Context Models and Out-of-context Objects

Myung Jin Choi, Antonio Torralba, Alan S. Willsky

Abstract

The context of an image encapsulates rich information about how natural scenes and objects are related to each other. Such contextual information has the potential to enable a coherent understanding of natural scenes and images. However, context models have been evaluated mostly based on the improvement of object recognition performance even though it is only one of many ways to exploit contextual information. In this paper, we present a new scene understanding problem for evaluating and applying context models. We are interested in finding scenes and objects that are "out-of-context". Detecting "out-of-context" objects and scenes is challenging because context violations can be detected only if the relationships between objects are carefully and precisely modeled. To address this problem, we evaluate different sources of context information, and present a graphical model that combines these sources. We show that physical support relationships between objects can provide useful contextual information for both object recognition and out-of-context detection.

Keywords:

1. Introduction

The context encapsulates rich information about how natural scenes and objects are related to each other, whether it be relative positions of objects with respect to a scene or co-occurrence of objects within a scene. Using such contextual information to improve object recognition has recently become popular [1, 2, 3, 4, 5, 6, 7, 8, 9] because contextual information can enforce a coherent scene interpretation, and eliminate false positives. Based on this success, context models have been evaluated on how much the context model improves the object recognition performance. However, comparing context models solely based on object recognition can be misleading because object recognition is only one of many ways to exploit the context information, and object recognition cannot adequately evaluate some of the dimensions in which the context model can be useful.

For example, context information can help predict the presence of occluded objects, which can be useful for robotics applications in which a robot can move to view occluded objects. Context information can also help predict the absence of important objects such as a TV missing in a living room. We can also use contextual information to suggest places to store objects, which can be useful when a robot tries to decide where to place a TV in a living room. We cannot evaluate the effectiveness of different context models in these scenarios just by evaluating the context models on object recognition tasks.

In this work, we are interested in finding scenes and objects that are "out-of-context". This application can be amenable to evaluating dimensions of context models not adequately evaluated by object recognition tasks. Fig.1 shows several out-of-context images with objects in unexpected scenes or in unexpected locations. Detecting out-of-context images is different from detecting changes in surveillance applications because the goal in surveillance is to identify the presence or absence of certain objects in a known scene, most likely with video data. In our problem setting, the task is detecting an object that is unusual for a given scene in a single image, even if the scene has not been observed before. Therefore, we need contextual relationships between objects to solve this problem. Detecting out-of-context objects can be challenging because contextual violations can be detected only if the relationships between objects are carefully and precisely modeled. For example, in the second image in Fig.1, many elements are in correct locations, but because a road sign appears next to an airplane, the airplane is out of context.

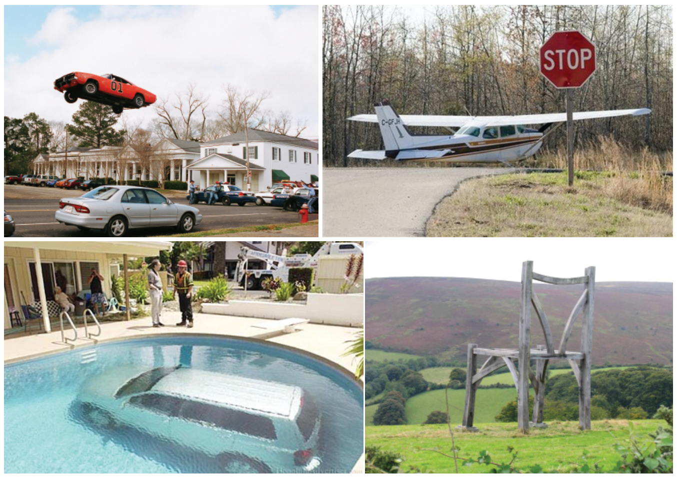


Figure 1: Examples of objects out of context (violations of support, probability, position, or size).

In addition to providing a new application of context models, we analyze different sources of contextual information and propose a graphical model that integrates them. We evaluate this context model in object recognition tasks on a SUN dataset [10] and analyze how much each contextual information contributes to the improved performance. We also test our context model in detecting out-of-context objects using (1) ground-truth labels and (2) noisy detector outputs, and show performance improvements compared to other context models.

2. Sources of Contextual Information

Biederman [11] provides five features that are important for human vision: support (objects should not be floating), interposition (objects should occupy different volumes), probability (objects should or should not appear in certain scenes), position (objects should appear in typical locations), and size (object have typical relative sizes). We analyze potential sources of contextual information that encode some of these features.

2.1. Global Context Models

The knowledge of scene category can help recognize individual objects in a scene, but identifying individual objects in a scene can also help estimate the scene category. One way to incorporate this idea in object recognition is to learn how likely each object appears in a specific scene category. The gist descriptor [12], which captures coarse texture and spatial layout of a scene, can be used to this end. For example, Murphy et al. [13] train gist using 15 pre-specified scene categories and use the gist regressor to adjust the likelihood of each detected object. Using the scene category information can greatly enhance object recognition performance, but hand-selected scene boundaries can be artificial, and sharing parameters among similar types of scenes, such as a street and a city, can be challenging.

Instead of first predicting the scene category and then estimating the presence of an object, we could use gist directly to predict the presence of an object. It is especially effective in predicting the presence of large objects with texture such as sky, sea, and mountain (commonly called *stuff*). The gist descriptor is also known to work well in predicting the expected vertical location of such objects in an image [14].

2.2. Object Co-occurrences

Some objects co-occur often, and some objects rarely appear together. Object co-occurrence statistics provide strong contextual information and have been widely used in context models [15, 4, 16, 10, 17]. A common framework for incorporating the co-occurrence statistics is conditional random field (CRF). An image is segmented into coherent regions or super pixels, and each region or super pixel becomes a node in a CRF. For example, Rabinovich et al. [15] first predict the labels of each node using local features, and adjust the predicted labels using pair-wise co-occurrence relationships. In Ladicky et al. [4], global potentials are defined to encode co-occurrence statistics and to encourage parsimonious interpretation of an image. Torralba et al. [16] combine boosting and CRFs to first detect easy objects (e.g., a monitor) and use contextual information to detect difficult objects that co-occur frequently with the detected objects (e.g., a keyboard). If the number of object categories is large, a compact representation of object co-occurrences can avoid over-fitting and enable efficient inference and learning algorithms. Choi et al. [10] learn a tree-structured graphical model to capture co-occurrence statistics of more than 100 object categories.

