Using the forest to see the trees: exploiting context for visual object detection and localization

A. Torralba
Massachusetts Institute of Technology
32 Vassar
Cambridge, USA
torralba@csail.mit.edu

K. P. Murphy
Department of Computer Science
University of British Columbia
Vancouver BC V6T 1Z4
murphyk@cs.ubc.ca

W. T. Freeman
Massachusetts Institute of Technology
32 Vassar
Cambridge, USA
billf@csail.mit.edu

ABSTRACT

Recognizing objects in images is an active area of research in computer vision. In the last two decades, there has been much progress and there are already object recognition systems operating in commercial products. However, most of the algorithms for detecting objects perform an exhaustive search across all locations and scales in the image comparing local image regions with an object model. That approach ignores the semantic structure of scenes and tries to solve the recognition problem by brute force. In the real world, objects tend to co-vary with other objects, providing a rich collection of contextual associations. These contextual associations can be used to reduce the search space by looking only in places in which the object is expected to be; this also increases performance, by rejecting patterns that look like the target but appear in unlikely places.

Most modeling attempts so far have defined the context of an object in terms of other previously recognized objects. The drawback of this approach is that inferring the context becomes as difficult as detecting each object. An alternative view of context relies on using the entire scene information holistically. This approach is algorithmically attractive since it dispenses with the need for a prior step of individual object recognition. In this paper we use a probabilistic framework for encoding the relationships between context and object properties and we show how an integrated system provides improved performance. We view this as a significant step towards general purpose machine vision systems.

1. INTRODUCTION

Visual object detection, such as finding cars and people in images, is an important but challenging task. It is important because of its inherent scientific interest (understanding how to make machines see may shed light on biological vision), and because it is useful for many applications, such as content-based image retrieval, robotics, etc. It is challenging because the appearance of objects can vary a lot from instance to instance, and from image to image, due to factors such as variation in pose, lighting, style, articulation, occlusion, low quality imaging, etc.

Over the last two decades, much progress has been made in visual object detection using machine learning techniques. Most of these approaches rely on using supervised learning to train a classifier to distinguish between instances of the object class and the background. The trained classifier is then applied to thousands of small overlapping patches or windows of each test image, and the locations of the high-confidence detections are returned. The features computed inside each patch are usually the outputs of standard image processing operations, such as a histogram of responses to Gabor filters at different scales and orientations. The classifiers themselves are standard supervised learning models such as SVMs, neural networks or boosted decision stumps [20].

This "sliding window classifer" technique has been quite successful in certain domains such as detecting cars, pedestrians and faces. Indeed most contemporary digital cameras emply such a technique to detect faces, which they use to set the auto-focus. Also, some cars now come equipped with pedestrian detection systems based on similar principles.

One major problem with the standard approach is that even a relatively low false-positive rate per class can be unacceptable when there are many classes or categories. For example, if each detector generates about 1 false alarm every 10 images, and there are 1000 classes, we will have 100 false alarms per image. An additional problem is that running every detector on every image can be slow. These are both fundamental obstacles to building a general purpose vision system.

One reason for the relatively high false alarm rate of standard approaches is that most object detection systems are "myopic", in the sense that they only look at local features of the image. One possible remedy is to leverage global features of the image, and to use these to compute the "prior" probability that each object category is present, and if so, its likely location and scale. Previous work (e.g., [17]) has shown that simple global image features, known as the "gist" of the image, are sufficient to provide robust predictions about the presence and location of different object categories. Such features are fast to compute, and provide information that is useful for many classes and locations simultaneously.

In this paper, which is an extension of our previous work [17, 9, 8], we present a simple approach for combining stan-

An early version of this paper, entitled "Using the forest to see the trees: a graphical model relating features, objects and scenes", was published in *Neural Information Processing Systems*, 2003, MIT Press. Ref. [9].

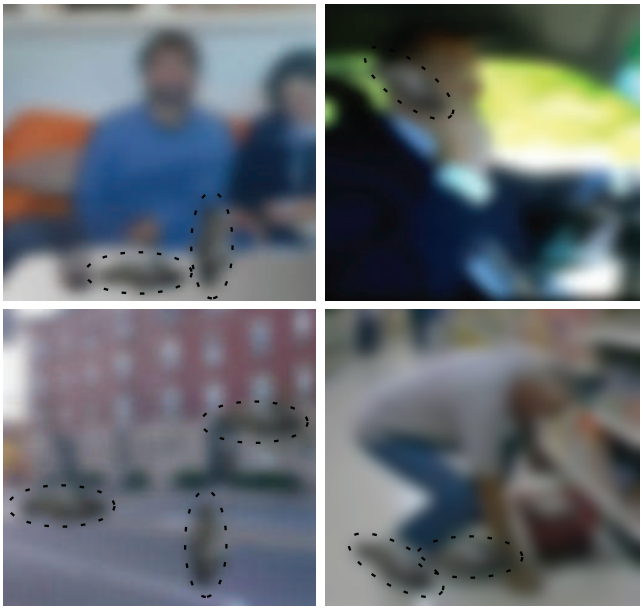

Figure 1: In presence of image degradation (e.g. blur), object recognition is strongly influenced by contextual information. The visual system makes assumptions regarding object identities based on its size and location in the scene. In these images, the same black blob can be interpreted as a plate, bottle, cell phone, car, pedestrian or shoe, depending on the context. (Each circled blob has identical pixels, but in some cases has been rotated.)

dard sliding-window object detection systems, which use local, "bottom up" image features, with systems that predict the presence and location of object categories based on global, or "top-down", image features. These global features serve to define the context in which the object detection is happening. The importance of context is illustrated in Figure 1, which shows that the same black "blob", when placed in different surroundings, can be interpreted as a plate or bottle on the table, a cell phone, a pedestrian or car, or even a shoe. Another example is shown in Figure 2: it is easy to infer that there is very probably a computer monitor behind the blacked out region of the image.

