Learning Spatial Context:
Using Stuff to Find Things

Geremy Heitz Daphne Koller

Department of Computer Science, Stanford University
{gaheitz,koller}@cs.stanford.edu

Abstract. The sliding window approach of detecting rigid objects (such as cars) is predicated on the belief that the object can be identified from the appearance in a small region around the object. Other types of objects of amorphous spatial extent (e.g., trees, sky), however, are more naturally classified based on texture or color. In this paper, we seek to combine recognition of these two types of objects into a system that leverages "context" toward improving detection. In particular, we cluster image regions based on their ability to serve as context for the detection of objects. Rather than providing an explicit training set with region labels, our method automatically groups regions based on both their appearance and their relationships to the detections in the image. We show that our things and stuff (TAS) context model produces meaningful clusters that are readily interpretable, and helps improve our detection ability over state-of-the-art detectors. We also present a method for learning the active set of relationships for a particular dataset. We present results on object detection in images from the PASCAL VOC 2005/2006 datasets and on the task of overhead car detection in satellite images, demonstrating significant improvements over state-of-the-art detectors.

1 Introduction

Recognizing objects in an image requires combining many different signals from the raw image data. Figure 1 shows an example satellite image of a street scene, where we may want to identify all of the cars. From a human perspective, there are two primary signals that we leverage. The first is the local appearance in the window near the potential car. The second is our knowledge that cars appear on roads. This second signal is a form of contextual knowledge, and our goal in this paper is to capture this idea in a rigorous probabilistic model.

Recent papers have demonstrated that boosted object detectors can be effective at detecting monolithic object classes, such as cars [1] or faces [2]. Similarly, several works on multiclass segmentation have shown that regions in images can effectively be classified based on their color or texture properties [3]. These two lines of work have made significant progress on the problems of identifying "things" and "stuff," respectively. The important differentiation between these two classes of visual objects is summarized in Forsyth et al. [4] as:

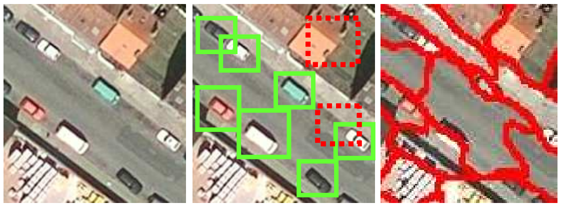


Fig. 1. (Left) An aerial photograph. (Center) Detected cars in the image (solid green = true detections, dashed red = false detections). (Right) Finding "stuff" such as buildings, by classifying regions, shown delineated by red boundaries.

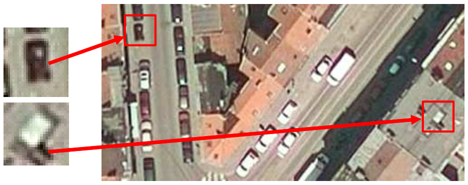


Fig. 2. Example detections from the satellite dataset that demonstrate context. Classifying using local appearance only, we might think that both windows at left are cars. However, when seen in context, the bottom detection is unlikely to be an actual car.

> The distinction between materials — "stuff" — and objects — "things" — is particularly important. A material is defined by a homogeneous or repetitive pattern of fine-scale properties, but has no specific or distinctive spatial extent or shape. An object has a specific size and shape.

Recent work has also shown that classifiers for both things or stuff can benefit from the proper use of contextual cues. The use of context can be split into a number of categories. *Scene-Thing context* allows scene-level information, such as the scale or the "gist" [5], to determine location priors for objects. *Stuff-Stuff context* captures the notion that sky occurs above sea and road below building [6]. *Thing-Thing context* considers the co-occurrence of objects, encoding, for example, that a tennis racket is more likely to occur with a tennis ball than with a lemon [7]. Finally *Stuff-Thing context* allows the texture regions (e.g., the roads and buildings in Figure 1) to add predictive power to the detection of objects (e.g., the cars in Figure 1). We focus on this fourth type of context. Figure 2 shows an example of this context in the case of satellite imagery.

In this paper, we present a probabilistic model linking the detection of things to the unsupervised classification of stuff. Our method can be viewed as an attempt to cluster "stuff," represented by coherent image regions, into clusters that are both visually similar and best able to provide context for the detectable "things" in the image. Cluster labels for the image regions are probabilistically linked to the detection window labels through the use of region-detection "relationships," which encode their relative spatial locations. We call this model the things and stuff (TAS) context model because it links these two components into a coherent whole. The graphical representation of the TAS model is depicted in Figure 3. At training time, our approach uses supervised (ground-truth) detection labels, without requiring supervised labels for the "stuff"-regions, and learns the model parameters using the Expectation-Maximization (EM) algorithm. Furthermore, we demonstrate a principled method for learning the set of active relationships (those used by the model). Our relationship selection through a variant of structural EM [8] is a novel aspect of our method, allowing the determination of which types of relationships are most appropriate without costly hand-engineering or cross-validation. At test time, these relationships are observed, and both the region labels and detection labels are inferred.