Due to computational complexity, most work focus on capturing pairwise co-occurrence statistics. However, some relationships require richer representation. For example, toilet and sink co-occur often, but a triplet (toilet, sink, refrigerator) can be unusual. Felzenszwalb et al. [17] addresses this issue using a support vector machine, which re-scores each detection using the maximum score of all other object categories detected in the same image.

2.3. Geometric Context

Knowing where objects are likely to appear is helpful for object localization. This information can be captured using geometric context. Geometric context arises because (1) most objects are supported by other objects, e.g., cars are supported by road, and people are supported by floor or sidewalk; (2) objects that have a common function tend to appear nearby and have a certain spatial configuration, e.g., a computer screen appears above a keyboard, and a mouse is located on the left or right of the keyboard; and (3) humans tend to take photographs with a common layout, e.g., floor is typically at the lower half of an image, and sky is in the upper half. Torralba et al. [12] note that the vertical locations of an object (either absolute or relative to other objects) is often more informative than its horizontal location. Hoiem et al. [18] introduce an explicit representation of the 3D geometry of the scene (i.e., the horizon line and the distinction between horizontal and vertical surfaces) .

Quantitative geometric models. One way to incorporate geometric information is by using Gaussian variables to model likely relative positions and scales of objects [10]. It is also common to represent an object location in a non-parametric way by dividing an image into a finite number of regions. Gould et al. [3] construct a non-parametric probability map by learning quantized representations for relative locations between pairs of object categories. Yao and Fei-fei [5] use binning function to represent location relationships between human body parts and a Gaussian distribution to represent relative scales.

Qualitative geometric models. In the real world, qualitative relationship among object locations is as important as quantitative relationships. Cars should be supported by a ground or road, and it is unlikely that a car is floating above a road, even if the distance between the two is small. In Galleguillos et al. [2], spatial relationships between pairs of segmented regions are quantized to four prototypical relationships - `above, below, inside, around`. A similar set of spatial relationships is used in Desai et al. [1] with the addition of `far` to capture non-local spatial relationships. Russell and Torralba [19] use attachment (a wheel is a part of a car) and supported-by (a car is supported by a road) to represent spatial relationships between overlapping polygons of object boundaries, and use those relationships to reconstruct a 3D scene from user annotations.

3. Support context model

We integrate different sources of contextual information mentioned in the previous section using a graphical model. Our context model takes the gist descriptor, local detector scores and bounding box locations as inputs, and computes the probability of each object's presence and the likelihood of each detection being correct. Our model consists of two tree-structured graphical models: the first part relates object categories using co-occurrence statistics, and the second part relates detector outputs using support relationships.

3.1. Latent tree model for co-occurrences

For each object category i, we associate a binary variable x_i to represent whether it is present or not in an image. Choi et al. [10] uses a tree-structured graphical model with these binary variables to capture co-occurrence probabilities between object categories. They note that just learning a tree structure results in a natural hierarchy of objects in which a representative object (e.g., sink) is placed at the root of a subtree of objects that commonly appear in the same scene (e.g., kitchen).

In this work, we add latent binary variables to a co-occurrence tree model to capture the dependencies of object categories due to scenes. Our co-occurrence latent tree model consists of observed binary variables representing each object category and latent binary variables representing some unspecified scenes or meta-objects. These latent variables are learned from a set of training images using the method we describe in Section 4. The additional latent variables allows a richer representation of object relationships. For example, a toilet is more likely to be present if a sink is present, but not if it is in a kitchen scene. Fig.2a shows an example of a small co-occurrence latent tree for 6 object categories.

3.2. Gist for global context

The co-occurrence latent tree model implicitly infers the context of an image by collecting measurements from all object categories. However, if there are false detector outputs with strong confidence, it is possible that those false detections confuse the co-occurrence tree to infer a wrong context. Thus, we use gist in addition to local detectors to enhance the context-inferring power of our co-occurrence tree model. Since a gist descriptor is especially effective in classifying scenes and meta-objects, we use gist as a measurement of each latent variable in the co-occurrence tree.

3.3. Support tree for quantitative geometric context

We use a binary detector variable c_{ik} to denote whether a detection k of object category i is correct or false. Each detector variable has an associated score s_{ik} and the location of the bounding box y_{ik}. By using relative locations of detector bounding boxes, we can prune out false positives with support violations. For example, if we have a strong detection of a floor and multiple detections of tables, then it is more likely that those supported by the floor detection are correct.

Given an image, we infer support relationships among detector outputs and construct a support tree model. Fig. 2b shows an example image with detector outputs and the its estimated support tree. The edge potentials of the support tree conditioned on presence variables $p(c_{ik}|x_i, c_{jl})$ encode that a detection k of object i is more likely to be correct if it is supported by a correct detection l of object j.

3.4. Expected locations and scales of bounding boxes

We use simple binning functions to represent the location and the size of a bounding box relative to the image height. The probability $p(y_{ik}|c_{ik})$ encodes the frequency that the bottom of a bounding box belongs to bottom, middle, or top part of an image, and the frequency that the height of the bounding box is less than a quarter, less than a half, or larger than a half of the image height. Together with the support tree, our context model captures relationships between object categories qualitatively and expected location and scale of correctly detected bounding boxes quantitatively.

4. Model Learning

Given a set of fully-labeled training images, we learn the structure and parameters of the co-occurrence latent tree model, the measurement model for gist, detector scores, and bounding box locations, and finally the parameters for inferring support trees in new images.

4.1. Co-occurrence latent tree

We assume that only the labels of object categories are given, and the scene information is unknown in the training image. Thus, it is necessary to learn the number of latent variables and how they are connected to object variables just from the samples of object presence variables. We use a recent approach introduced in [20] to efficiently learn a latent tree graphical model from samples of observed variables. Our tree model relating 107 object categories learned from SUN 09 training set [10] includes 26 hidden variables, many of which can be interpreted as corresponding to scene categories such as indoor, bedroom, or street. Fig. 3 shows the structure of a latent tree learned from co-occurrence statistics of 107 object categories. Many hidden variables in the latent tree can be interpreted as scene categories. For example, h_4 corresponds to an outdoor scene, and is negatively correlated with wall, floor, and h_5, which corresponds to an indoor scene. In addition, h_6, h_{15}, and h_{13} can be interpreted as a kitchen, a living room, and a street, respectively.