We are not the first to point out the importance of context in computer vision. For example, Strat and Fischler emphasized its importance in their 1991 paper [16]. However, there are two key differences between our approach and previous work. First, in early work, such as [16], the systems consist of hand-engineered if-then rules, whereas more recent systems rely on statistical models that are fit to data. Second, most other approaches define the context in terms of other objects [6, 14, 18, 13]; but this introduces a chicken-and-egg problem: to detect an object of type 1 you first have to detect an object of type 2. By contrast, we propose a hierarchical approach, in which we define the context in terms of an overall scene category. This can be reliably inferred using global images features. Conditioned on the scene category, we assume that objects are independent. While not strictly true, this results in a simple yet effective approach, as we will show below.

In the following sections, we describe the different components of our model. We will start by showing how we can represent contextual information without using objects as an intermediate representation. Then we will show how that representation can be integrated with an object detector.

Figure 2: What is hidden behind the mask? In this example, context is so strong that one can reliably infer that the hidden object is a computer monitor.

2. GLOBAL IMAGE FEATURES: THE GIST OF AN IMAGE

In the same way that an object can be recognized without decomposing it into a set of nameable parts (e.g., the most successful face detectors do not try to detect the eyes and mouth first, instead they search for less semantically meaningful features), scenes can also be recognized without necessarily decomposing them into objects. The advantage of this is that it provides an additional source of information that can be used to provide contextual information for object recognition. As suggested in [10, 11] it is possible to build a global representation of the scene that bypasses object identities, in which the scene is represented as a single entity. Recent work in computer vision has highlighted the importance of global scene representations for scene recognition [11, 1, 7] and as a source of contextual information [17, 9, 3]. These representations are based on computing statistics of low level features (similar to representations available in early visual areas such as oriented edges, vector quantized image patches, etc.) over fixed image regions. One example of a global image representation is the gist descriptor [11]. The gist descriptor is a vector of features g, where each individual feature g_k is computed as:

$$g_k = \sum_{x,y} w_k(x,y) \times |I(x,y) \otimes h_k(x,y)|^2 \qquad (1)$$

where $\otimes$ denotes image convolution and $\times$ is a pixel-wise multiplication. $I(x,y)$ is the luminance channel of the input image, $h_k(x,y)$ is a filter from a bank of multiscale oriented Gabor filters (6 orientations and 4 scales), and $w_k(x,y)$ is

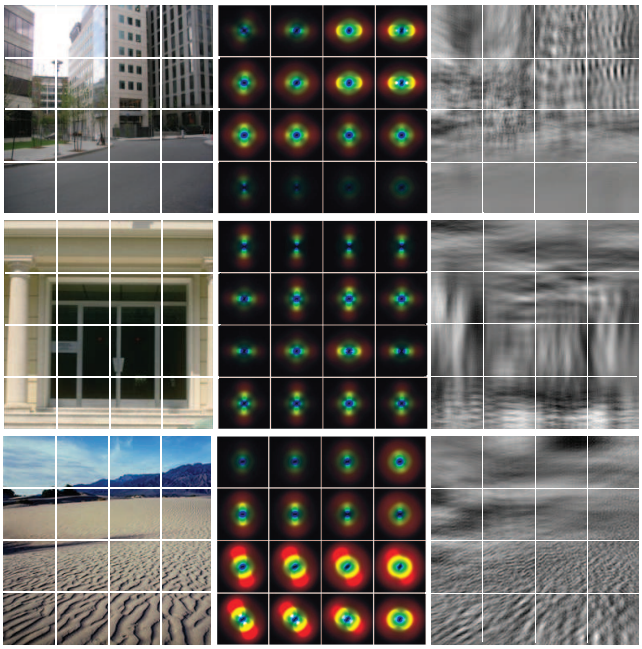

Figure 3: This figure illustrates the information encoded by the gist features for three different images. See text for details.

a spatial window that will compute the average output energy of each filter at different image locations. The windows $w_k(x, y)$ divide the image in a grid of 4×4 non-overlapping windows. This results in a descriptor with a dimensionality of $4 \times 4 \times 6 \times 4 = 384$.