We present results of the TAS method on diverse datasets. Using the Pascal Visual Object Classes challenge datasets from 2005 and 2006, we utilize one of the top competitors as our baseline detector [9], and demonstrate that our TAS method improves the performance of detecting cars, bicycles and motorbikes in street scenes and cows and sheep in rural scenes (see Figure 5). In addition, we consider a very different dataset of satellite images from Google Earth, of the city and suburbs of Brussels, Belgium. The task here is to identify the cars from above; see Figure 2. For clarity, the satellite data is used as the running example throughout the paper, but all descriptions apply equally to the other datasets.

2 Related Work

The role of context in object recognition has become an important topic, due both to the psychological basis of context in the human visual system [10] and to the striking algorithmic improvements that "visual context" has provided [11].

The word "context" has been attached to many different ideas. One of the simplest forms is co-occurrence context. The work of Rabinovich et al. [7] demonstrates the use of this context, where the presence of a certain object class in an image probabilistically influences the presence of a second class. The context of Torralba et al. [11] assumes that certain objects occur more frequently in certain rooms, as monitors tend to occur in offices. While these methods achieve excellent results when many different object classes are labeled per image, they are unable to leverage unsupervised data for contextual object recognition.

In addition to co-occurrence context, many approaches take into account the spatial relationships between objects. At the descriptor level, Wolf et al. [12] detect objects using a descriptor with a large capture range, allowing the detection of the object to be influenced by surrounding image features. Because these methods use only the raw features, however, they cannot obtain a holistic view of an entire scene. This is analogous to addressing image segmentation using a wider feature window rather than a Markov random field (MRF). Similarly, Fink and Perona [13] use the output of boosted detectors for other classes as additional features for the detection of a given class. This allows the inclusion of signal beyond the raw features, but requires that all "parts" of a scene be supervised. Murphy et al. [5] use a global feature known as the "gist" to learn statistical priors on the locations of objects within the context of the specific scene. The gist descriptor is excellent at predicting large structures in the scene, but cannot handle the local interactions present in the satellite data, for example.

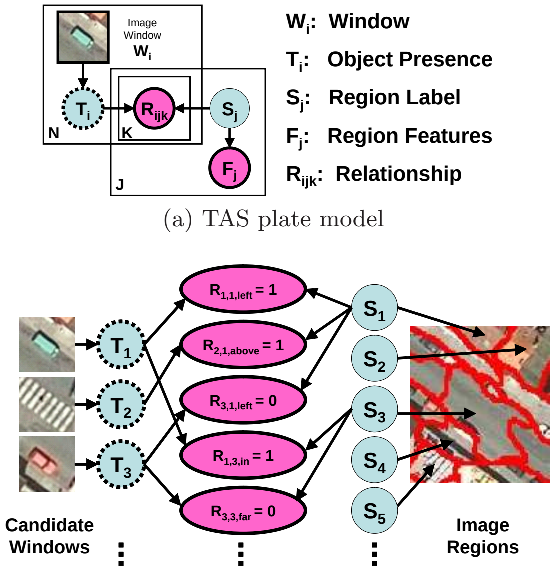
Another approach to modeling spatial relationships is to use a Markov Random Field (MRF) or variant (CRF,DRF) [14, 15] to encode the preferences for certain spatial relationships. These techniques offer a great deal of flexibility in the formulation of the affinity function and all the standard benefits of a graphical model formulation (e.g., well-known learning and inference techniques). Singhal et al. [6] also use similar concepts to the MRF formulation to aggregate decisions across the image. These methods, however, suffer from two drawbacks. First, they tend to require a large amount of annotation in the training set. Second, they put things and stuff on the same footing, representing both as "sites" in the MRF. Our method requires less annotation and allows detections and image regions to be represented in their (different) natural spaces.

Perhaps the most ambitious attempts to use context involves the attempt to model the scene of an image holistically. Torralba [1], for instance, uses global image features to "prime" the detector with the likely presence/absence of objects, the likely locations, and the likely scales. The work of Hoiem and Efros [16] takes this one level further by explicitly modeling the 3D layout of the scene. This allows the natural use of scale and location constraints (e.g., things closer to the camera are larger). Their approach, however, is tailored to street scenes, and requires domain-specific modeling. The specific form of their priors would be useless in the case of satellite images, for example.

