Computer Science and Artificial Intelligence Laboratory

Technical Report

MIT-CSAIL-TR-2010-050 October 29, 2010

A Tree-Based Context Model for Object Recognition

Myung Jin Choi, Joseph J. Lim, Antonio Torralba,
and Alan S. Willsky

massachusetts institute of technology, cambridge, ma 02139 usa — www.csail.mit.edu

A Tree-Based Context Model for Object Recognition

Myung Jin Choi, *Student Member, IEEE*, Joseph J. Lim, Antonio Torralba, *Member, IEEE*, 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 datasets with only a few object categories, in which most images contain one or two object categories. In this paper, we introduce a new dataset 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 dataset.

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 [4], [7] 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 detecting unexpected objects that are out of their normal context, which requires modeling expected scene configurations. Fig. 1d-e show 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 sink, refrigerator, and 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 we do not explicitly impose a hierarchical structure in our learning procedure, a tree structure learned from a set of fully labeled images 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 detecting 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 the standard datasets such as PASCAL 07 [6]. However, those datasets were originally designed to evaluate single-object detectors, and most of the images have no co-occurring instances. We introduce a new dataset 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. [21] use local detectors to first assign an object label to each image segment, and then adjusts these labels using a conditional random field (CRF). This approach is extended in [9] and [10] to encode spatial relationships between a pair of objects. In [9], spatial relationships are quantized to four prototypical relationships - above, below, inside and around, whereas in [10], a non-parametric map of spatial priors are learned for each pair of objects. Torralba et al. [25] 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 [27] 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 [26] demonstrates that a global image feature called a "gist" can predict the presence or absence of objects and their locations without running an object detector. This is extended in [18] to combine patch-based local features and the gist feature. Heitz and Koller [12] 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 [13] links scene categorization, multi-class image segmentation, object detection, and 3D reconstruction.

Hierarchical models can be used to incorporate both local and global image features. He et al. [11] use multiscale conditional random fields to combine local classifiers with regional and global features. Sudderth et al. [24] 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 [19] 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], [20], which have been shown to be very flexible to represent complex relationships. Bayesian hierarchical

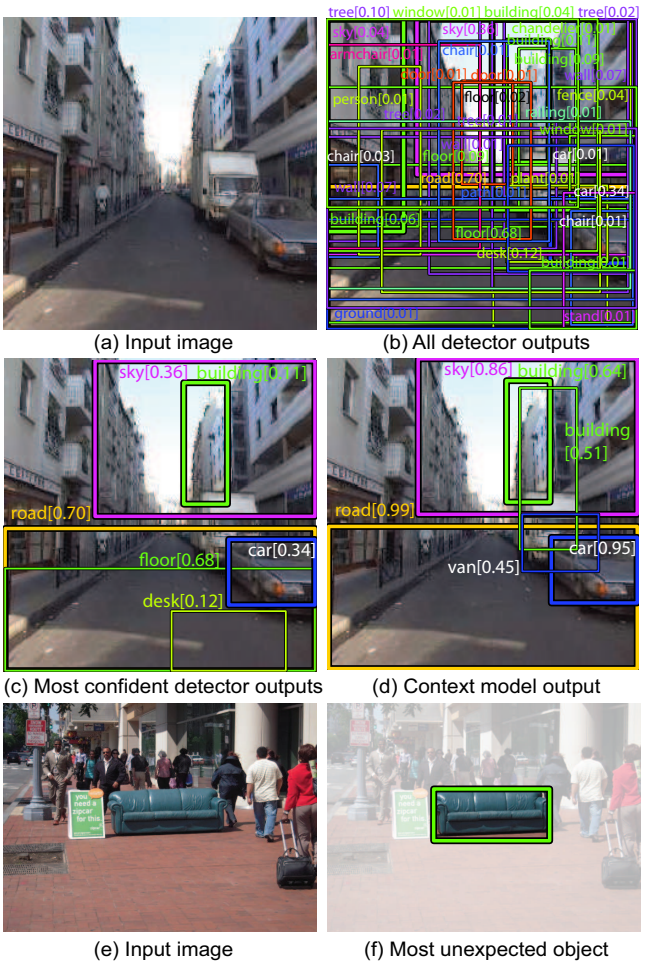

Fig. 1. Detecting objects in and out of context. a) Input image. b) Output of 107 class detectors. c) Six most confident detections using the detector scores. d) Six most confident detections using our context model. e) Input image. f) Most unexpected object in the image.

models also provide a powerful mechanism to build generative scene models [17].

In this work, we use a tree-structure graphical model to capture dependencies among object categories. A fully-connected model as in [21] is computationally expensive for modeling relationships among many object categories and may overfit with limited number of samples. In the scene-object hierarchical model [18], 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.

The rest of the paper is organized as follows: In Section 2, we introduce the new SUN 09 dataset in more detail 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 Dataset

We introduce a new dataset (SUN 09) suitable for leveraging contextual information. The dataset contains 12,000 annotated images covering a large number of scene categories (indoor and outdoors) 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 close-up 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 [23]. 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 dataset can be used both for training and performance evaluation.

