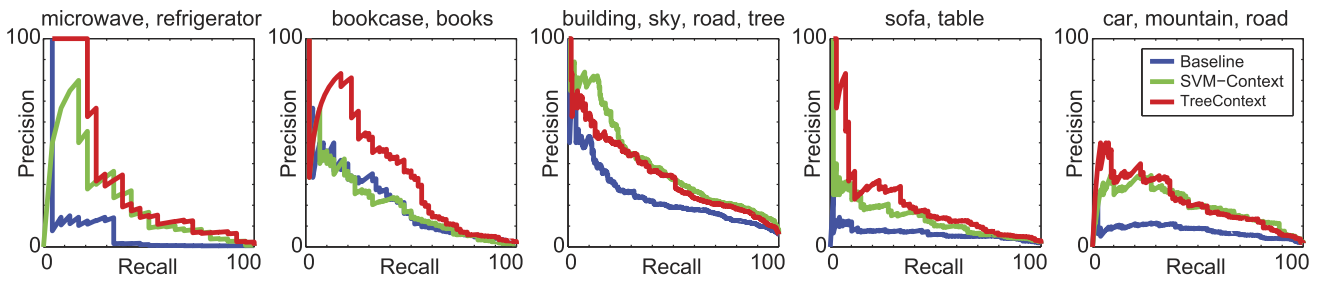


Fig. 10. Precision-recall curves for querying images with a set of object categories.



Fig. 11. Examples of the top seven images retrieved by the baseline detectors [8], context rescoring method with SVMs [8], and our tree-based context model. Correctly retrieved images (images in which all the objects in the query set are present) are shown in blue boxes and incorrect images are shown in red boxes.



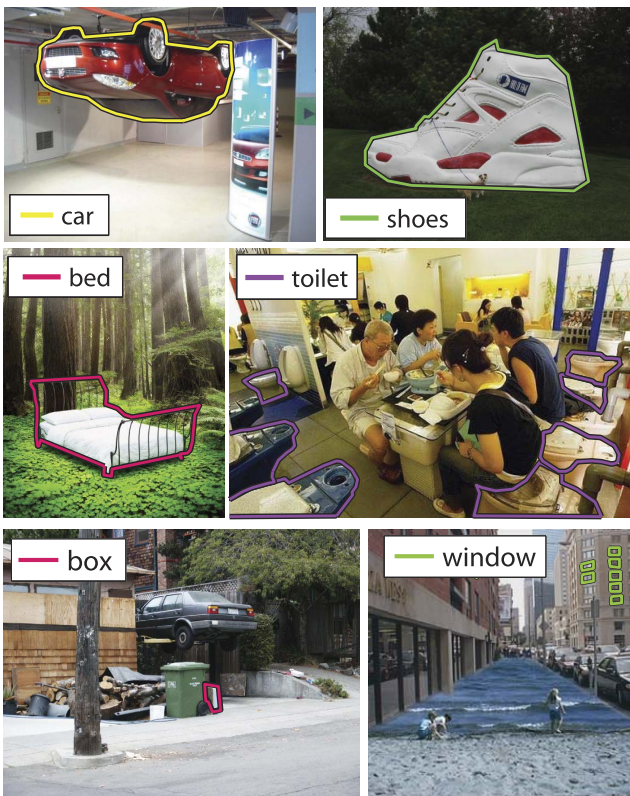


Fig. 12. Six examples of objects out of context (unusual pose, scale, co-occurrence, or position). The highlighted segments show the objects selected by our context model as the most unexpected object category in each image (using ground-truth labels). In the first four images, out-of-context objects are correctly identified, and in the last two images, other objects are selected.

For example, the sixth and the seventh retrieved images for {microwave, refrigerator} using TreeContext are incorrect since they do not contain microwaves, but they are both kitchen scenes, which are semantically much closer to the correctly retrieved images than the results obtained using the baseline detectors or SVM-Context.

### 6.3.2 Finding Objects Out of Context

Fig. 12 shows some images with one or more objects in an unusual setting such as a wrong scale, position, or scene. We have a collection of 26 such images with one or more objects that are out of their normal context. Even if we have perfect local detectors (i.e., ground-truth labels), we still need to use contextual information of a scene to detect whether images or objects are unusual. In this section, we use a variation of our tree-based context model to detect objects out of context from each image.