3 Things and Stuff (TAS) Context Model

Our probabilistic context model builds on two standard components that are commonly used in the literature. The first is sliding window object detection, and the second is unsupervised image region clustering. A common method for finding "things" is to slide a window over the image, score each window's match to the model, and return the highest matching such windows. We denote the features in the i^{th} candidate window by W_i, the presence/absence of the target class in that window by T_i (T for "thing"), and assume that what we learn in our detector is a conditional probability $P(T_i \mid W_i)$ from the window features to the probability that the window contains the object; this probability can be derived from most standard classifiers, such as the highly successful boosting approaches [1]. The set of windows included in the model can, in principle, include all windows in the image. We, however, limit ourselves to looking at the top scoring detection windows according to the detector (i.e., all windows above some low threshold, chosen in order to capture most of the true positives).

The second component we build on involves clustering coherent regions of the image into groups based on appearance. We begin by segmenting the image into regions, known as superpixels, using the normalized cut algorithm of Ren and Malik [17]. For each region j, we extract a feature vector $\boldsymbol{F}_j$ that includes color and texture features. For our stuff model, we will use a generative model where



(a) TAS plate model

(b) TAS ground network

Fig. 3. The TAS model. The plate representation (a) gives a compact visualization of the model, which unrolls into a "ground" network (b) for any particular image. Nodes with dotted lines represent variables that are unobserved during training; only the pink nodes (which represent image features) are observed during testing.

each region has a hidden class, and the features are generated from a Gaussian distribution with parameters depending on the class. Denote by S_j (S for "stuff") the (hidden) class label of the j^{th} region. We assume that the features are derived from a standard naive Bayes model, where we have a probability distribution $P(\boldsymbol{F}_j \mid S_j)$ of the image features given the class label.

In order to relate the detector for "things" (T's) to the clustering model over "stuff" (S's), we need to develop an intuitive representation for the relationships between these units. Human knowledge in this area comes in sentences like "cars drive on roads," or "cars park 20 feet from buildings." We introduce indicator variables R_{ijk} that indicate whether candidate detection i and region j have relationship k. The different k's represent different relationships, such as: "detection i is *in* region j" ($k = 1$), or "detection i is about 100 pixels away from region j" ($k = 2$). For now we assume the set of relationships (the meaning of R_{ijk} for each k) is known, and in Section 4 we describe how this set of relationships is learned from the data.

We can now link our component models into a single coherent probabilistic things and stuff (TAS) model, as depicted in the plate model of Figure 3(a). Probabilistic influence flows between the detection window labels and the image region labels through the v-structures that are activated by observing the relationship variables. For a particular input image, this plate model unrolls into a "ground" network that encodes a distribution over the detections and regions of the image. Figure 3(b) shows a toy example of a partial ground network for the image of Figure 1. It is interesting to note the similarities between TAS and the MRF approaches in the literature. In effect, the relationship variables link the detections and the regions into a probabilistic web where signals at one point in the web influence a detection in another part of the image, which in turn influences the region label in yet another location. Although similar to an MRF in its ability to define a holistic view of the entire image, our model is generative, allowing us to train very efficiently, even when the S_j variables are unobserved.

All variables in the TAS model are discrete except for the feature variables $\boldsymbol{F}_j$. This allows for simple table conditional probability distributions (CPDs) for all discrete nodes in this Bayesian network. The probability distribution over these variables decomposes according to:

$$P(\boldsymbol{T}, \boldsymbol{S}, \boldsymbol{F}, \boldsymbol{R} \mid \boldsymbol{W}) = \prod_i P(T_i \mid W_i) \prod_j P(S_j) P(\boldsymbol{F}_j \mid S_j) \prod_{ijk} P(R_{ijk} \mid T_i, S_j).$$

We note that TAS is very modular. It allows us to "plug in" any detector and any generative model for regions (e.g., [3]).

4 Learning and Inference in the TAS Model

Because TAS unrolls into a Bayesian network for each image, we can use standard learning and inference methods. In particular, we learn the parameters of our model using the Expectation-Maximization (EM) [18] algorithm and perform inference using Gibbs sampling, a standard variant of MCMC sampling [19]. Furthermore, we show how to learn the set of active relationships from a large candidate relationship pool using a structure search interleaved with the EM [8].

Learning the Parameters with EM. At learning time, we have a set of images with annotated labels for our target object class(es). We first train the base detectors using this set, and select as our candidate windows all detections above a threshold; we thus obtain a set of candidate detection windows $W_1 \ldots W_N$ along with their ground-truth labels $T_1 \ldots T_N$. We also have a set of regions and a feature vector F_j for each, and an R_{ijk} relationship variable for every window-region pair and relationship type, which is observed for each pair based on their relative spatial relationship. Our goal is to learn parameters for the TAS model.