After learning the tree structure, the parameters of the tree model is estimated using the EM algorithm, which is efficient for a tree graphical model with binary variables.

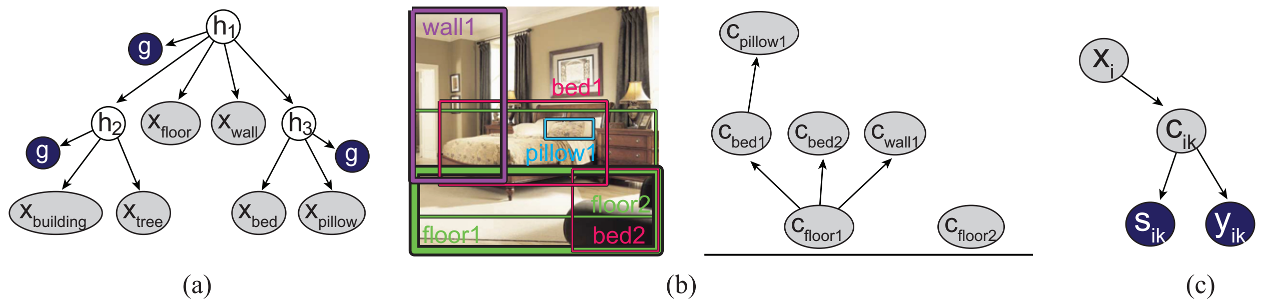
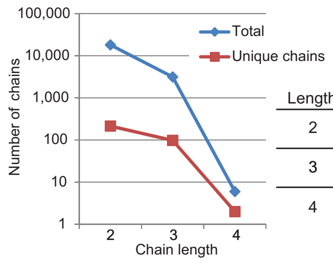


Figure 2: An illustrative example of our support context model for 6 object categories. White nodes are hidden, solid nodes are observed, and grey nodes are observed only during training. (a) Co-occurrence latent tree with the gist descriptor g to infer hidden variables. (b) An example image with a few detector outputs and its inferred support tree. The tree is drawn upside down to emphasize that a parent variable is physically supporting its child variables. (c) The connection between a presence variable x_i and its detector variable c_{ik} is drawn separately here to simplify the figures. Each detector variable has an associated score s_{ik} and the location of the bounding box y_{ik}.



Length	Two most common chains	Number
2	window/ building	2805
	car/ road	1415
3	window/ wall/ floor	387
	books/bookcase/ floor	336
4	flowers/ vase/ table/ floor	4
	candle/ tray/ table/ floor	2

Figure 4: Distribution of support-chains in the SUN 09 training set.

Measurement models for gist, local detector scores, and bounding box locations. After learning the structure and parameters of a co-occurrence latent tree, we estimate the values of hidden variables for each training image. Using these estimated samples, we train the likelihood of gist conditioned on hidden variables [13]. The probability of correct detection conditioned on a detector score $p(c_{ik}|s_{ik})$ is trained using logistic regression. The probabilities of correct detection given presence variable $p(c_{ik}|x_i)$, and the quantized heights and bottom locations of bounding boxes relative to image heights $p(y_{ik}|c_{ik})$ are trained by counting in the training set.

4.2. Parameters for support trees

A majority of object categories such as cars, buildings, and people are always supported by other objects such as roads, sidewalks, and floors. Russell and Torralba [19] construct a support tree in an image annotated by humans using the following approach:

Training For each pair of object categories i and j, count $N_1(i,j)$, the number of times that an instance of i and an instance of j appear together in an image in the training set, and $N_2(i,j)$, the number of times that the bottom point of the instance of i is inside the instance of j. For each object i, and for a given threshold θ_1, obtain a list of possible supporting objects $S_{\theta_1}(i) := \{j \text{ s.t. } N_2(i,j)/N_1(i,j) > \theta_1\}$. In all our experiments, we use $\theta_1 = 0.25$.

Constructing a support tree in an image Let (i,k) be the k-th object instance of object category i. For each (i,k) in the image, choose a supporting object instance (j,l) such that

$$\frac{N_2(i,j)}{N_1(i,j)} e^{-\alpha d_{ik;jl}} \qquad (1)$$

is maximized, where $d_{ik,jl}$ is the distance from (j,l) to the bottom point of (i,k). The exponential function is used to only allow instances that are adjacent or a few pixels apart due to annotation errors to be considered as potential supporting pairs. If the maximum of (1) is less than a threshold θ_2, then (i,k) is not supported by any other object. We use $\alpha = 0.1$ and $\theta_2 = 0.1$. Since each object instance chooses at most one supporting object, the resulting support relationships form a tree structure.

In a support tree, we can consider a chain of support relationships from the top to the ground objects. For example, (a plate supported by a table supported by a floor) is a length-3 support chain. Fig.4 shows the number of total and unique chains, and the two most common chains with lengths 2, 3, and 4 in the SUN 09 training set. Note that a majority of support chains have length 2 or 3. The number of unique chains is significantly less than the number of total chains, which implies that the support relationships are consistent across images.

We extend the above approach to infer support relationships among detector outputs and to construct a support tree for a new image. Let $N_3(i,j)$ be the number of times that j is supporting i in support trees in the training set. Let $p_{sup}(i;j) = N_3(i,j)/N_1(i,j)$. Intuitively, if $p_{sup}(i;j)$ is high, then it is more likely that i is supported by j whenever both objects appear together. We define $p_{sup}(i;0)$:= (Number of instances that i appears without a support / Number of instances of i). Ground objects such as a road and a floor have $p_{sup}(i;0) = 1$. For other objects, $p_{sup}(i;0)$ can be high if a supporting object of i is typically occluded (e.g., ground supporting a building is often occluded) or not on the list of our 107 object categories (e.g., a person can be supported by a sidewalk, which is not on our list).