Fig. 2 shows statistics of out dataset and compares them with PASCAL 07 [6]. The PASCAL dataset provides an excellent framework for evaluating object detection algorithms. However, this dataset, as shown in Fig. 2, is not suitable to test context-based object recognition algorithms. The PASCAL dataset contains 20 object classes, but more than 50% of the images contain only a single object class. MSRC [28] provides more co-occurring objects but it only contains 23 object classes. The cascaded classification models (DS1) dataset [13] is designed for evaluating scene

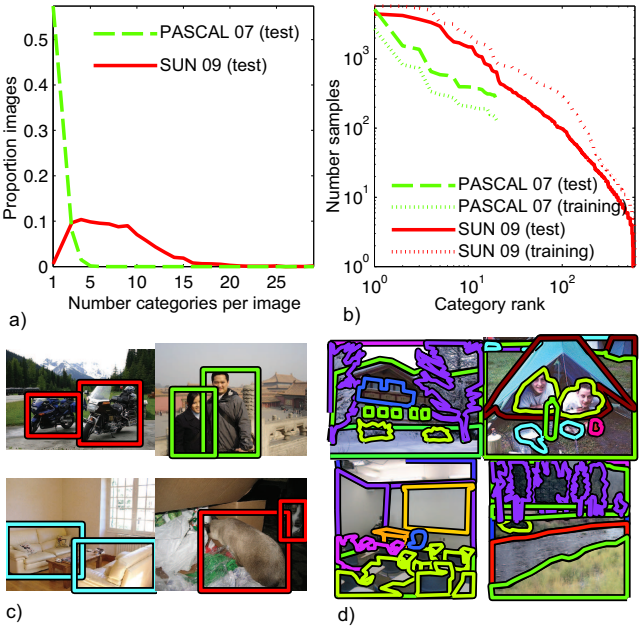

Fig. 2. Comparison of PASCAL 07 and SUN 09. a) Histogram of number of object categories present in each image. b) Distribution of training and test samples per each object category. c) 4 examples of PASCAL images. A typical PASCAL image contains two instances of a single object category, and objects occupy 20% of the image. d) 4 examples of SUN images. A typical SUN image has 7 object categories (with around 14 total annotated object instances) and occupy a wide range of sizes (average 5%).

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% of the image. On the other hand, in our dataset, the average object size is 5% of the image size, and a typical image contains 7 different object categories. Fig. 2c-d show typical images from each dataset.

3 Tree-based Context Model

In Section 3.1, we describe a prior model that captures co-occurrence statistics and spatial relationships among objects, 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 an alphabet 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 ℓ_x, ℓ_y be the horizontal and vertical coordinates of the center of the bounding box, and ℓ_w, ℓ_h be the width and height of the box. 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. Constellation model [8] 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:

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

where f is the distance from observer to the image plane, which we set to 1, and L_z is the distance between the observer and the object. H_i is the physical height of an object i, which could be inferred from the annotated data using the algorithm in [14], but instead, we manually encode real object sizes (e.g., person = 1.7m, car = 1.5m). We assume that all objects have fixed aspect ratios.

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 [26]. 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 from 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. 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.

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:

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

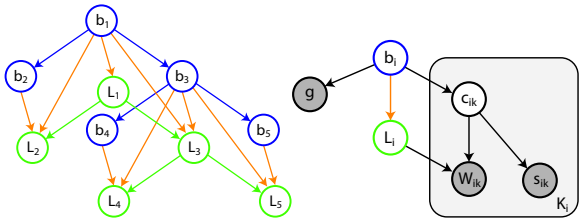

Fig. 3. Graphical model representations for parts of our context model. All nodes are observed during training, and only the shaded nodes are observed during testing. (Left) Prior model relating object presence variables b_i's and location variables L_i's. (Right) Measurement model for object i. The 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.

3.2 Measurement Model

3.2.1 Incorporating Global Image Features

The gist descriptor [26] 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 from 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 [3] 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 dataset. Note that we do not impose that the learned tree have a hierarchical structure. However, by choosing a root node for the learned tree, such hierarchical structure is recovered. For this example, we have selected `sky` to be the root of the tree, and wee see that even though the Chow-Liu algorithm is simply selecting strong pairwise dependencies, our tree 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 non-leaf 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 significant improvements in object recognition and scene understanding tasks 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 child and parent 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 from 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 [18], 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.

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:

$$\hat{b}, \hat{c}, \hat{L} = \underset{b,c,L}{\operatorname{argmax}}\, p(b, c, L|g, W, s) \tag{5}$$

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 forms a Gaussian tree. On the other hand, conditioned on L, the presence variables b and the correct detection variables c together form a binary tree. For each of these trees, there exist efficient inference algorithms [1]. Therefore, we infer b, c and L in an alternating manner.

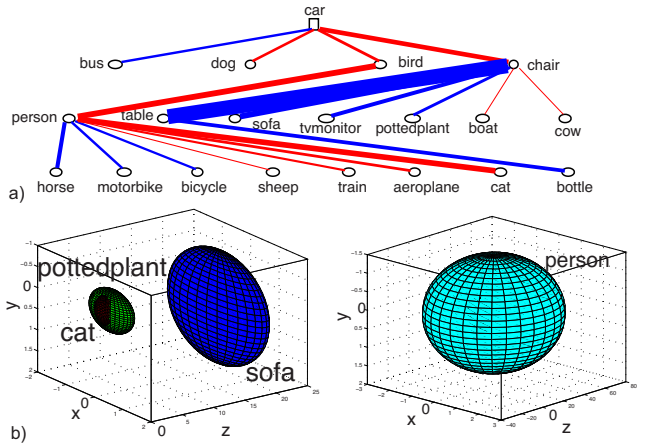

Fig. 4. a) Object dependency structure learned from PASCAL 07. Red edges correspond to negative correlations between categories. The thickness of each edge represents the strength of the link. b) 3D samples generated from the context model. The ellipsoids represents one standard deviation from the means.