Because our variables are discrete and have relatively small cardinality, we can learn our model using the EM algorithm. EM iterates between using probabilistic inference to derive a soft completion of the hidden variables (E-step) and finding maximum-likelihood parameters relative to this soft completion (M-step). The E-step here is particularly easy: at training time, only the S_j's are unobserved; moreover, because the T variables are observed, the S_j's are conditionally independent of each other. Thus, the inference step turns into a simple computation for each S_j separately, a process which can be performed in linear

```
Algorithm LearnTAS
Input:    Candidate relationships $\mathcal{C}$, Dataset $\mathcal{D} = \{(\boldsymbol{W}[m], \boldsymbol{T}[m], \boldsymbol{F}[m], \boldsymbol{R}[m])\}$
    $\mathcal{R} \leftarrow \emptyset$ (all relationships "inactive")
    Repeat until convergence
        Repeat until convergence (EM over Parameters)
            E-step: $Q[m] \leftarrow P(\boldsymbol{S} \mid \boldsymbol{T}, \boldsymbol{F}, \boldsymbol{R}; \theta_{\mathcal{R}}) \quad \forall m$
            M-step: $\theta_{\mathcal{R}} \leftarrow \text{argmax} E_Q \left[\sum_m \ell(\boldsymbol{S}, \boldsymbol{T}, \boldsymbol{F}, \boldsymbol{R} \mid \boldsymbol{W}; \theta_{\mathcal{R}})\right]$
        Repeat until convergence (Greedy Structure Search)
            Forall $k$, $score_k = \sum_m \ell(\boldsymbol{T} \mid \boldsymbol{F}, \boldsymbol{R}; \theta_{\mathcal{R} \oplus k})$ (score with $k$ "activated")
            $\mathcal{R} \leftarrow \mathcal{R} \oplus k^*$    where $k^* = \text{argmax } score_k$
Return    Set $\mathcal{R}$ of "active" relationships, TAS parameters $\theta_{\mathcal{R}}$
```

Fig. 4. Learning a TAS model. Here ℓ represents the log-likelihood of the data, and $\oplus$ represents the set exclusive-or operation.

time. The M-step for table CPDs can be performed easily in closed form. To provide a good starting point for EM, we initialize the cluster assignments using the K-means algorithm. EM is guaranteed to converge to a local maximum of the likelihood function of the observed data.

Learning the Relationships. So far, we have assumed a known set of relationships. However, because different data may require different types of contextual relationships, we would like to learn which to use. We begin by defining a large set $\mathcal{C}$ of "candidate relationships" (i.e., all possible relationships that we want to consider for inclusion). For instance, we may include both "above by 100 pixels," and "above by 200 pixels" in $\mathcal{C}$, even though we believe only one of these will be chosen. Our goal is to search through $\mathcal{C}$ for the subset of relationships that will best facilitate the use of context. We denote this "active" set by $\mathcal{R}$.

Intuitively, if a particular type of relationship (e.g., , "above by 100 pixels") is "inactive" then we want to force the value of the corresponding R_{ijk} variables to be independent of the value of the T_i's and S_j's. In the language of Bayesian networks, we can achieve this by removing the edges from all T_i and S_j variables to the R_{ijk} variables for this particular k. With this view of "activating" relationships by including the edges in the Bayesian Network, we can formulate our search for $\mathcal{R}$ as a structure learning problem. To learn this network structure, we turn to the method of structural EM [8]. In particular, if we are considering K possible relationships, there are 2^K possible subsets to consider (each relationship can be "active" or "inactive"). We search this space using a greedy hill-climbing approach that is interleaved with the EM parameter learning. The hill-climbing begins with an empty set of relationships ($\mathcal{R} = \emptyset$), and adds or removes relationships one at a time until a local maximum is reached.

Standard structural EM scores each network using the log probability of the expected data, which is easily computed from the output of the E-step above. However, because our final goal is to correctly classify the "things," we would rather score each structure using the log probability of the T_i's. While this requires us to perform inference (described below) for each candidate structure, it is both theoretically more sound and produced far better results than the standard score. In order to avoid overfitting, we initially used a BIC penalty, but found that this resulted in too few “active” relationships. Instead, in experiments below, we include a Gaussian prior over the number of active relationships. Our learning process outputs an active set of relationships, $\mathcal{R}$, and the parameters of the TAS model for that set, $\theta_{\mathcal{R}}$. The algorithm is outlined in Figure 4.

Inference with Gibbs Sampling. At test time, our system must determine which windows in a new image contain the target object. We observe the candidate detection windows (W_i's, extracted by thresholding the base detector output), the features of each image region ($\boldsymbol{F}_j$'s), and the relationships (R_{ijk}'s). Our task is to find the probability that each window contains the object:

$$P(\boldsymbol{T} \mid \boldsymbol{F}, \boldsymbol{R}, \boldsymbol{W}) = \sum_{\boldsymbol{S}} P(\boldsymbol{T}, \boldsymbol{S} \mid \boldsymbol{F}, \boldsymbol{R}, \boldsymbol{W}) \tag{1}$$