A bounding box B_1 is supported by another bounding box B_2 if the bottom side of B_1 lies entirely within B_2. We construct a support tree using detector outputs by selecting a bounding box (j, l) that maximizes

$$p_{sup}(i, j)p(c_{jl} = 1) \tag{2}$$

among all bounding boxes that support (i, k). This encourages a more confident detection to be selected as a supporting object. If the maximum is less than θ_2, then (i, k) is not supported by any other object. Fig.5 shows examples of support trees inferred from detector outputs. The edge potentials $p(c_{ik}|x_i, c_{jl})$ on the support tree can be trained by counting the parent-child pairs of correct detections in the training set.

5. Using support-context for object recognition

Given local detector outputs in a new image, we infer the presence and detector variables using the following step: (1) For each detection, estimate the probability of correct detection c_{ik} using detector scores s_{ik} and bounding box locations y_{ik}. (2) Infer x_i's using the co-occurrence latent tree with gist of the image g as additional measurements. This step implicitly infers the global context of the image and encourage objects that are semantically coherent to be present. (3) Re-estimate $p(c_{ik} = 1)$ using the marginal probabilities of the presence variables x_i, and construct a support tree. The edge potentials of the support tree are approximately computed using $p(c_{ik}|c_{jl}) \approx \sum_{x_i} p(c_{ik}|x_i, c_{jl})p(x_i)$, which results in better estimates than sampling x_i's and conditioning on their values. (4) Update the probabilities of correct detection $p(c_{ik} = 1)$ using the support tree with the edge potentials computed in the previous step. This encourages a detection supported by a strong detection of its typical supporting object to be correct.

We iterate between steps (2)-(4), which generally converges in a couple of iterations. Note that each of the step corresponds to passing messages in our graphical model, and the message-passing schedules are designed to best utilize the part-tree structures in the context model. Each step in our inference algorithm corresponds to passing messages in a binary tree, so it is efficient even with more than a hundred object categories.

6. Context models for out-of-context detection

In this section, we describe algorithms for out-of-context detection using context models. In [10], a tree-structured co-occurrence tree with binary presence variables and Gaussian location variables has been proposed. It has been demonstrated that given ground-truth labels of an image, the model detects co-occurrence violations accurately, but not position or support violations. This is because a Gaussian distribution is in general poor at enforcing strong constraints, which makes it inappropriate in modeling support relationships.

Let o_{ik} be a binary variable indicating whether (i, k), a detection k of object i, is out of its normal context. We are interested not only in detecting out-of-context objects but also in predicting whether the violation is in co-occurrences or support. Thus, we consider additional binary variables oc_i and os_{ik} to represent out-of-context due to co-occurrences (which does not depend on k) and support, respectively. We assume that an object is out-of-context if it violates either co-occurrence or support constraints, and compute the probability of out-of-context as

$$p(o_{ik} = 1) = \max(p(oc_i = 1), p(os_{ik} = 1)). \tag{3}$$

For an out-of-context object, we assume that it has a support violation if $p(oc_i = 1) < p(os_{ik} = 1)$, and a co-occurrence violation otherwise.

Using Ground-truth Labels. Let us first consider the case when ground-truth labels are available. To detect co-occurrence violations, we use a similar approach as in [10] and assume that if $oc_i = 1$, the presence variable x_i is independent of all other presence variables, thus independent of the context imposed by other objects. Fig.6a shows a graphical model with presence variables and the corresponding out-of-context variables. Conditioned on ground-truth labels of presence variables $\mathbf{x}$, out-of-context variables $\{oc_i\}$ and latent scene variables $\{h_i\}$ form a tree structure, so the marginal probability $p(oc_i = 1|\mathbf{x})$ can be computed efficiently.

Recall from Section 4.2 that given the ground-truth labels of an image, we choose a supporting instance of (i, k) by finding an instance (j, l) that maximizes $p_{sup}(i, j)e^{-\alpha d_{ik;jl}}$. Since this value lies in the range $[0, 1]$, we can consider the quantity as the probability that the instance (i, k) is "in-context" with regard to the support relationships. Thus, the probability of (i, k) out-of-context due to support violations can be computed as

$$p(os_{ik} = 1) = 1 - p_{sup}(i, j)e^{-\alpha d_{ik;jl}} \tag{4}$$

where (j, l) is the supporting object instance of (i, k). If an instance (i, k) does not have a supporting object instance, then

$$p(os_{ik} = 1) = 1 - p_{sup}(i, 0). \tag{5}$$

Using Detector Outputs. Detecting objects out-of-context using noisy detector outputs is a challenging task. If there are strong false alarms, it may distort the contextual information of the entire image, and if the actual out-of-context object is weakly detected, it is hard to differentiate it from many other false positives. In some of the images, out-of-context objects have distorted appearances (e.g., a car in the swimming pool in Fig.1), making it more difficult for local detectors to confidently detect the objects. Unlike in normal images, we cannot use context models to improve the scores of such detections, since it is likely that context models further lower the confidence of out-of-context detections.

The probability of an object instance out-of-context can be computed as follows:[1]

$$p(o_{ik} = 1) = p(o_{ik} = 1|c_{ik} = 1)p(c_{ik} = 1) + p(c_{ik} = 0). \tag{6}$$

[1]Here, our goal is to detect an out-of-context object present in an image, so if an object is not present, it is not considered as out of context. It is also possible to consider the problem of detecting a missing object (e.g., a living room without a sofa).

The first term, the probability of ik out of context assuming that it is a correct detection, can be computed similarly to the case when ground-truth labels are available. For co-occurrence violations, we generate samples of presence variables using detector scores, and compute $p(oc_i = 1|x_i = 1)$ by averaging over samples.

In order to consider the probability of support violations, we first infer a support tree using detector outputs. For each instance (i, k) and its supporting instance (j, l), we assume that (i, k) does not violate support constraints if (1) (j, l) is a correct detection and i is typically supported by j or (2) (j, l) is not a correct detection but i can appear without a support. Therefore, the probability of a support violation can be computed as follows:

$$\begin{aligned} & p(os_{ik} = 1|c_{ik} = 1) \\ = \quad & 1 - \Big(p_{sup}(i; j)p(c_{jl} = 1) + p_{sup}(i; 0)p(c_{jl} = 0)\Big) \qquad (7) \end{aligned}$$

Fig.6b shows a support tree with two detector variales and the corresponding out-of-context variables representing support violations.