Figure 3 illustrates the amount of information preserved by the gist descriptor. The middle column shows the average of the output magnitude of the multiscale-oriented filters on a polar plot (note that the orientation of each plot is orthogonal to the direction of the edges in the image). The average response of each filter is computed locally by splitting the image into 4×4 windows. Each different scale is color coded (red for high spatial frequencies, and blue for the low spatial frequencies), and the intensity is proportional to the energy for each filter output. In order to illustrate the amount of information preserved by this representation, the right column of figure 3 shows noise images that are coerced to have the same gist features as the target image, using the texture synthesis method of [2]. As shown in figure 3, the gist descriptor provides a coarse description of the textures present in the image and their spatial organization. The gist descriptor preserves relevant information needed for categorizing scenes into categories (e.g., classifying an image as being a beach scene, a street or a living-room). As reported in [12], when trying to discriminate across 15 different scene categories, the gist descriptor classifies correctly 75 % of the images. Recognizing the scene depicted by a picture is an important task on its own, but in addition it can be used to provide strong contextual priors as we will discuss in the next section.

3. JOINT SCENE CLASSIFICATION AND OBJECT DETECTION

In this section, we describe our approach in more detail. In Section 3.1, we briefly describe the standard approach to object detection and localization using local features. In Sections 3.3 and 3.2 we describe how to use global features for object localization and detection respectively. In Section 3.4 we discuss how to integrate these local and global features. A comparison of the performance of local and global features is deferred until Section 4.

3.1 Object presence detection and localization using local features

In our previous paper [9], we considered detecting four different types or classes of objects: cars, people, keyboards and screens (computer monitors). In this paper, we will mostly focus on cars, for brevity. We use a subset of the LabelMe dataset [11, 15] for training and testing (details are in Section 4).

There are two tasks that we want to address: object presence detection (where the goal is to predict if the object is present or absent in the image, i.e., to answer the question: is there any car in this image?) and object localization (where the goal is to precisely locate all the instances of an object class within each image). Solving the object presence detection task can be done even if the object localization is not accurate.

We can formalize the object presence detection and localization problem as follows. Let $P^t = 1$ if one or more objects of type t are present anywhere in the image, and $P^t = 0$ otherwise. The goal of object *presence detection* is to estimate the probability $p(P^t = 1|I)$, where I is the image. Later we will generalize this slightly by trying to estimate the number of instances of the object class that might be present, $p(N^t|I)$, where $N^t \in \{0, 1, 2, 3-5, 5-10, > 10\}$. We call this object *counting*.

The goal of object *localization* is to specify the location and size of each of the object instances. More precisely, let O_i^t be a binary random variable representing whether image patch i contains an object of type t or not, for $i \in \{1, \ldots, N\}$, where $N \sim 1000$ is the number of image patches. (The size and shape of the image patches varies according to the object type; for side views of cars, we use patches of size 30×80; to handle cars of different sizes, we apply the technique to multiple versions of the image at different scales.) One way to perform localization is to compute the log-likelihood ratio

$$c_i^t = \log p(f_i^t|O_i^t = 1)/p(f_i^t|O_i^t = 0), \quad (2)$$

for each i and t, and then to return all the locations where this log likelihood ratio is above some threshold. Here f_i^t is a set of local features extracted from image I at patch i for class t. The details of the features and classifier that we used can be found in [19].

For simplicity, in this paper we select the D most confident detections (after performing local non-maximum suppression); let their locations be denoted by ℓ_i^t, for $i \in \{1, \ldots, D\}$. Figure 6(a) gives an illustration of the output of our system on a typical image. For the results in this paper, we set $D = 10$ so that no correct detections are discarded and still small enough to be efficient. In the figure we show the top $D = 4$ detections to avoid clutter. The locations of each detection ℓ_i^t are indicated by the position and scale of the box, and their confidences c_i^t are indicated by the thickness of the border. In Figure 6(b-top), we see that although the system has detected the car, it has also detected 3 false positives.