Unfortunately, this expression involves a summation over an exponential set of values for the $\boldsymbol{S}$ vector of variables. We solve the inference problem approximately using a Gibbs sampling [19] MCMC method. We begin with some assignment to the variables. Then, in each Gibbs iteration we first resample all of the S's and then resample all the T's according to the following two probabilities:

$$P(S_j \mid \boldsymbol{T}, \boldsymbol{F}, \boldsymbol{R}, \boldsymbol{W}) \propto P(S_j) P(F_j \mid S_j) \prod_{ik} P(R_{ijk} \mid T_i, S_j) \tag{2}$$

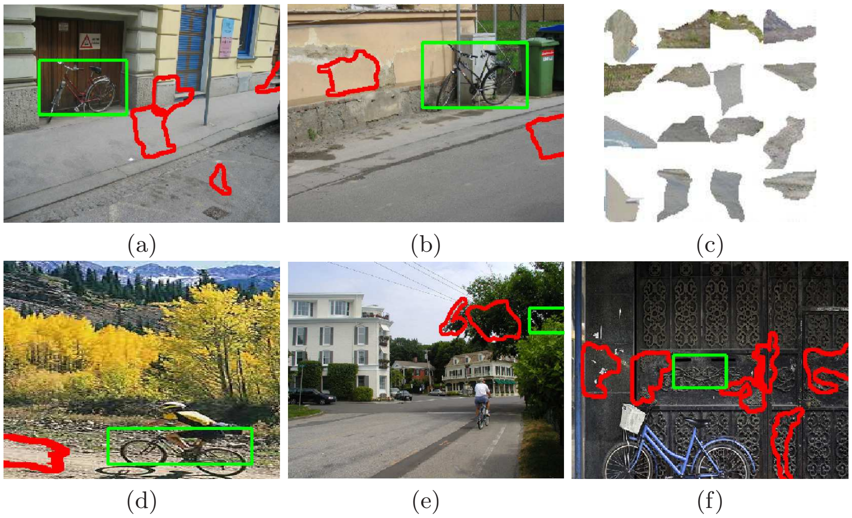
$$P(T_i \mid \boldsymbol{S}, \boldsymbol{F}, \boldsymbol{R}, \boldsymbol{W}) \propto P(T_i \mid W_i) \prod_{jk} P(R_{ijk} \mid T_i, S_j). \tag{3}$$

These sampling steps can be performed efficiently, as the T_i variables are conditionally independent given the S's and the S_j's are conditionally independent given the T's. In the last Gibbs iteration for each sample, rather than resampling $\boldsymbol{T}$, we compute the posterior probability over $\boldsymbol{T}$ given our current $\boldsymbol{S}$ samples, and use these distributional particles for our estimate of the probability in (1).

5 Experimental Results

In order to evaluate the TAS model, we perform experiments on three datasets. The first two are from the PASCAL Visual Object Classes challenges 2005 and 2006[20]. The scenes are urban and rural, indoor and outdoor, and there is a great deal of scale variation amongst the objects. The third is a set of satellite images acquired from Google Earth, with the goal of detecting cars. Because of the impoverished visual information, there are many false positives when a sliding window detector is applied. In this case, context provides a filtering mechanism to remove the false positives. Because these two applications are different, we use different detectors for each. We allow S a cardinality of $|S| = 10$,[1] and use 44 features for the image regions (color, texture, shape, etc. [21]).

[1] Results were robust to a range of $|S|$ between 5 and 20.



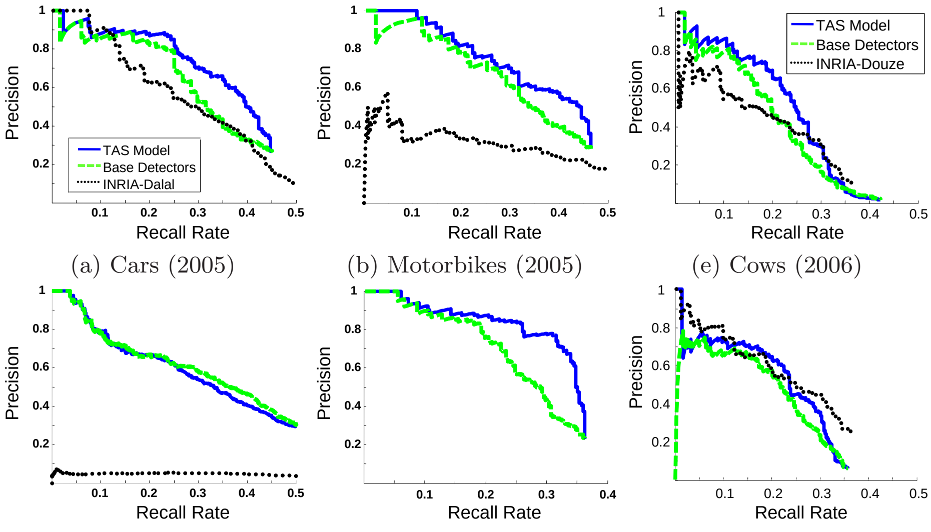
(a) (b) (c)

(d) (e) (f)

Fig. 5. (a,b) Example training detections from the bicycle class, with detection windows outlined by the green rectangles. The image regions with active relationships to the detection window are outlined in red. (c) 16 of the most representative regions for cluster #3. This cluster corresponds to "roads" or "bushes" as things that are gray/green and occur near cars. (d) A case where context helped find a true detection. (e,f) Two examples where incorrect detections are filtered out by context.