**Using ground-truth labels.** We first consider the problem of classifying out-of-context objects when the ground-truth object labels and their segmentations are available. Fig. 13 shows a modified version of our original prior model (see Fig. 3) in which we added a set of binary variables  $h_i$ s to indicate whether to use the object dependency or not for object  $i$ . For example,  $p(b_3, L_3 | b_1, L_1, h_3 = 1)$  is defined to have the same probability as in the original prior model, but  $p(b_3, L_3 | b_1, L_1, h_3 = 0)$  is equal to  $p(b_3, L_3)$  regardless of the values of  $b_1$  and  $L_1$ , thus removing the dependencies between objects 1 and 3.

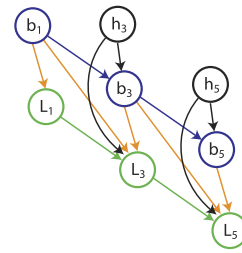


Fig. 13. A modified context model with new binary variables  $h_i$ s to detect objects out of context. If  $h_3 = 0$ , then  $(b_3, L_3)$  become independent of  $(b_1, L_1)$ .

Conditioned on the ground-truth labels ( $b$  and  $L$ ), the context variables  $h_i$ s are independent of each other. In addition, from the tree structure,  $h_i$  only depends on  $b_i, L_i, b_{pa(i)}$ , and  $L_{pa(i)}$ , where  $pa(i)$  is the parent of  $i$ . Thus, if we assume that  $p(h_i = 0) = 0.5$  for all  $i$ ,

$$p(h_i = 0 | b, L) = p(h_i = 0 | b_i, b_{pa(i)}, L_i, L_{pa(i)}) = \frac{1}{1 + C(b_i, b_{pa(i)}, L_i, L_{pa(i)})}, \quad (6)$$

where

$$C(b_i, b_{pa(i)}, L_i, L_{pa(i)}) = \frac{p(b_i | b_{pa(i)}, h_i = 1) p(L_i | L_{pa(i)}, b_i, b_{pa(i)}, h_i = 1)}{p(b_i | b_{pa(i)}, h_i = 0) p(L_i | L_{pa(i)}, b_i, b_{pa(i)}, h_i = 0)} \quad (7)$$

is the *context score* of object  $i$ . The context score measures the likelihood of the labels under the context model relative to an independent model in which all object categories are independent of each other. We can classify an object with the lowest context score (i.e., highest  $p(h_i = 0 | b, L)$ ) as the most unexpected object in the image.

Fig. 14a shows the number of images in the 26-image collection in which at least one out-of-context object was included in the  $N$  most unexpected objects estimated by the context score (i.e.,  $N$  objects with the lowest context score). In 19 out of 26 images, an object with the lowest context score is the correct out-of-context object, which is clearly better than a random guess (assigning random context

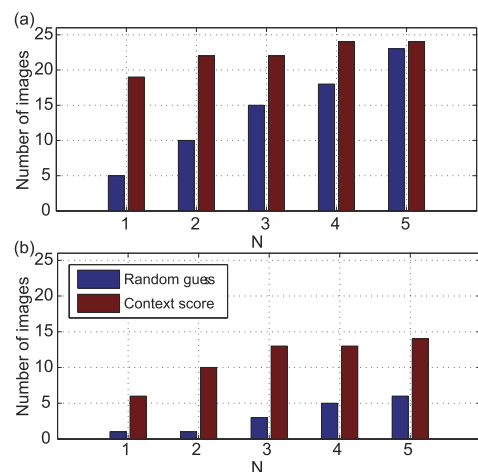


Fig. 14. The number of images in which at least one out-of-context object is included in the set of  $N$  most unexpected objects estimated by our context model. (a) Using ground-truth labels and segmentations. (b) Using local detector outputs.



scores to the objects present in the image). The highlighted segments in Fig. 12 show objects with the lowest context score, which are correct for the first four images and incorrect for the two bottom images. For the bottom left image, the location of the car is not normal, but since the bounding boxes of the car and the road are relatively close to each other, the relative location is not penalized enough in the context score. In the bottom right image, the sand and the sea are out of context, but since quite a few images in the training set have buildings on the beach or cars next to the sea, the unusual combination of objects in the image is not detected by the context model.