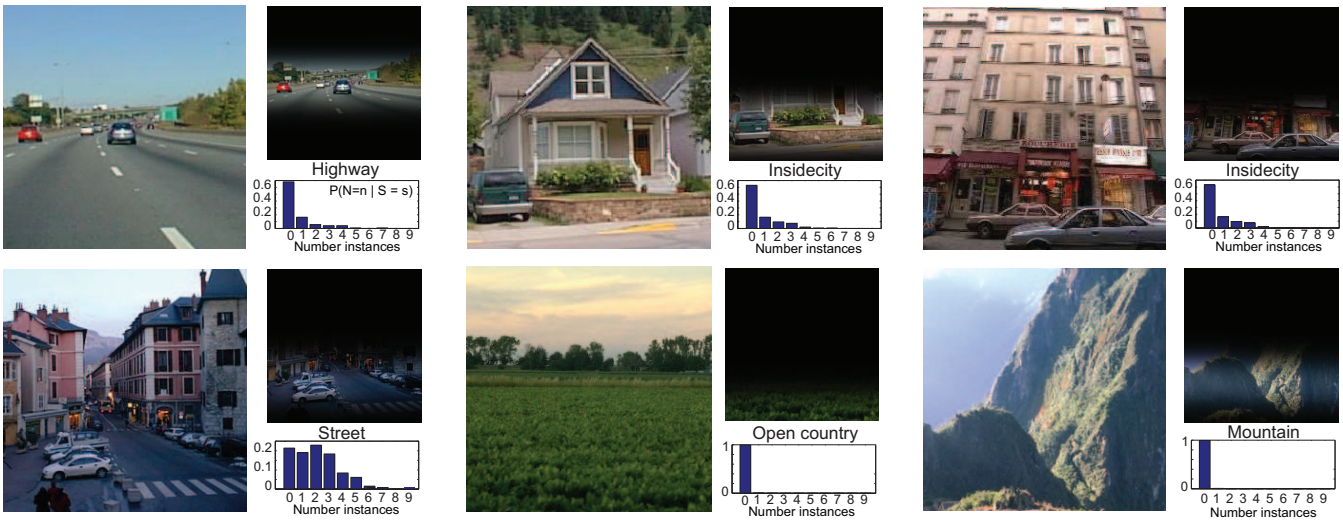

Figure 4: Predicting the presence/absence of cars in images and their locations using gist. The outputs shown here do not incorporate any information coming from a car detector and are only based on context. Note that in the dataset used to fit the distributions of object counts for each scene category, it is more common to find cars in street scenes (with many cars circulating and parked) than in highway scenes, where there are many shots of empty roads. Hence the histogram for highway shows $p(N^{car} = 0) = 0.6$.

This is fairly typical of this kind of approach. Below we will see how to eliminate many of these false positives by using global context.

3.2 Object presence detection using global image features

To determine if an object class is present in an image given the gist, we could directly learn a binary classifier of the form $p(P^t = 1|g)$. Similarly, to predict the number of objects, we could learn an ordinal regression function of the form $p(N^t|g)$. Instead, we choose a two-step approach in which we first estimate the category or type of scene, $p(S = s|g)$, and then use this to predict the number of objects present, $p(N^t|S = s)$. This approach has the benefit of having an explicit representation of the scene category (e.g., a street, a highway, a forest) which is also an important desired output of an integrated model.

We can classify the scene using a simple Parzen-window based density estimator

$$p(S = s|g) \propto p(g|S = s) = \frac{1}{J}\sum_{j=1}^{J} \mathcal{N}(g|\mu_j, \sigma_j^2 I),$$

where J is the number of mixture components for each class conditional density. Some examples of scene classification are shown in Figure 4. As shown in [12], this technique classifies 75% of the images correctly across 15 different scene categories. Other classifiers give similar performance.

Once we have estimated the scene category, we can predict the number of objects that are present using

$$p(N^t = n|g) = \sum_s p(N^t = n|S = s)p(S = s|g) \qquad (3)$$

where $p(N^t = n|S = s)$ is estimated by simple counting.

3.3 Object localization using global image features

The gist captures the overall spatial layout of the image, and hence can be used to predict the expected vertical location of each object class before running any detectors; we call this *location priming*. However, the gist is not useful for predicting the *horizontal* locations of objects, which are usually not very constrained by the overall structure of the scene (except possibly by the horizontal location of other objects, a possibility we ignore in this paper).

We can use any non-linear regression function to learn the mapping from gist to expected vertical location. We used a mixture of experts model [4], which is a simple weighted average of locally linear regression models. More precisely, we define

$$p(Y^t|g) = \sum_{k=1}^{K} w_k(g)\mathcal{N}(Y^t|\beta_k^T g, \sigma_k^2)$$

where Y^t is the vertical location of class t, K is the number of experts or mixture components, $\mathcal{N}$ represents a Gaussian or normal distribution, β_k are the regression weights for mixture component k, σ_k^2 is the residual variance, and $w_k(g)$ is the weight or "responsibility" of expert k, given by the softmax or multinomial logistic function:

$$w_k(g) = \frac{\exp(v_k^T g)}{\sum_{k'=1}^{K} \exp(v_{k'}^T g)}$$

We illustrate the predictions made by this model in Figure 6(b), where we scale the intensity of each image pixel by the probability density function $p(Y^t|g)$. We see that the effect is to "mask out" regions of the image which are unlikely to contain the object of interest. Some more examples can be seen in Figure 4.

3.4 Integrated model

We now discuss how to combine the various pieces described above. The basic idea is to use the global features

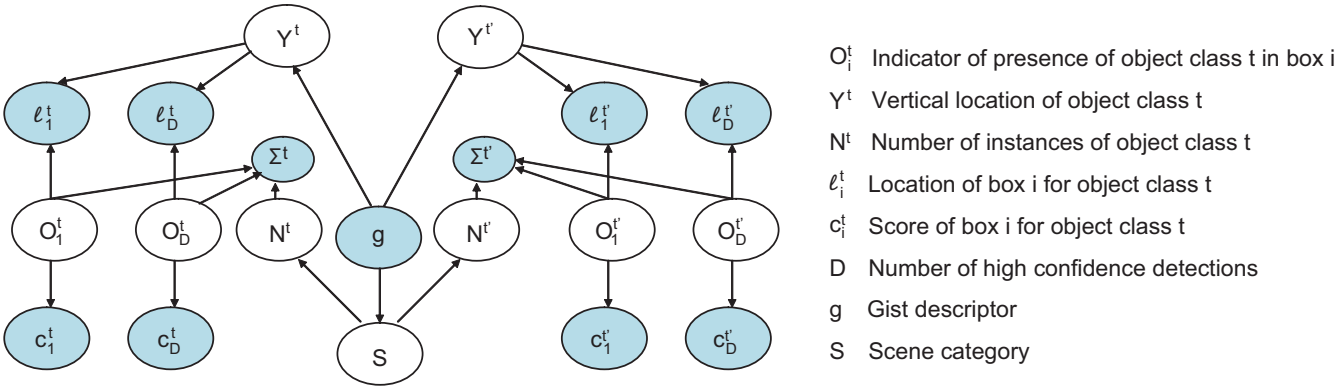

Figure 5: Integrated system represented as a directed graphical model. We show two object types, t and t', for simplicity. The observed variables are shaded circles, the unknown variables are clear circles. Variables are defined in the text. The Σ^t node is a dummy node used to enforce the constraint between the N^t nodes and the O_i^t nodes.

to make "top-down" predictions about how many object instances should be present, and where, and then to use the local patch classifiers to provide "bottom-up" signals.