PASCAL VOC Datasets. For these experiments, we used four classes from the VOC2005 data, and two classes from the VOC2006 data. The VOC2005 dataset consists of 2232 images, manually annotated with bounding boxes for four image classes: cars, people, motorbikes, and bicycles. We use the "train+val" set (684 images) for training, and the "test2" set (859 images) for testing. The VOC2006 dataset contains 5304 images, manually annotated with 12 classes, of which we use the cow and sheep classes. We train on the "trainval" set (2618 images) and test on the "test" set (2686 images). To compare with the results of the challenges, we adopted as our detector the HOG (histogram of oriented gradients) detector of Dalal and Triggs [9]. This detector uses an SVM and therefore outputs a score $\text{margin}_i \in (-\infty, +\infty)$, which we convert into a probability by learning a logistic regression function for $P(T_i \mid \text{margin}_i)$. We also plot the precision-recall curve using the code provided in the challenge toolkit.

We learned our model with a set of 25 candidate relationships that included regions within the window, at offsets in eight directions (every 45 degrees) at two different distances, and the union of various combinations of features (e.g., R_{24} indicates regions to the right *or* left of the window by one bounding box). Figure 5 (top row) shows example bicycle detection candidates, and the related image regions, suggesting the type of context that might be learned. For example, the region beside and below both detections (outlined in red) belongs to



(a) Cars (2005) (b) Motorbikes (2005) (e) Cows (2006)

(c) People (2005) (d) Bicycles (2005) (f) Sheep (2006)

Object Class	Base AP	TAS AP (Fixed R)	TAS AP (Learned R)	Improvement (TAS - Base)
Cars	0.325	0.360	**0.363**	0.038
Motorbikes	0.341	**0.390**	0.373	0.032
People	**0.346**	0.346	0.337	-0.009
Bicycles	0.281	0.310	**0.325**	0.044
Cows	0.224	0.241	**0.258**	0.034
Sheep	0.206	0.233	**0.248**	0.042

Fig. 6. (top) Precision-recall (PR) curves for the VOC classes. (bottom) Average precision (AP) scores for each experiment. AP is a robust variant of the area under the PR curve. We show the AP score for TAS with hand-selected relationships as well as with learned relationships, with results from the best performing model in bold.

cluster #3, which looks visually like a road or bush cluster (see Figure 5(c)). The learned values of the model parameters also indicate that being to the left or right of this cluster increases the probability of a window containing a bicycle (e.g., by about 33% in the case where $R_{ijk} = 1$ for this relationship).

We performed a single run of EM learning with structure search to convergence, which takes 1-2 hours on an Intel Dual Core 1.9 GHz machine with 2 GB of memory. We run separate experiments for each class, though in principle it would be possible to learn a single joint model over all classes. By separating the classes, we are able to isolate the contextual contribution from the stuff, rather than between the different types of things present in the images. For our MCMC inference, we found that, due to the strength of the baseline detectors, the Markov chain converged fairly rapidly; we achieved very good results using merely 10 MCMC samples, where each is initialized randomly and then undergoes 5 Gibbs iterations. Inference takes about 0.5 seconds per image.

The bottom row of Figure 5 shows some detections that were corrected using context. We show one example where a true bicycle was discovered using context, and two examples where false positives were filtered out by our model. These examples demonstrate the type of information that is being leveraged by TAS. In the first example, the dirt road to the left of the window gives a signal that this detection is at ground level, and is therefore likely to be a bicycle.

Figure 6 shows the full recall-precision curve for each class. For (a-d) we compare to the 2005 **INRIA-Dalal** challenge entry, and for (e,f) we compare to the 2006 **INRIA-Douze** entry, both of which used the HOG detector. We also show the curve produced by our **Base Detector** alone. [2] Finally, we plot the curves produced by our **TAS Model**, trained using full EM, which scores windows using the probability of (1). From these curves, we see that the TAS model provided an improvement in accuracy for all but the "people" class. We believe the lack of improvement for people is due to the wide variation of backgrounds in these images, providing no strong context cues to latch onto. Furthermore, the base HOG detector was in fact originally optimized to detect people.