**Using detector outputs.** Using local detector outputs to detect objects out of context is a much more challenging task. Objects that are not in their normal settings generally have different appearances or viewpoints from typical training examples, making local detectors perform poorly. Even if a local detector confidently detects an out-of-context object and the context score of the object is low, it is not clear whether the object is present but out of context or the object is not present and the local detector is incorrect.

Given the set of measurements in an image (gist  $g$ , local detector scores  $s$ , and bounding boxes  $W$ ), we would like to estimate the probability of object  $i$  favoring it being independent of its parent object:

$$p(h_i = 0|g, W, s) = \sum_{b_i, b_{pa(i)}} \int p(h_i = 0|b_i, b_{pa(i)}, L_i, L_{pa(i)}) \times p(b_i, b_{pa(i)}, L_i, L_{pa(i)}|g, W, s) dL_i dL_{pa(i)}. \quad (8)$$

In order to simplify the integral, we approximate the joint probability  $p(b_i, b_{pa(i)}, L_i, L_{pa(i)}|g, W, s)$  by assuming that  $i$  and  $pa(i)$  are independent and approximating the Gaussian distribution  $p(L_i|b_i = 1, g, W_i, s_i)$  as a delta function at the mean  $\hat{L}_i$ . Then,

$$p(h_i = 0|g, W, s) \approx \sum_{b_i, b_{pa(i)}} \frac{1}{1 + C(b_i, b_{pa(i)}, \hat{L}_i, \hat{L}_{pa(i)})} \times p(b_i|g, W_i, s_i) p(b_{pa(i)}|g, W_{pa(i)}, s_{pa(i)}), \quad (9)$$

where the context score  $C(b_i, b_{pa(i)}, L_i, L_{pa(i)})$  is defined in (7). In other words, we estimate the label and the location of each object assuming that all objects are independent of each other, and then compute the context score to see whether the resulting configuration fits well with the context model. Note that with the ground-truth labels, we can treat  $p(b_i|g, W_i, s_i)$  and  $p(b_{pa(i)}|g, W_{pa(i)}, s_{pa(i)})$  as delta functions and the above equation reduces to (6).

Fig. 14b shows the result of using local detector outputs to classify objects out of context in each image. Since we do not know the actual objects present in the image, the set of candidates for out-of-context objects is much larger than using the ground-truth labels, so a random guess is incorrect most of the time. In 10 out of 26 images, at least one out-of-context object is correctly identified when we consider two objects with the lowest weighted context score in (9).

## 7 DISCUSSION

We develop an efficient framework to exploit contextual information in object recognition and scene understanding problems by modeling object dependencies, global image features, and local detector outputs using a tree-based graphical model. Our context model enables a parsimonious modeling of object dependencies, and can easily scale to capture the dependencies of over 100 object categories. The SUN 09 data set presented in this paper has richer contextual information than PASCAL 07, and is more suitable for training and evaluating context models. We demonstrate that our context model learned from SUN 09 significantly improves the accuracy of object recognition and image query results, and can be applied to find objects out of context. The SUN 09 data set and the Matlab implementation of our algorithm can be downloaded from <http://people.csail.mit.edu/myungjin/HContext.html>.

We conclude by discussing some possible extensions of the work presented in this paper. Our location model captures spatial relationships of object categories using Gaussian distributions. While this greatly reduces computational complexity, it does not capture some physical relationships such as *a car is supported by a road*. In addition, the location model can be improved by encoding different types of interactions or poses among object instances (e.g., person 1 is riding a horse and person 2 is standing next to it), or spatial relationships based on different viewpoints [19].

The tree structure shown in Fig. 6 captures the inherent hierarchy among object categories. For example, most of the objects that commonly appear in a kitchen are descendants of the node *sink*, and all the vehicles are descendants of *road*. This suggests that a more intuitive structure for object dependencies could be a hierarchy including some metaobjects (such as a desk area) or scenes (kitchen or street) as nodes at coarser scales. Learning a full hierarchical tree structure with such additional nodes may discover important relationships among objects, metaobjects, and scenes, which is an interesting direction for further research.