The key issue is how to combine these two information sources. The approach we take is as follows (this differs slightly from the method originally described in [9]). Let us initially ignore location information. We treat the confidence score of the detector (c_i^t, defined in Equation 2) as a local likelihood term, and fit a model of the form $p(c_i^t|O_i^t = o) = \mathcal{N}(c_i^t|\mu_o^t, \sigma_o^t)$ for $o \in \{0, 1\}$. We can learn the parameters of this Gaussian by computing the empirical mean and variance of the scores when the detector is applied to a set of patches which do contain the object (so $o = 1$) and which do not contain the object (so $o = 0$). If we have a uniform prior over whether each detection is a true or false positive, $p(O_i^t = 1) = 0.5$, we can compute the posterior using Bayes rule as follows:

$$p(O_i^t = 1|c_i^t) = \frac{p(c_i^t|O_i^t = 1)}{p(c_i^t|O_i^t = 1) + p(c_i^t|O_i^t = 0)}$$

However, the detections are not all independent, since we have the constraint that $N^t = \sum_{i=1}^{D} I(O_i^t = 1)$, where N^t is the number of objects of type t. If we have top-down information about N^t from the gist, based on Equation 3, then we can compute the posterior distribution over detections in $O(2^D)$ time, given the gist, as follows:

$$p(O_{1:D}^t|g) \propto \sum_{n=0}^{D} p(O_{1:D}^t|n)p(N^t = n|g)$$

Here the term $p(O_{1:D}^t|n)$ is 1 only if the bit vector $O_{1:D}^t$ of length D has precisely n elements turned on. For compactness, we use the notation $1 : D$ to denote the indices $1, ..., D$. We can combine this with the local detectors as follows:

$$p(O_{1:D}^t|c_{1:D}^t, g) \propto p(O_{1:D}^t|g) \prod_{i=1}^{D} p(c_i^t|O_i^t)$$

If the gist strongly suggests that the object class is absent, then $p(N^t = 0|g) \approx 1$, so we turn all the object bits off in the posterior regardless of the detector scores, $p(O_{1:D}^t = \mathbf{0}|c_{1:D}^t, g) \approx 1$. If the gist strongly indicates that one object is present, then $p(N^t = 1|g) \approx 1$, and only one O_i^t bit will be turned on in the posterior; this will be the one with the highest detector score. And so on.

Now we discuss how to integrate location information. Let ℓ_i^t be the location of the i'th detection for class t. Since Y^t represents the expected location of an object of class t, we define another local likelihood term $p(\ell_i^t|O_i^t = 1, Y^t) = \mathcal{N}(\ell_i^t|Y^t, \tau^t)$, where τ^t is the variance around the predicted location. If the object is absent, we use a uniform distribution $p(\ell_i^t|O_i^t = 0, Y^t) \propto 1$. Of course, Y^t is not observed directly, but we can predict it based on the gist; this yields

$$p(\ell_i^t|O_i^t, g) = \int p(\ell_i^t|O_i^t, Y_t)p(Y_t|g)dY_t$$

which can be solved in closed form, since it is the convolution of two Gaussians. We can now combine expected location and detections as follows:

$$p(O_{1:D}^t|c_{1:D}^t, \ell_{1:D}^t, g) \propto p(O_{1:D}^t|g) \prod_{i=1}^{D} p(c_i^t|O_i^t)p(\ell_i^t|O_i^t, g)$$

To see the effect of this, suppose that the gist strongly suggests that only one object of type t is present, $p(N^t = 1|g) \approx 1$; in this case, the object bit which is turned on will be the one that has the highest score and which is in the most likely location. Thus confident detections in improbable locations are suppressed; similarly, unconfident detections in likely locations are boosted.

Finally, we discuss how to combine multiple types of objects. Intuitively, the presence of a car makes the presence of a pedestrian more likely, but the presence of a computer monitor less likely. However, it is impractical to encode a joint distribution of the form $p(P^1, \ldots, P^T)$ directly, since this would require $O(2^T)$ parameters. (Encoding $p(N^1, \ldots, N^T)$ directly would be even worse.) Instead, we introduce the scene category latent variable S, and assume that the presence (and number) of object types is conditionally independent given the scene category:

$$p(N^1, \ldots, N^T) = \sum_s p(S = s) \prod_{t=1}^{T} p(N^t|S = s)$$

Given this assumption, we can perform inference for multiple object types in parallel as follows: for each possible scene

category, compute the posterior $p(O^t_{1:D}|c^t_{1:D}, \ell^t_{1:D}, g, S = s)$ as described above, and then combine them using a weighted average with $p(S = s|g)$ as the weights.