Satellite Images. The second dataset is a set of 30 images extracted from Google Earth. The images are color, and of size 792 × 636, and contain 1319 manually labeled cars. The average car window is approximately 45 × 45 pixels, and all windows are scaled to these dimensions for training. We used 5-fold cross-validation, and results below report the mean performance across the folds.

Here, we use a patch-based boosted detector very similar to that of Torralba [1]. We use 50 rounds of boosting with two level decision trees over patch cross-correlation features that were computed for 15,000–20,000 rectangular patches (intensity and gradient) of various aspect ratios and widths of 4-22 pixel. As above, we convert the boosting score into a probability using logistic regression. For training the TAS model, we used 10 random restarts of EM, selecting the parameters that provided the best likelihood of the observed data. For inference, we need to account for the fact that our detectors are much weaker, and so more samples are necessary to adequately capture the posterior. We utilize 20 samples per image, where each sample undergoes 20 iterations.

Figure 7 shows some learned "stuff" clusters. Eight of the ten learned clusters are shown, visualized by presenting 16 of the image regions that rank highest with respect to $P(\boldsymbol{F} \mid S)$. These clusters have a clear interpretation: cluster #4, for instance, represents the roofs of houses and cluster #6 trees and water regions. With each cluster, we also show the odds-ratio of a candidate window containing a car given that it is in this region. Clusters #7 and #8 are road clusters, and increase the chance of a nearby window being a car by a factor of 2 or more. Clusters #1 and #6, however, which represent forest and grass areas, decrease the probability of nearby candidates being cars by factors of 9 or

[2] Differences in PR curves between our base detector and the **INRIA-Dalal/INRIA-Douze** results come from the use of slightly different training windows and parameters. **INRIA-Dalal** did not report results for "bicycle."

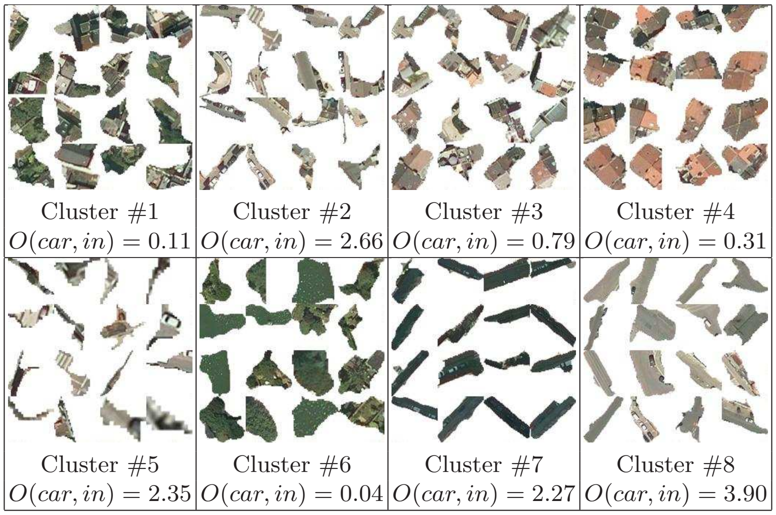


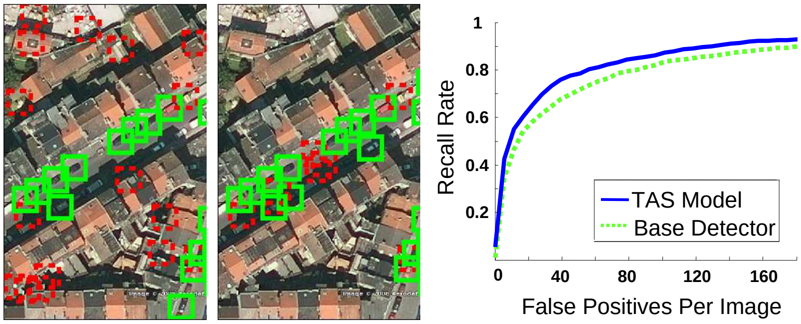
Fig. 7. Example clusters learned by the context model on the satellite dataset. Each cluster shows 16 of the training image regions that are most likely to be in the cluster based on $P(\boldsymbol{F} \mid S)$. For each cluster, we also show the odds-ratio of a window "in" a region labeled by that cluster containing a car ($O(car, in) = P(in|car, q)/P(in|no\ car, q)$). A higher odds-ratio indicates that this contextual relationship increases the model's confidence that the window contains a car.

more. Figure 8 shows an example with the detections of the detector alone and of the TAS model, which filters out many of the false positives that are not near roads. Because there are many detections per image, we plot the recall versus the number of false detections per image in Figure 8(c). The **Base Detectors** are compared to the **TAS Model**, verifying that context indeed improves our results, by filtering out many of the false positives.