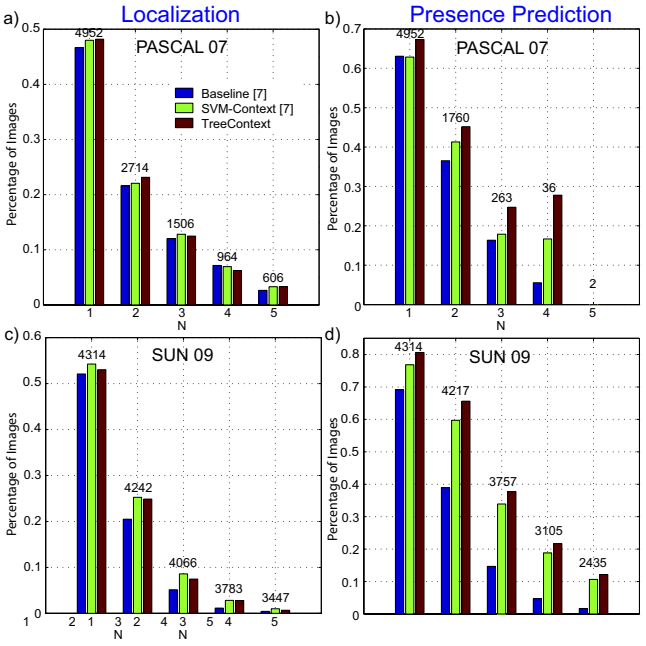

Fig. 5. Image annotation results for PASCAL 07 and SUN 09. a-b) Percentage of images in which the top N most confident detections are all correct. The numbers on top of the bars indicate the number of images that contain at least N ground-truth object instances. c-d) Percentage of images in which the 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.

In our first iteration, we ignore the location information W, and sample[1] b and c conditioned only on the gist g and the candidate windows scores s: $\hat{b}, \hat{c} \sim p(b, c|s, g)$. Conditioned on these samples, we infer the expected locations of objects $\hat{L} = \text{argmax}_L\, p(L|\hat{b}, \hat{c}, W)$ using belief propagation on the resulting Gaussian tree. Then conditioned on the estimates of locations $\hat{L}$, we re-sample b and c conditioned also on the window locations: $\hat{b}, \hat{c} \sim p(b, c|s, g, \hat{L}, W)$, which is equivalent to sampling from 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.

We iterate between sampling on the binary tree and inference on the Gaussian tree, and select samples with the highest likelihood. We use 4 different starting samples of b_i's each with 3 iterations in our experiments. Our inference procedure is efficient even for models with hundreds of objects categories and thousands of candidate windows. For the SUN 09 dataset, it takes about 0.5 second in MATLAB to produce estimates from one image.

6 Results

6.1 Recognition Performance on PASCAL 07

Context learned from the training set We train the context model for PASCAL 07 using 2,501 images in the training set. Fig. 4a shows the dependency structure of 20 object categories learned from the training set, and Fig. 4b shows a few samples generated from the prior model. Since a majority of training images contain a single object category, the context model favors to have one or few objects in each image, so there is limited co-occurrence or spatial contextual information that can be exploited.

Object recognition performance Fig. 5a shows the performance in object localization (i.e., detecting the correct bounding box). We look at the N most confident detections in each image and check whether they are all correct. The numbers on top of the bars indicate the number of images that contain at least N ground-truth object instances. We use the discriminative part-based models described in [7] as the baseline local detectors. In order to normalize scores across different categories, we use a logistic regression to compute the probability of correct detection based on the detector score. For our tree-based context model, we compute the probability of correct detection given gist and detector outputs (i.e. $p(c_{ik} = 1|s, g, W)$) using the efficient inference algorithm described in Section 5. We also show the performance of the context rescoring method introduced in [7], which we denote here as SVM-Context. They train 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 other object categories in the

1. We can also compute the MAP estimates of these binary variables efficiently, but starting from the MAP estimates and iterating between the binary and Gaussian trees typically leads to a local maximum that is close to the initial MAP estimates.

image. Thus, for M object categories, it requires M different SVMs with an (M+5)-dimensional feature vector for each candidate window.

Fig. 5b shows the performances of different methods in presence predication (i.e., is the object present in the scene?). 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. The most confident detection for each object category is used for the baseline detector, and $p(b_i = 1|s, g, W)$ is used for the tree-based context model. For SVM-context, we extended the approach in [7] 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).

Table 1 provides the average precision-recall (APR)[2] 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 incorporates contextual information [5], [7]. 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 dataset 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 with a wide range of difficulties are present simultaneously in an image, contextual information is crucial in object recognition, and that our tree-based context model does improve the performance significantly in the new dataset SUN 09.

Category	Baseline	Gist	Tree Context	SVM-Context	[5] Baseline	[5] Context	Bound
aeroplane	28.12	31.30	**32.05**	30.46	27.80	28.80	50.88
bicycle	51.52	50.79	50.56	51.93	55.90	**56.20**	58.76
bird	1.93	0.75	0.89	**5.14**	1.40	3.20	27.45
boat	13.85	**15.06**	14.90	15.02	14.60	14.20	28.14
bottle	23.44	25.58	25.28	24.05	25.70	**29.40**	40.51
bus	38.87	35.83	36.98	**39.40**	38.10	38.70	47.89
car	47.01	46.74	46.74	46.86	47.00	**48.70**	65.95
cat	14.73	16.72	**18.93**	17.17	15.10	12.40	48.60
chair	16.01	17.91	**18.12**	16.90	16.30	16.00	49.08
cow	18.24	18.07	18.22	**18.60**	16.70	17.70	36.89
diningtable	21.01	23.18	22.93	20.91	22.80	**24.00**	30.58
dog	10.73	11.26	**12.43**	11.60	11.10	11.70	46.22
horse	43.22	45.32	**47.29**	46.51	43.80	45.00	69.54
motorbike	40.27	40.99	41.87	**42.39**	37.30	39.40	59.69
person	35.46	34.77	35.46	**36.34**	35.20	35.50	58.92
pottedplant	14.90	**16.55**	15.67	16.11	14.00	15.20	43.75
sheep	19.37	21.77	**21.81**	18.74	16.90	16.10	35.13
sofa	20.56	19.43	20.40	**23.40**	19.30	20.10	42.67
train	37.74	37.43	38.80	**41.53**	31.90	34.20	61.35
tvmonitor	37.00	34.27	35.75	**37.85**	37.30	35.40	54.87
AVERAGE	26.70	27.19	27.75	**28.05**	26.41	27.10	47.84