## ACKNOWLEDGMENTS

The authors would like to thank Taeg Sang Cho for valuable discussions and Joseph J. Lim for his help with experiments. This research was partially funded by Shell International Exploration and Production, Inc., by the US Army Research Office under award W911NF-06-1-0076, by US National Science Foundation (NSF) Career Award (ISI 0747120), and by the US Air Force Office of Scientific Research under Award No. FA9550-06-1-0324. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the US Air Force.

## REFERENCES

- [1] C.M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.
- [2] C.K. Chow and C.N. Liu, "Approximating Discrete Probability Distributions with Dependence Trees," *IEEE Trans. Information Theory*, vol. 14, no. 3, pp. 462-467, May 1968.

- [3] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, 2005.
- [4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2009.
- [5] C. Desai, D. Ramanan, and C. Fowlkes, "Discriminative Models for Multi-Class Object Layout," *Proc. 12th IEEE Int'l Conf. Computer Vision*, 2009.
- [6] S.K. Divvala, D. Hoiem, J.H. Hays, A.A. Efros, and M. Hebert, "An Empirical Study of Context in Object Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2009.
- [7] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results," <http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html>, 2011.
- [8] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object Detection with Discriminatively Trained Part Based Models," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627-1645, Sept. 2010.
- [9] R. Fergus, P. Perona, and A. Zisserman, "Object Class Recognition by Unsupervised Scale-Invariant Learning," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, 2003.
- [10] C. Galleguillos, A. Rabinovich, and S. Belongie, "Object Categorization Using Co-Occurrence, Location and Appearance," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2008.
- [11] S. Gould, J. Rodgers, D. Cohen, G. Elidan, and D. Koller, "Multi-Class Segmentation with Relative Location Prior," *Int'l J. Computer Vision*, vol. 80, pp. 300-316, 2007.
- [12] X. He, R.S. Zemel, and M.Á. Carreira-Perpinán, "Multiscale Conditional Random Fields for Image Labeling," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, 2004.
- [13] G. Heitz and D. Koller, "Learning Spatial Context: Using Stuff to Find Things," *Proc. 10th European Conf. Computer Vision*, 2008.
- [14] G. Heitz, S. Gould, A. Saxena, and D. Koller, "Cascaded Classification Models: Combining Models for Holistic Scene Understanding," *Proc. Neural Information Processing Systems*, 2008.
- [15] D. Hoiem, A. Efros, and M. Hebert, "Putting Objects in Perspective," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, 2006.
- [16] Y. Jin and S. Geman, "Context and Hierarchy in a Probabilistic Image Model," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, 2006.
- [17] J.-F. Lalonde, D. Hoiem, A.A. Efros, C. Rother, J. Winn, and A. Criminisi, "Photo Clip Art," *ACM Trans. Graphics*, vol. 26, no. 3, p. 3, Aug. 2007.
- [18] L.-J. Li, R. Socher, and L. Fei-Fei, "Towards Total Scene Understanding: Classification, Annotation and Segmentation in an Automatic Framework," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2009.
- [19] T. Malisiewicz and A.A. Efros, "Beyond Categories: The Visual Memex Model for Reasoning about Object Relationships," *Proc. Neural Information Processing Systems*, 2009.
- [20] M. Marszałek and C. Schmid, "Semantic Hierarchies for Visual Object Recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2007.
- [21] K.P. Murphy, A. Torralba, and W.T. Freeman, "Using the Forest to See the Trees: A Graphical Model Relating Features, Objects and Scenes," *Proc. Neural Information Processing Systems*, 2003.
- [22] D. Parikh and T. Chen, "Hierarchical Semantics of Objects (hSOs)," *Proc. IEEE Int'l Conf. Computer Vision*, 2007.
- [23] J. Porway, K. Wang, B. Yao, and S.C. Zhu, "A Hierarchical and Contextual Model for Aerial Image Understanding," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2008.
- [24] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie, "Objects in Context," *Proc. IEEE Int'l Conf. Computer Vision*, 2007.
- [25] B.C. Russell, A. Torralba, K.P. Murphy, and W.T. Freeman, "LabelMe: A Database and Web-Based Tool for Image Annotation," *Int'l J. Computer Vision*, vol. 77, pp. 157-173, 2008.
- [26] E.B. Sudderth, A. Torralba, W.T. Freeman, and A.S. Willsky, "Learning Hierarchical Models of Scenes, Objects, and Parts," *Proc. IEEE Int'l Conf. Computer Vision*, 2005.
- [27] A. Torralba, K.P. Murphy, and W.T. Freeman, "Contextual Models for Object Detection Using Boosted Random Fields," *Advances in Neural Information Processing Systems*, MIT Press, 2005.