6 Discussion and Future Directions

In this paper, we have presented the TAS model, a probabilistic framework that captures the contextual information between "stuff" and "things", by linking discriminative detection of objects with unsupervised clustering of image regions. Importantly, the method does not require extensive labeling of image regions; standard labeling of object bounding boxes suffices for learning a model of the appearance of stuff regions and their contextual cues. We have demonstrated that the TAS model improves the performance even of strong base classifiers, including one of the top performing detectors in the PASCAL challenge.

The flexibility of the TAS model provides several important benefits. The model can accommodate almost any choice of object detector that produces a score for candidate windows. It is also flexible to any generative model over



(a) Base Detectors (b) TAS Detections (c) PR Curve

Fig. 8. Example image, with detections found by the base detector (a), and by the TAS model (b) with a threshold of 0.15. The TAS model filters out many of the false positives far away from roads. (c) shows a plot of recall rate vs. false positives per image for the satellite data. The results here are averaged across 5 folds, and show a significant improvement from using TAS over the base detectors.

any type of region features. For instance, we might pre-cluster the regions into visual words, and then use a multinomial distribution over these words [21]. Additionally, because our model discovers which relationships to use, our method has the ability to discover spatial interactions that are not already known to the modeler. Indeed, automated structure-learning such as the one we employ here can provide a valuable substitute for the laborious process of manual feature construction that is necessary for engineering a computer vision system.

Because the image region clusters are learned in an unsupervised fashion, they are able to capture a wide range of possible concepts. While a human might label the regions in one way (say trees and buildings), the automatic learning procedure might find a more contextually relevant grouping. For instance, the TAS model might split buildings into two categories: apartments, which often have cars parked near them, and factories, which rarely co-occur with cars.

As discussed in Section 2, recent work has amply demonstrated the importance of context in computer vision. The context modeled by the TAS framework is a natural complement for many of the other types of context in the literature. In particular, while many other forms of context can relate known objects that have been labeled in the data, our model can extract the signals present in the unlabeled part of the data. However, a major limitation of the TAS model is that it captures only 2D context. This issue also affects our ability to determine the appropriate scale for the contextual relationships. It would be interesting to integrate a TAS-like definition of context into an approach that attempts some level of 3D reconstruction, such as the work of Hoiem and Efros [16] or of Saxena et al. [22], allowing us to utilize 3D context and address the issue of scale.

References

1. Torralba, A.: Contextual priming for object detection. IJCV **53**(2) (2003)
2. Viola, P., Jones, M.: Robust real-time face detection. In: ICCV. (2001)
3. Shotton, J., Winn, J., Rother, C., Criminisi, A.: Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In: ECCV. (2006)
4. Forsyth, D.A., Malik, J., Fleck, M.M., Greenspan, H., Leung, T.K., Belongie, S., Carson, C., Bregler, C.: Finding pictures of objects in large collections of images. In: Object Representation in Computer Vision. (1996)
5. Murphy, K., Torralba, A., Freeman, W.: Using the forest to see the tree: a graphical model relating features, objects and the scenes. In: NIPS. (2003)
6. Singhal, A., Luo, J., Zhu, W.: Probabilistic spatial context models for scene content understanding. In: CVPR. (2003)
7. Rabinovich, A., Vedaldi, A., Galleguillos, C., Wiewiora, E., Belongie, S.: Objects in context. In: ICCV. (2007)
8. Friedman, N.: Learning belief networks in the presence of missing values and hidden variables. In: ICML. (1997)
9. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR. (2005)
10. Oliva, A., Torralba, A.: The role of context in object recognition. Trends Cogn Sci (2007)
11. Torralba, A., Murphy, K., Freeman, W., Rubin, M.: Context-based vision system for place and object recognition. In: ICCV. (2003)
12. Wolf, L., Bileschi, S.: A critical view of context. IJCV **69**(2) (2006)
13. Fink, M., Perona, P.: Mutual boosting for contextual inference. In: NIPS. (2003)
14. Kumar, S., Hebert, M.: A hierarchical field framework for unified context-based classification. In: ICCV. (2005)
15. Carbonetto, P., de Freitas, N., Barnard, K.: A statistical model for general contextual object recognition. In: ECCV. (2004)
16. Hoiem, D., Efros, A.A., Hebert, M.: Putting objects in perspective. In: CVPR. (2006)
17. Ren, X., Malik, J.: Learning a classification model for segmentation. In: ICCV. (2003)
18. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society. Series B (Methodological) **39**(1) (1977)
19. Geman, S., Geman, D.: Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. (1987)
20. Everingham, M.: The 2005 pascal visual object classes challenge. In: MLCW. (2005)
21. Barnard, K., Duygulu, P., Forsyth, D., de Freitas, N., Blei, D.M., Jordan, M.I.: Matching words and pictures. JMLR **3** (2003)
22. Saxena, A., Sun, M., Ng, A.Y.: Learning 3-d scene structure from a single still image. In: CVPR. (2007)