TABLE 1
Average precision-recall for localization. Baseline) baseline detector without contextual information [7]; Gist) baseline and gist [22]; TreeContext) our context model; SVM-Context [7]) Context rescoring method from [7]; [5]) results from [5] (the baseline in [5] is the same as our baseline, but performances slightly differ); Bound) Maximal APR that can be achieved by any context model given the baseline detectors.

6.2 Recognition Performance on SUN 09

We divide the SUN 09 dataset 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 [7], 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 dataset 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 4 correct candidate windows in the entire training set, 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.

Context learned from the training set Fig. 6 shows the dependency structure relating the 107 ob-

2. Precision = 100 × Number of correct detections / Number of detections estimated as correct; Recall = 100 · Number of correct detections / Number of ground-truth object instances; Average precision-recall can be computed by taking the average of precisions values with varying thresholds (and thus varying recall values). The APR ranges from 0 to 100, and a higher APR indicates better performance.

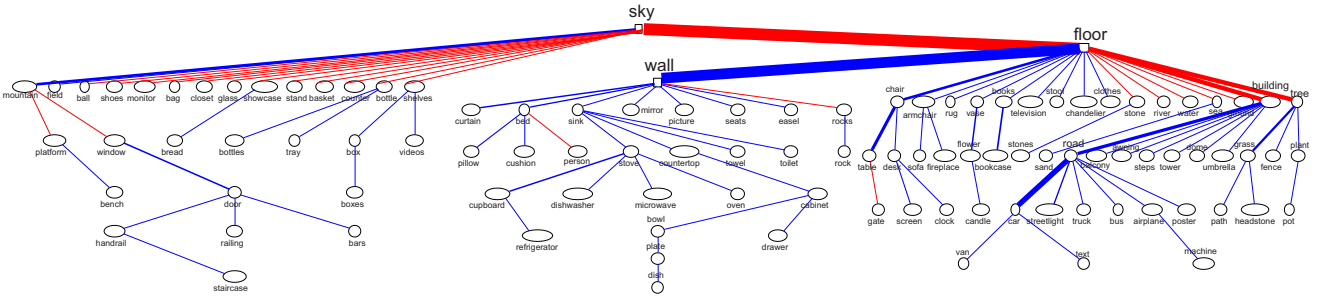

Fig. 6. Object dependency structure learned from SUN 09. Red edges denote negative correlation between categories. The thickness of each edge represents the strength of the link.