Out-of-context Detection Using Other Context Models. Out-of-context object detections for general context models can be performed by comparing the confidence before and after applying the context model. For example, for the SVM-context model in [17], we set $p(o_{ik} = 1|c_{ik} = 1) := s_{ik}/(\bar{s}_{ik} + s_{ik})$, where $\bar{s}_{ik}$ is the score adjusted by SVM using the scores of all other object categories. Note that such methods do not provide answers for whether the violation is due to co-occurrence or support.

7. Experiments

In this section, we show experimental results of (1) detecting objects in normal images and (2) identifying out-of-context objects. We use the discriminative part-based models in [17] as baseline local detectors, and compare object recognition and out-of-context object detection performances with two other context models - tree-based context models with Gaussian location variables in [10] and SVM re-score method in [17].

7.1. Recognition Performance

We use the SUN 09 dataset introduced in [10] for object recognition evaluation of 107 object categories, which contains 4,367 training images and 4,317 testing images [10]. Table 1 shows the average precision-recalls (APRs) of object localization for selected object categories, and the mean APR averaged over all 107 object categories. For our support context model, we show the results using each component alone and the full context model with all contextual information combined together. The support model without co-occurrence tree has limited performance since there are many false positives of floor or road in almost all images, but when combined with other components, it increases performances especially for indoor objects. Despite the differences in encoding contextual information, all context models have similar performances.

Category	Baseline	Support	Location	Co-occurr	Gist + Co-occur	Full context	SVM-Context	Gauss-Context
bookcase	2.32	5.42	2.39	3.95	3.8	5.21	2.95	4.71
bowl	0.90	0.80	0.83	1.47	3.11	3.13	0.69	2.45
floor	31.34	36.64	31.33	39.85	40.99	43.48	44.85	43.22
mountain	17.23	16.65	17.62	17.16	17.49	17.90	20.12	18.38
oven	8.07	8.07	8.04	11.56	14.74	14.34	6.62	10.27
sky	55.34	55.34	55.78	56.68	57.64	58.57	61.05	60.48
sofa	11.47	11.74	11.86	11.67	13.65	14.34	12.34	15.30
tree	10.88	13.29	12.16	11.35	12.41	12.70	13.65	12.69
AVERAGE	7.06	7.09	7.40	7.44	8.02	8.37	8.34	8.37

Table 1: Average precision-recall for localization and presence prediction. Baseline) baseline detector without contextual information [17]; Full context) Support context model; SVM-Context) Context rescoring method in [17]; Tree-context) Co-occurrence tree with Gaussian model for spatial relationships [10]

Fig.7 shows example images with six most confident detections using the baseline local detectors (first row) and our support context model (second row). Note that the context model effectively removes semantically-incorrect false positives (e.g., a floating refrigerator in the first image), and increases the probabilities of correct detections that satisfy co-occurrences (e.g., streetlight on the street) and support (e.g., sink supported by countertop, armchair supported by floor). Note that in some cases, the confident detections by context models can be semantically coherent but incorrect, as shown in the last image.

7.2. Detecting Objects out-of-context

We extended the out-of-context dataset first introduced in [10], and collected 209 out-of-context images. Among these, we select 161 images that have at least one out-of-context object corresponding to one of the 107 object categories in our model. Fig.8 shows examples of detecting objects out-of-context using our support context model. Segments highlighted in red are chosen as out-of-context due to co-occurrence violations and segments in yellow are chosen as objects with support violations. Note that for the last two images, we need a deeper understanding of the scene which may not be captured by co-occurrences and support relationships alone. In the last image, it appears as if truck is supported by a car due to occlusion, and the car squeezed by two buses is not considered out-of-context.

Even with such challenging images included in the dataset, our context model performs well as shown in Fig.10(a). The plot shows the number of images in which at least one out-of-context object is included in the top N most unexpected objects estimated by our context model. Both our support and co-occurrence models, even when used separately, outperform the tree-context with Gaussian location variables [10], and our combined model selects the correct out-of-context object instance as the most unexpected object in 118 out of 161 images.

In Fig.9, we show success and failure cases of detecting out-of-context objects using our context model based on detector outputs. In each image pair, the first image shows six most confident detections using local detectors, and the second image shows three most confident out-of-context objects estimated by our context model (red for co-occurrences and yellow for support). In the last row, we show failure cases due to noise in detector outputs: in the first image, the false positive of a window has a higher detector score and is also out-of-context, and

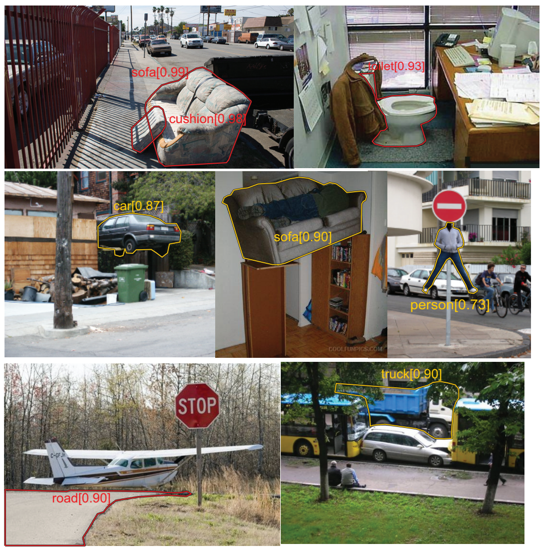


Figure 8: Example images of detecting out-of-context objects using ground-truth labels. For each image, objects with the probability of being out-of-context greater than 0.9, or one object with the highest probability is shown. For the full support context model, objects in yellow have been selected due to the support tree, and in red due to the co-occurrence latent tree.

in the second image, none of the car detections have a strong confidence score. More example images are presented in [21].

Fig.10(b) shows the number of images in which top N unexpected object category includes at least one true out-of-context object. Although context models perform better than a random guess, most models except our support model do not outperform the baseline of sorting objects by their strongest detection scores. One reason for this is a photographer bias - some photographs in the dataset have been taken with a strong focus on out-of-context objects, which tend to make their detections highly confident. In other types of images in which out-of-context objects are not detected well, it is difficult to differentiate them from other false positives as illustrated in Fig.9. It is interesting to note that our support model significantly outperforms all other context models (the performance of the full context model combining co-occurrence and support drops slightly due to mistakes made by the co-occurrence part). This implies that physical support relationships do provide useful contextual information.