In summary, our whole model is the following joint probability distribution:

$$p(O^{1:T}_{1:D}, N^{1:T}, Y^{1:T}, S|c^{1:T}_{1:D}, \ell^{1:T}_{1:D}, g) \propto p(S|g) \times \prod_{t=1}^{T} p(Y^t|g)p(N^t|S)p(O^t_{1:D}|N_t) \prod_{i=1}^{D} p(\ell^t_i|O_i, Y^t)p(c^t_i|O_i)$$

This is illustrated as a probabilistic graphical model (see e.g., [5]) in Figure 5. There is one node for each random variable: the shaded nodes are observed (these are deterministic functions of the image), and the unshaded nodes are hidden or unknown, and need to be inferred. There is a directed edge into each node from all the variables it directly depends on. For example, the $g \rightarrow S$ arc reflects the scene classifier; the $g \rightarrow Y^t$ arc reflects the location priming based on the gist; the $S \rightarrow N^t$ arc reflects the object counts given the scene category; the $O^t_i \rightarrow c^t_i$ arc reflects the fact that the presence or absence of an object of type t in patch i affects the the detector score or confidence c^t_i; the $O^t_i \rightarrow \ell^t_i$ arc is a deterministic link encoding of the location of patch i; the $Y^t \rightarrow \ell^t_i$ arc reflects the $p(\ell^t_i|Y^t, O^t_i)$ term; finally, there are the $O^t_i \rightarrow \Sigma^t$ and $N^t \rightarrow \Sigma^t$ arcs, which is simply a trick for enforcing the $N^t = \sum_{i=1}^{D} I(O^t_i = 1)$ constraint. The Σ^t node is a dummy node used to enforce the constraint between the N^t nodes and the O^t_i nodes. Specifically, it is "clamped" to a fixed state, and we then define $p(\Sigma^t|O^t_{1:D}, N^t = n) = \mathcal{I}(\sum_i O^t_i = n)$ (conditional on the observed child Σ^t, all the parent nodes, N^t and O^t_i, become correlated due to the "explaining away" phenomenon [5]).

From Figure 5, it is clear that by conditioning on S, we can perform inference on each type of object independently in parallel. The time complexity for exact inference in this model is $O(ST2^D)$, ignoring the cost of running the detectors. (Techniques for quickly evaluating detectors on large images, using cascades of features, are discussed in [20].) We can speed up inference in several ways. For example, we can prune out improbable object categories (and not run their detectors) if $p(N^t > 0|g)$ is too low, which is very effective since g is fast to compute. Of the categories that survive, we can just run their detectors in the primed region, near $E(Y^t|g)$. This will reduce the number of detections D per category. Finally, if necessary, we can use Monte Carlo inference (such as Gibbs sampling) in the resulting pruned graphical model to reduce time complexity.

4. RESULTS

Example of the integrated system in action are shown in Figure 6(c): We see that location priming, based on the gist, has down-weighted the scores of the detections in improbable locations, thus eliminating false positives. In the second row, the local detector is able to produce a confident detection, but the second car produces a low confidence detection. As the low confident detection falls inside the predicted region, the confidence of the detection increases. Note that in this example there are two false alarms that happens to also fall within the prediction region. In this case, the overall system will increase the magnitude of the error. If the detector produces errors that are contextually correct, the integrated model will not be able to discard those. The third row shows a different example of failure of the integrated model. In this case, the structure of the scene makes the system think that this is a street scene, and then mixes the boats with cars. Despite these sources of errors, the performances of the integrated system are substantially better than the performances of the car detectors in isolation.

For a more quantative study of the performance of our method, we used the scenes dataset from [11] consisting of 2688 images covering 8 scene categories (streets, building facades, skyscrapers, highways, mountainous landscapes, coast, beach and fields). We use half of the dataset to train the models and the other half for testing.

Figure 7 shows performances at two tasks: object localization and object presence detection. The plots correspond to precision-recall plots: the horizontal axis denotes the percentage of cars in the database that have been detected for a particular detection threshold and the vertical axis is the percentage of correct detections for the same threshold. Different points in the graph are achieved by varying the decision threshold. For both tasks, the plot shows the performances using an object detector alone, the performances of the integrated model, and the performance of an integrated model with an oracle that tells for each image the true context. The performance of the integrated model has to be within the performance of the detector alone and the context oracle.

Figure 7(right) shows a precision-recall curve which quantifies the performance of 3 different systems for detecting object presence. The worst one is based on an object detector using local features alone, the middle one is our integrated system which uses local and global features, and the best one is an oracle system based on using the true scene category label. We see that our integrated model does much better than just using a detector, but it is clear that better scene classification would improve the results further. It is important to note that detecting if an object is present in an image can be done with good accuracy even without object localization. The knowledge of the scene depicted by the image can be enough. For instance, in a picture of a street it is quite certain that a car will appear in the picture, while it is unlikely that a car will appear on a beach scene. Therefore, the relation between the scene category and the object can provide a lot of information even when the detector fails to locate the object in the image.