- [28] A. Torralba, "Contextual Priming for Object Detection," *Int'l J. Computer Vision*, vol. 53, pp. 169-191, 2003.
- [29] Z. Tu, "Auto-Context and Its Application to High-Level Vision Tasks," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2008.
- [30] J. Winn, A. Criminisi, and T. Minka, "Object Categorization by Learned Universal Visual Dictionary," *Proc. IEEE Int'l Conf. Computer Vision*, 2005.



**Myung Jin Choi** received the BS degree in electrical engineering and computer science from Seoul National University, Korea, in 2005, and the SM degree in 2007 and the PhD degree in 2011 in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge. She is a Samsung scholarship recipient. Her research interests include statistical signal processing, graphical models, and computer vision. She is a student member of the IEEE.



**Antonio Torralba** received the degree in telecommunications engineering from Telecom BCN, Spain, in 1994 and the PhD degree in signal, image, and speech processing from the Institut National Polytechnique de Grenoble, Grenoble, France, in 2000. He is an associate professor of electrical engineering and computer science in the Computer Science and Artificial Intelligence Laboratory (CSAIL), Massachusetts Institute of Technology (MIT), Cambridge. From

2000 to 2005, he spent postdoctoral training in the Brain and Cognitive Science Department and the Computer Science and Artificial Intelligence Laboratory at MIT. He is an associate editor of the *IEEE Transactions on Pattern Analysis and Machine Intelligence* and of the *International Journal in Computer Vision*. He received the 2008 US National Science Foundation (NSF) Career award, the best student paper award at the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) in 2009, and the 2010 J.K. Aggarwal Prize from the International Association for Pattern Recognition (IAPR). He is a member of the IEEE.



**Alan S. Willsky** joined the Massachusetts Institute of Technology (MIT) in 1973 and is the Edwin Sibley Webster Professor of Electrical Engineering and director of the Laboratory for Information and Decision Systems. He was a founder of Alphatech, Inc., and chief scientific consultant, a role in which he continues at BAE Systems Advanced Information Technologies. From 1998 to 2002, he served on the US Air Force Scientific Advisory Board. He has received a number of awards, including the 1975 American Automatic Control Council Donald P. Eckman Award, the 1979 ASCE Alfred Noble Prize, the 1980 IEEE Browder J. Thompson Memorial Award, the IEEE Control Systems Society Distinguished Member Award in 1988, the 2004 IEEE Donald G. Fink Prize Paper Award, Doctorat Honoris Causa from Université de Rennes in 2005, and the 2009 Technical Achievement Award from the IEEE Signal Processing Society. In 2010, he was elected to the National Academy of Engineering. He and his students have also received a variety of Best Paper Awards at various conferences and for papers in journals, including the 2001 IEEE Conference on Computer Vision and Pattern Recognition, the 2003 Spring Meeting of the American Geophysical Union, the 2004 Neural Information Processing Symposium, Fusion 2005, and the 2008 award from the journal *Signal Processing* for the outstanding paper in the year 2007. He has delivered numerous keynote addresses and is coauthor of the text *Signals and Systems*. His research interests are in the development and application of advanced methods of estimation, machine learning, and statistical signal and image processing. He is a fellow of the IEEE.

► For more information on this or any other computing topic, please visit our Digital Library at [www.computer.org/publications/dlib](http://www.computer.org/publications/dlib).