8. Conclusion

We analyze different sources of contextual information, and present a new context model that incorporates global contextual information, object co-occurrence statistics, and geometric context. An interesting component of our context model is a physical support relationship between object instances, which provides a useful contextual information in scene understanding. We demonstrate the performance of our context model for both object recognition and out-of-context object detection.

References

[1] C. Desai, D. Ramanan, C. Fowlkes, Discriminative models for multi-class object layout, in: International Conference on Computer Vision (ICCV), 2009.

[2] C. Galleguillos, A. Rabinovich, S. Belongie, Object categorization using co-occurrence, location and appearance, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2008.

[3] S. Gould, J. Rodgers, D. Cohen, G. Elidan, D. Koller, Multi-class segmentation with relative location prior, International Journal of Computer Vision (IJCV) 80 (3) (2007) 300–316.

[4] L. Ladicky, C. Russell, P. Kohli, P. Torr, Graph cut based inference with co-occurrence statistics, in: European Conference on Computer Vision (ECCV), 2010.

[5] B. Yao, L. Fei-Fei, Modeling mutual context of object and human pose in human-object interaction activities, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010.

[6] G. Heitz, D. Koller, Learning spatial context: Using stuff to find things, in: European Conference on Computer Vision (ECCV), 2008.

[7] Y. Jin, S. Geman, Context and hierarchy in a probabilistic image model, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2006.

[8] L.-J. Li, R. Socher, L. Fei-Fei, Towards total scene understanding:classification, annotation and segmentation in an automatic framework, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009.

[9] Z. Tu, Auto-context and its application to high-level vision tasks, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2008.

[10] M. J. Choi, A. Torralba, A. S. Willsky, A tree-based context model for object recognition, to appear in IEEE Transactions on Pattern Analysis and Machine Intelligence (2011).

[11] I. Biederman, R. J. Mezzanotte, J. C. Rabinowitz, Scene perception: Detecting and judging objects undergoing relational violations, Cognitive Psychology 14.

[12] A. Torralba, Contextual priming for object detection, International Journal of Computer Vision (IJCV) 53 (2003) 2.

[13] K. P. Murphy, A. Torralba, W. T. Freeman, Using the forest to see the trees: a graphical model relating features, objects and scenes, in: Advances in Neural Information Processing Systems (NIPS), 2003.

[14] A. Torralba, K. Murphy, W. T. Freeman, Using the forest to see the trees: object recognition in context, Comm. of the ACM, Research Highlights 53 (2010) 107–114.

[15] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, S. Belongie, Objects in context, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2007.

[16] A. Torralba, K. P. Murphy, W. T. Freeman, Contextual models for object detection using boosted random fields, in: Advances in Neural Information Processing Systems (NIPS), 2005.

[17] P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part based models, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (9) (2010) 1627–1645.

[18] D. Hoiem, A. Efros, M. Hebert, Putting objects in perspective, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2006.

[19] B. C. Russell, A. Torralba, Building a database of 3d scenes from user annotations, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010.

[20] M. J. Choi, V. Y. F. Tan, A. Anandkumar, A. S. Willsky, Learning latent tree graphical models, Journal of Machine Learning Research (JMLR) 12 (2011) 1771–1812, arXiv:1009.2722v1.

[21] M. J. Choi, Trees and beyond: Exploiting and improving tree-structured graphical models, Ph.D. thesis, Massachusetts Institute of Technology (February 2011).