Figure 7(left) shows a precision-recall curve which quantifies the performance of 3 different systems for localizing objects. Again the worst one is based on an object detector using local features alone, the middle one is our integrated system which uses local and global features, and the best one is a oracle system based on using the true scene category label. In this case, knowing the true scene category does not help as much: it can eliminate false positives such as cars in indoor scenes, but it cannot eliminate false positives such as cars detected in a street scene but up in the sky. (Of course, the gist-based location priming system tries to eliminate such spatial outliers, but knowing the scene category label does not help with localization.)

Object localization is a much harder task than merely detecting the presence of an object. This is evident from the horizontal scale in Figure 7(left): the recall never goes beyond about 30%, meaning that about 70% of cars are missed by the detector, mostly due to occlusion. Even if context can be used to narrow down the search space and to remove false alarms that occur outside the relevant image

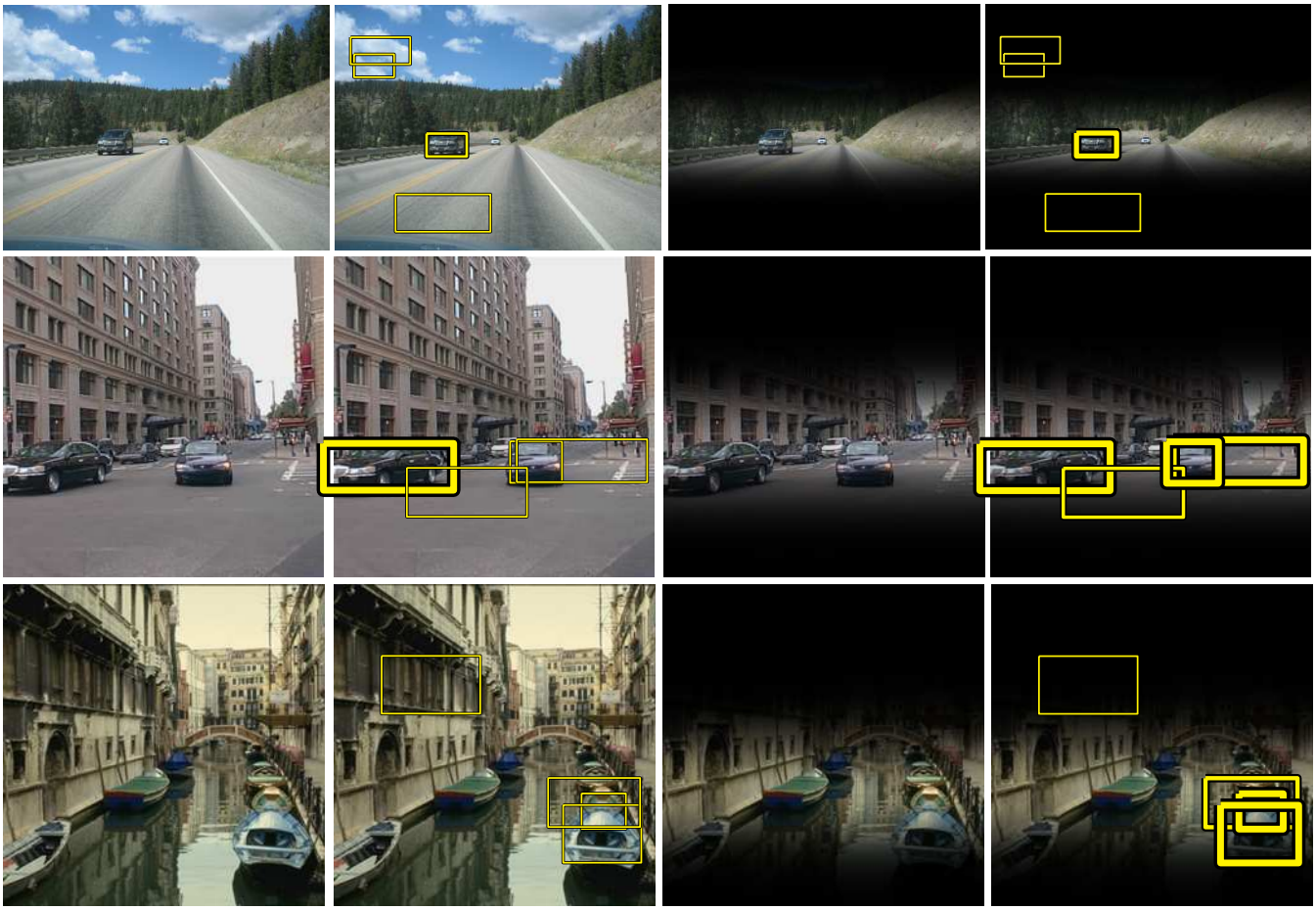

Figure 6: a) Three input images b) Top 4 detections from an object detector based on local features. The thickness of the boxes is related to the confidence of the detection. c) Predicted location of the car based on global features. d) Combining local and global features.