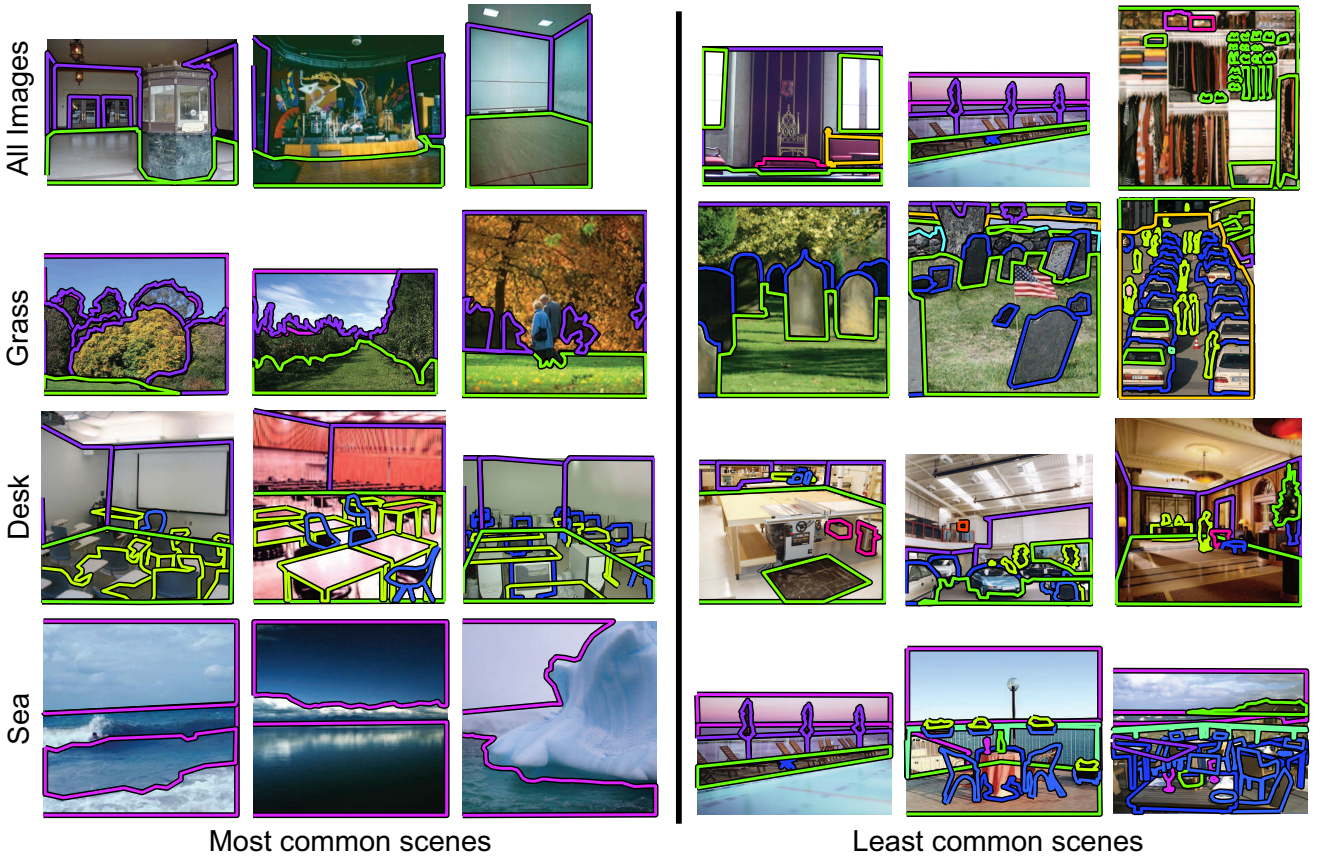

Fig. 7. The most typical scenes and the least typical scenes in the SUN 09 test set estimated using the context model. The first row shows scenes selected from all images, and the remaining rows show scenes that contain grass, desk, and sea, respectively. Only the outlined objects are used to evaluate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

jects. 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 ($\approx 13\%$) 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. 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

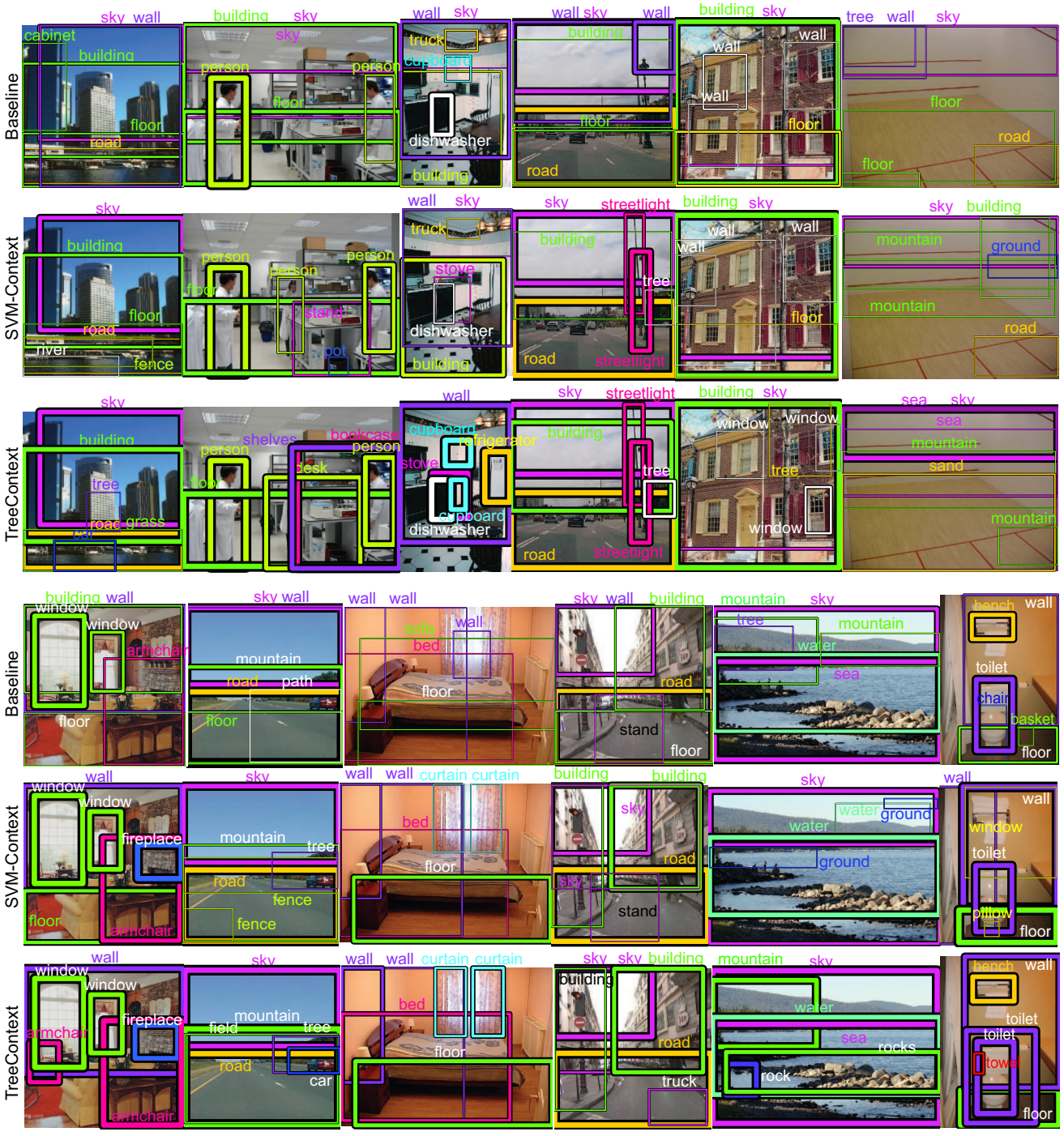

Fig. 9. Examples of scenes showing the six most confident detections with and without context. The figure shows successful examples of using context as well as failures.

are included in the 107 object categories, and all other objects are ignored. The three most common scenes among the entire test set consists 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, desk, and 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).