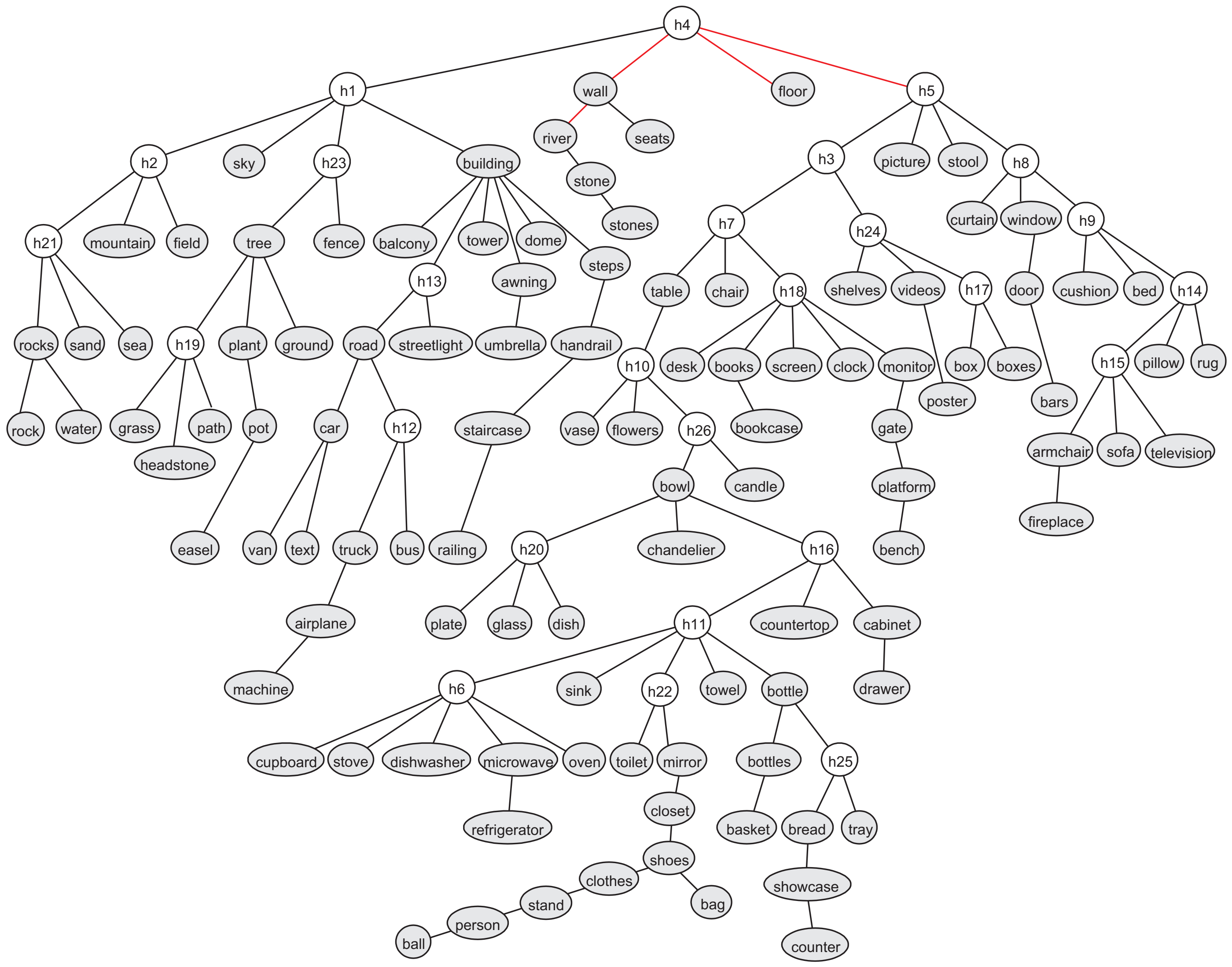


Figure 3: The structure of the latent tree model capturing object co-occurrences among 107 object categories. Red edges denote negative relationships.

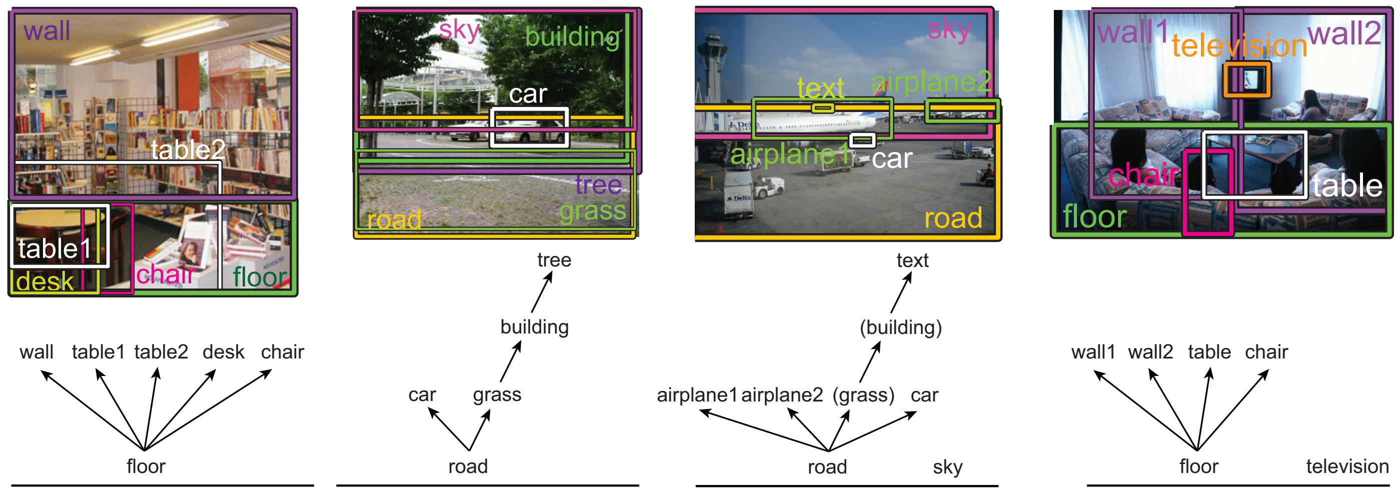


Figure 5: Examples of support trees inferred using detector outputs. For each image, we show six most confident detections and a part of the inferred support tree relating those six detections.

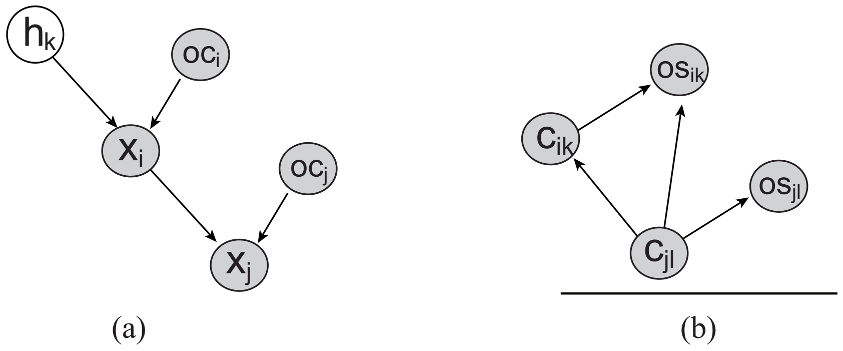


Figure 6: Context model for out-of-context detection. (a) Binary variables $\{oc_i\}$ representing out-of-context due to co-occurrences. If $oc_i = 1$, the corresponding presence variable x_i is independent of all other presence variables. (b) A support tree and the out-of-context variables $\{os_{ik}\}$ representing support violations. The probability of $os_{ik} = 1$ depends on both c_{ik} and c_{jl}.

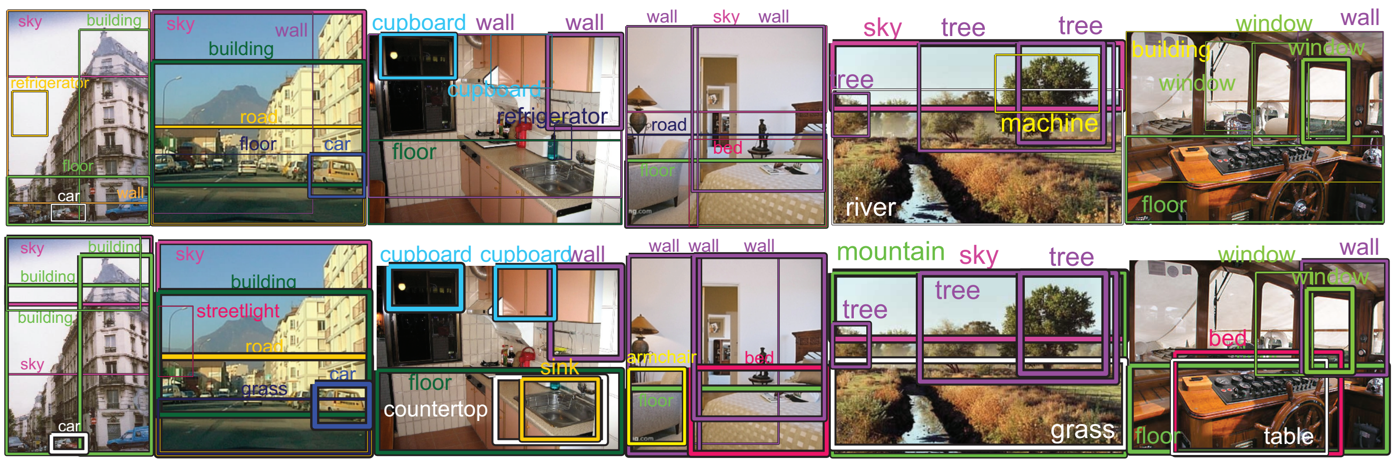


Figure 7: Examples of images showing six most confident detections using the baseline local detector (first row) and our context model (second row). The thickness of bounding boxes indicate the confidence of each detection.