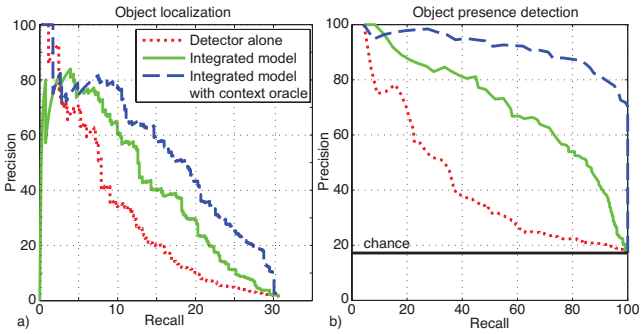

Figure 7: Performance on car localization (left) and car presence detection (right).

region, still, if the detector is not able to localize the object, context information will not be able to precisely localize the object. The use of global context (even with the oracle) does not increase the recall (as this requires the detector to work), however context is able to increase the precision as it is able to remove false alarms in scenes in which cars are not expected. It is possible that a finer grained notion of context, perhaps based on other objects, could help in such cases. Note, however, that for image retrieval applications (e.g., on the web), object presence detection is sufficient. For speed reasons, we could adopt the following two stage approach: first select images that are predicted to contain the object based on the gist alone, since this is much faster than applying a sliding window classifier; then apply the integrated model to further reduce false positives.

5. CONCLUSIONS

We have discussed one approach for combining local and global features in visual object detection and localization. Of course, the system is not perfect. For example, sometimes objects appear out of context and may be accidently eliminated if the local evidence is ambiguous (see Figure 8). The only way to prevent this is if the local detector gives a sufficiently strong bottom-up signal. Conversely, if the detector makes a false positive error in a contextually plausible location, it will not be ruled out by our system. But even people can also suffer from such "hallucinations".

In more general terms, we see our system as a good example of probabilistic information fusion, an approach which is widely used in other areas such as speech recognition, which combines local acoustic models which longer-range language models. Since computer vision is inherently a difficult inverse problem, we believe it will be necessary to combine as many sources of evidence as possible when trying to infer the true underlying scene structure.

6. ACKNOWLEDGMENTS

Funding for this work was provided by NGA NEGI-1582-04-0004, MURI Grant N00014-06-1-0734, NSF Career award IIS 0747120, NSF contract IIS-0413232, a National Defense Science and Engineering Graduate Fellowship, and gifts from Microsoft and Google. KPM would like to thank NSERC and CIFAR for support.

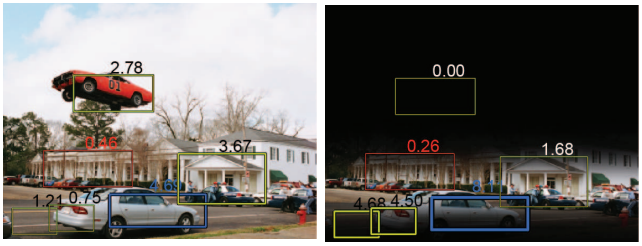

Figure 8: An object which is out of context may be falsely eliminated by our system.

7. REFERENCES

[1] L. Fei-Fei and P. Perona. A Bayesian hierarchical model for learning natural scene categories. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 524–531, 2005.

[2] D. Heeger and J. R. Bergen. Pyramid-based texture analysis/synthesis. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 229–238, New York, NY, USA, 1995. ACM.

[3] D. Hoiem, A. Efros, and M. Hebert. Geometric context from a single image. In *IEEE Intl. Conf. on Computer Vision*, 2005.

[4] M. I. Jordan and R. A. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6:181–214, 1994.

[5] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[6] S. Kumar and M. Hebert. Discriminative random fields: A discriminative framework for contextual interaction in classification. In *IEEE Intl. Conf. on Computer Vision*, 2003.

[7] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 2169–2178, 2006.

[8] K. Murphy, A. Torralba, D. Eaton, and W. T. Freeman. Object detection and localization using local and global features. In J. Ponce, M. Hebert, C. Schmidt, and A. Zisserman, editors, *Toward Category-Level Object Recognition*. 2006.

[9] K. Murphy, A. Torralba, and W. Freeman. Using the forest to see the trees: a graphical model relating features, objects and scenes. In *Advances in Neural Info. Proc. Systems*, 2003.

[10] A. Oliva and P. G. Schyns. Diagnostic color blobs mediate scene recognition. *Cognitive Psychology*, 41:176–210, 2000.

[11] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *International Journal in Computer Vision*, 42:145–175, 2001.

[12] A. Quattoni and A. Torralba. Recognizing indoor scenes. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 413–420, 2009.

[13] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie. Objects in context. In

IEEE Intl. Conf. on Computer Vision, Rio de Janeiro, 2007.

[14] X. H. Richard, R. S. Zemel, and M. A. Carreira-perpinan. Multiscale conditional random fields for image labeling. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 695–702, 2004.

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

[16] T. M. Strat and M. A. Fischler. Context-based vision: recognizing objects using information from both 2-D and 3-D imagery. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(10):1050–1065, 1991.

[17] A. Torralba. Contextual priming for object detection. *Intl. J. Computer Vision*, 53(2):153–167, 2003.

[18] A. Torralba, K. Murphy, and W. Freeman. Contextual models for object detection using boosted random fields. In *Advances in Neural Info. Proc. Systems*, 2004.

[19] A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing visual features for multiclass and multiview object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(5):854–869, 2007.

[20] P. Viola and M. Jones. Robust real-time object detection. *Intl. J. Computer Vision*, 57(2):137–154, 2004.