Object recognition performance Fig. 5(c-d) show localization and presence prediction results on SUN 09. We see bigger improvements from incorporating contextual information for both localization and

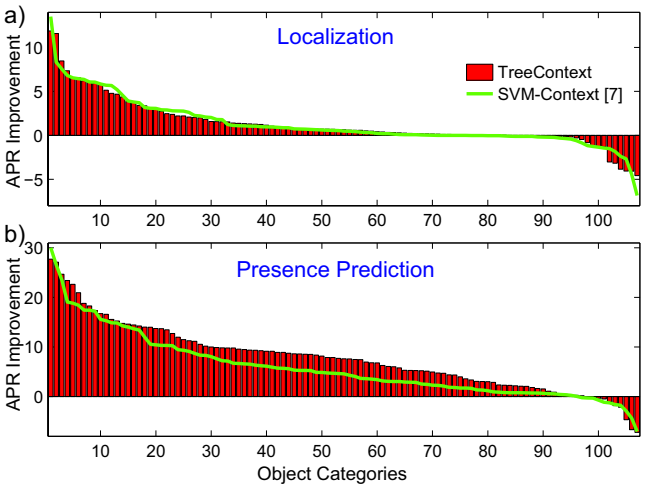

Fig. 8. Improvement of the context models over the baseline detectors. Object categories are sorted by the improvement in APR.

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% of images (and only 15% without context).

Fig. 8 show the improvement in average precision-recall (APR) for each object category sorted by the APR improvement over the baseline. Due to the large number of objects in our database, there are many objects that benefit in different degrees from context. Six objects with the largest improvement with TreeContext for object localization are floor (+11.88 over the baseline), refrigerator (+11.58), bed (+8.46), seats(+7.34), monitor (+6.57), and road (+6.55). In localization, the performance of TreeContext and SVM-Context are comparable - the APR averaged over all object categories is 7.06 for the baseline, 8.34 for SVM-Context, and 8.37 for TreeContext. In presence prediction, our tree-based context model (mean APR 25.7) performs better than both the baseline (mean APR 17.9) and SVM-Context (mean APR 23.8).

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 `truck` in a kitchen scene) or may lead to incorrect detections (e.g., hallucinating `car` because of a strong detection of `road` in the first image).

6.3 Scene Understanding Performance on SUN 09

The SUN 09 dataset 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 an image. Our context model performs significantly better than the baseline detectors in predicting whether an object is present or not as shown in Fig. 5 and Fig. 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 as well.

Fig. 10 shows precision-recall curves for image query results using different combinations 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,[3] 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 advantage over the baseline detectors, and in four of the five query sets, performs better than SVM-Context as well. Fig. 11 show examples of top queries using different methods. Note that even when the query result of TreeContext is incorrect, the content of the image strongly resembles that of a correct query result. 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 Detecting 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 images or objects that are unusual. In this section, we use a variation of our tree-based context model to detect objects out of context from each image.

We first consider a simpler 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

3. 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 for both the context model and the baseline detectors.

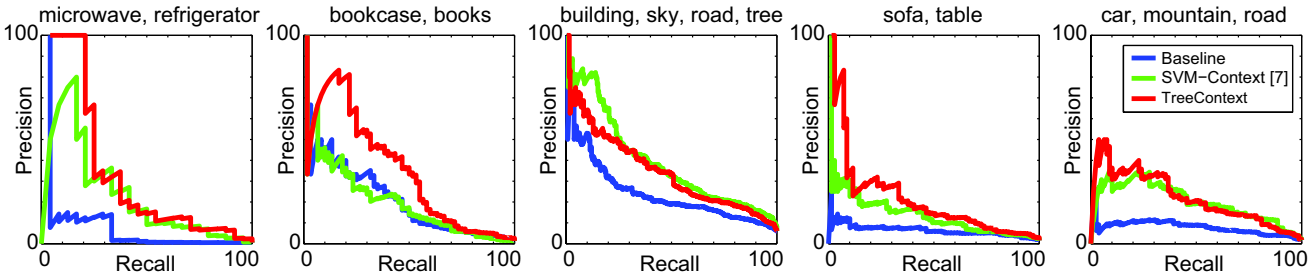

Fig. 10. Precision-recall curves for querying images with a set of object categories.