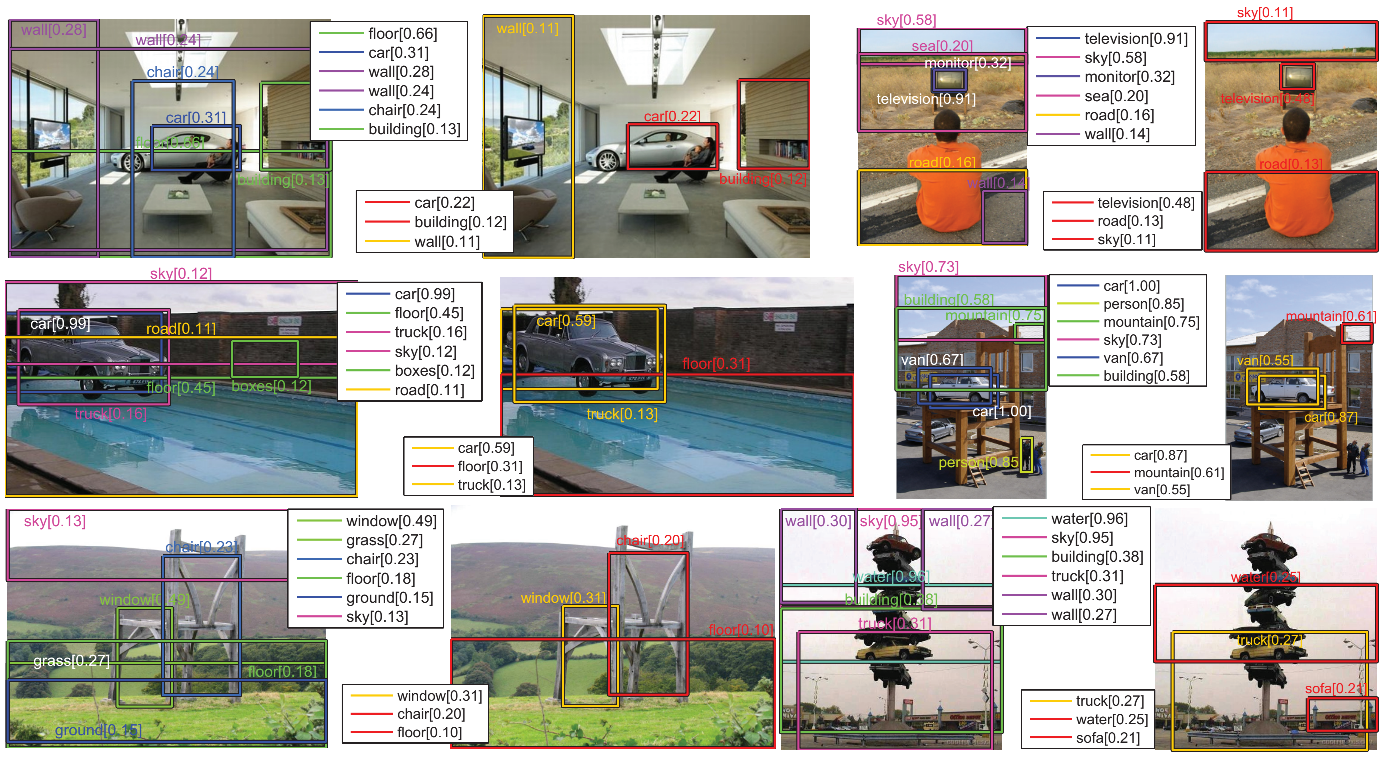


Figure 9: Examples of detecting out-of-context objects using detector outputs. On the left, six most confident detections (using only local detector scores) are shown, and on the right, three most confident out-of-context objects selected by our context model are shown (yellow for support and red for co-occurrence violations).The first and the second rows show correctly identified out-of-context objects, and the third row shows failure cases.

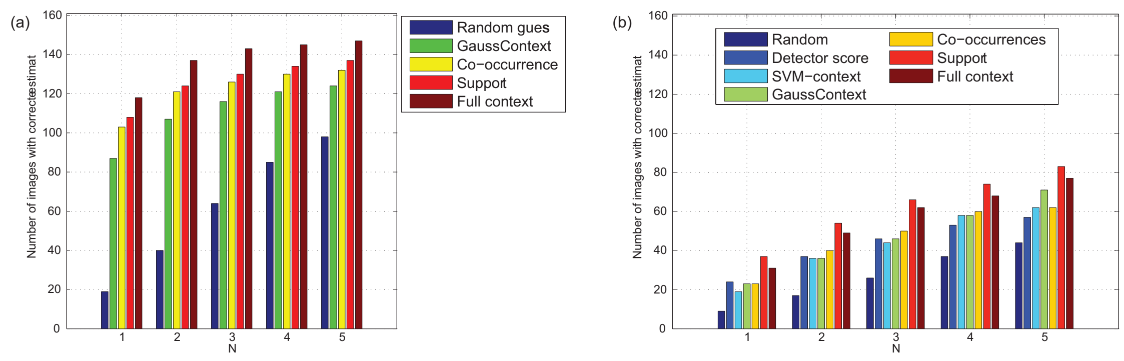


Figure 10: The number of images in which at least one out-of-context object is included in the set of N most unexpected objects estimated by our context model. (a) Using ground-truth labels and segmentations. (b) Using local detector outputs.