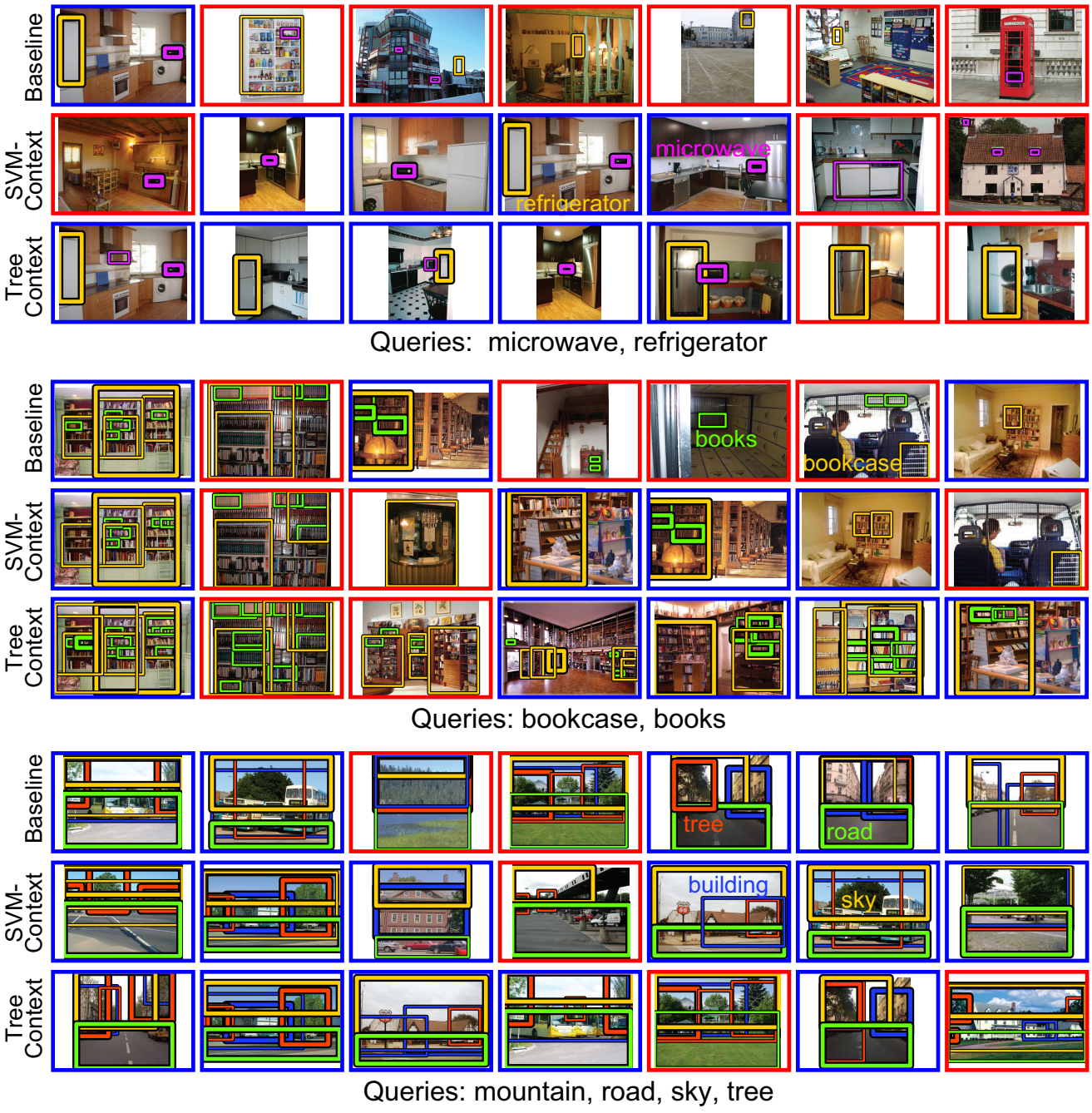

Fig. 11. Examples of top 7 images retrieved by the baseline detectors [7], context rescoring method with SVMs [7], and our tree-based context model. Correctly retrieved images (images in which all the objects in the query set are present) and shown in blue boxes, and incorrect images are shown in red 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 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.

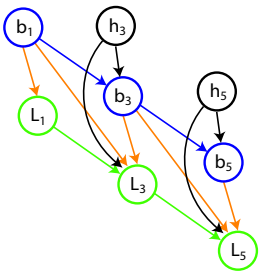

Fig. 13. A modified context model with new binary variables h_i's to detect objects out of context. If $h_3 = 0$, then (b_3, L_3) become independent from (b_1, L_1).

(see Fig. 3) for object dependencies 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 context model, but $p(b_3, L_3|b_1, L_1, h_3 = 0)$ is equal to the marginal probability $p(b_3, L_3)$ regardless of the values of b_1 and L_1, thus removing the dependencies between objects 1 and 3.

Conditioned on the ground-truth labels (b and L), the context variables h_i's are independent from 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,

$$\begin{aligned} p(h_i|b, L) &= p(h_i|b_i, b_{pa(i)}, L_i, L_{pa(i)}) \\ &= \frac{p(b_i, L_i|b_{pa(i)}, L_{pa(i)}, h_i)p(h_i)}{\sum_{h_i'} p(b_i, L_i|b_{pa(i)}, L_{pa(i)}, h_i')p(h_i')} \end{aligned} \quad (6)$$

and if we assume that $p(h_i) = 0.5$ for all i,

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

where

$$\begin{aligned} &C(b_i, b_{pa(i)}, L_i, L_{pa(i)}) \\ &\equiv \frac{p(b_i|b_{pa(i)}, h_i = 1)p(L_i|L_{pa(i)}, b_i, b_{pa(i)}, h_i = 1)}{p(b_i|b_{pa(i)}, h_i = 0)p(L_i|L_{pa(i)}, b_i, b_{pa(i)}, h_i = 0)}. \end{aligned} \quad (8)$$

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 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 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),

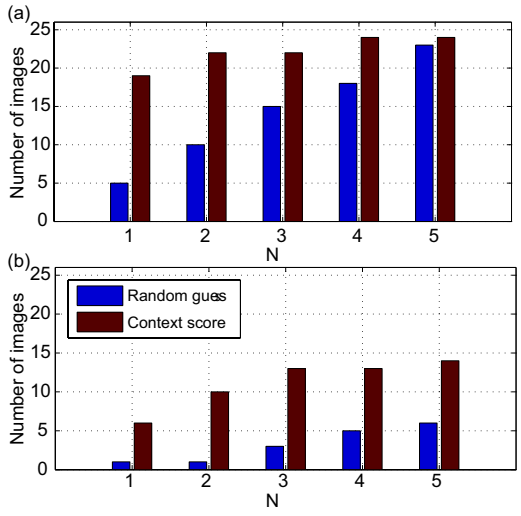

Fig. 14. The number of images in which at least one out-of-context object is included in the set of N most unexpected objects estimated by our context model. (a) Using ground-truth labels and segmentations. (b) Using local detector outputs.

we would like to estimate the probability of object i favoring to be independent from its parent object:

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

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,

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

where the context score $C(b_i, b_{pa(i)}, L_i, L_{pa(i)})$ is defined in (8). 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 (7).

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 2 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 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 descendents of the node `sink`, and all the vehicles are descendents of `road`. This suggests that a more intuitive structure for object dependencies could be a hierarchy including some meta-objects (such as a desk area) or scenes (kitchen or street) as nodes at coarser scales. Since it is not clear how many and what kind of meta-object nodes and scene nodes should be used, it is difficult to get training samples for such nodes from a human annotator. Hence, we need to discover those hidden nodes during our structure learning procedure. Learning a model with hidden nodes is in general a challenging problem, but for a certain class of tree models, there are efficient algorithms to learn a tree structure with hidden nodes using the samples of observed nodes [2]. The algorithm developed in [2] first learns a tree among the observed variables using the Chow-Liu algorithm, and then applies a local graph transformation to recover hidden nodes. Thus, our tree-based object dependency presented in this paper can be regarded as the output of the first stage of this learning procedure. Learning a full hierarchical tree structure with hidden nodes may discover important relationships among objects, meta-objects, and scenes, which is an interesting direction for further research.

The SUN 09 dataset presented in this paper has richer contextual information than PASCAL 07, which was originally designed for training and testing single object detectors. We demonstrate that our context model learned from SUN 09 significantly improves the accuracy of object recognition and image query results, and can even be applied to detect objects out of context. The SUN 09 dataset and the MATLAB implementation of our algorithm can be downloaded from http://people.csail.mit.edu/myungjin/HContext.html. Our experiments provide compelling evidence that rich datasets and modeling frameworks that incorporate contextual information can be more effective at a variety of computer vision tasks such as object classification, object detection, and scene understanding.

ACKNOWLEDGMENT

This research was partially funded by Shell International Exploration and Production Inc., by Army Research Office under award W911NF-06-1-0076, by NSF Career Award (ISI 0747120), and by the 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 Air Force.

REFERENCES

[1] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.
[2] M. J. Choi, V. Y. F. Tan, A. Anandkumar, and A. S. Willsky, "Consistent and efficient reconstruction of latent tree models," preprint.
[3] C. K. Chow and C. N. Liu, "Approximating discrete probability distributions with dependence trees," *IEEE Transactions on Information Theory*, 1968.
[4] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, 2005.
[5] C. Desai, D. Ramanan, and C. Fowlkes, "Discriminative models for multi-class object layout," in *ICCV*, 2009.
[6] 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.
[7] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.
[8] R. Fergus, P. Perona, and A. Zisserman, "Object class recognition by unsupervised scale-invariant learning," in *CVPR*, 2003.
[9] C. Galleguillos, A. Rabinovich, and S. Belongie, "Object categorization using co-occurrence, location and appearance," in *CVPR*, 2008.
[10] S. Gould, J. Rodgers, D. Cohen, G. Elidan, and D. Koller, "Multi-class segmentation with relative location prior," *International Journal of Computer Vision*, vol. 80, pp. 300–316, 2007.
[11] X. He, R. S. Zemel, and M. Á. Carreira-Perpinñán, "Multiscale conditional random fields for image labeling," in *CVPR*, 2004.
[12] G. Heitz and D. Koller, "Learning spatial context: Using stuff to find things," in *ECCV*, 2008.
[13] G. Heitz, S. Gould, A. Saxena, and D. Koller, "Cascaded classification models: Combining models for holistic scene understanding," in *NIPS*, 2008.
[14] D. Hoiem, A. Efros, and M. Hebert, "Automatic photo pop-up," in *SIGGRAPH*, 2005.
[15] ——, "Putting objects in perspective," in *CVPR*, 2006.
[16] Y. Jin and S. Geman, "Context and hierarchy in a probabilistic image model," in *CVPR*, 2006.
[17] L.-J. Li, R. Socher, and L. Fei-Fei, "Towards total scene understanding:classification, annotation and segmentation in an automatic framework," in *CVPR*, 2009.
[18] K. P. Murphy, A. Torralba, and W. T. Freeman, "Using the forest to see the trees: a graphical model relating features, objects and scenes," in *NIPS*, 2003.
[19] D. Parikh and T. Chen, "Hierarchical semantics of objects (hSOs)," in *ICCV*, 2007.
[20] J. Porway, K. Wang, B. Yao, and S. C. Zhu, "A hierarchical and contextual model for aerial image understanding," *CVPR*, 2008.
[21] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie, "Objects in context," in *CVPR*, 2007.
[22] B. C. Russell, A. Torralba, C. Liu, R. Fergus, and W. T. Freeman, "Object recognition by scene alignment," in *NIPS*, 2007.
[23] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "LabelMe: a database and web-based tool for image annotation," *International Journal of Computer Vision*, vol. 77, pp. 157–173, 2008.
[24] E. B. Sudderth, A. Torralba, W. T. Freeman, and A. S. Willsky, "Learning hierarchical models of scenes, objects, and parts," in *ICCV*, 2005.
[25] A. Torralba, K. P. Murphy, and W. T. Freeman, "Contextual models for object detection using boosted random fields," in *NIPS*, 2005.
[26] A. Torralba, "Contextual priming for object detection," *International Journal of Computer Vision*, vol. 53, pp. 169–191, 2003.
[27] Z. Tu, "Auto-context and its application to high-level vision tasks," in *CVPR*, 2008.
[28] J. Winn, A. Criminisi, and T. Minka, "Object categorization by learned universal visual dictionary," in *ICCV*, 2005.