Microsoft Research Technical Report
MSR-TR-2011-99

A Framework for Encoding Object-level Image Priors

Jenny Yuen[1] C. Lawrence Zitnick [2] Ce Liu[2] Antonio Torralba[1]
{jenny, torralba}@csail.mit.edu {larryz, celiu}@microsoft.com

[1]MIT [2] Microsoft Research

Abstract. Although context is a key component to the success of building an object recognition system, it is difficult to scale and integrate existing formulations of contextual rules to take into account multiple-sources of information. In this paper, we propose a generic, object-level image prior to represent rich, complicated contextual relationships. A maximum entropy distribution is learned to model the possible layouts of objects and scenes by placing constraints on the prior distribution. We demonstrate that this new object-level image prior not only scales well to include arbitrary high-order object relationships, but also seamlessly integrates multiple-sources of image information such as scene categorization, scene parsing and object detection. The result is a more comprehensive understanding of the image.

1 Introduction

The occurrence of objects in images is far from random. The likelihood of observing an object is highly dependent on contextual information such as the scene depicted, and the presence and location of other objects in the image. For instance, a street scene is more likely to contain a car than an indoor scene. Similarly, a window is likely to occur in a image with a building. However, a window that is not spatially supported by a building is unlikely.

Recently, many works have attempted to take advantage of contextual information using a variety of representations. Contextual information may be encoded at the scene level by describing if and where certain objects are likely to occur in a scene [1, 2]. Several works represent pair-wise object relationships, such as co-occurrence [3, 4], and spatial relationships [5–7], using conditional random fields (CRFs). 3D geometry may be used to constrain the relative position of objects [8]. The direct use of large labeled image datasets can also inherently represent contextual information by providing a dense sampling of exemplars [9]. Each of these representations essentially encode a prior on the set of possible object occurrences, object locations and scenes.

One challenge of using contextual information is the integration and scale of various amounts and types of information. The above representations are designed to model certain subsets of information, but it is difficult to combine these formulations. We want to utilize a variety of information sources for scene parsing, *e.g.* scene categorization, segmentation and object detection such that reliable sources can help disambiguate other, less reliable sources. Furthermore, some representations may be intractable when considering relationships beyond pair-wise, such as ternary and quaternary relationships.

In this paper, we propose a generic object-level image prior for modeling contextual relationships between both objects and scenes. The prior is a function of binary variables encoding the presence of objects, specific spatial relationships between objects, and scene types. The relationships between variables is encoded using a maximum entropy framework that places constraints on the joint distribution. A maximum entropy distribution is learned to agree with everything that is known and encoded by constraints, and to avoid assuming anything that is unknown. This approach provides two main advantages. First, high order relationships such as the presence of multiple objects may be encoded without requiring exponential growth in problem complexity. Second, the prior may be used to combine object and scene likelihood information from multiple sources.

We demonstrate the generality of our approach using both varying types of contextual information and different sources of object and scene information. Experiments on contextual information include the use of pair-wise and higher order object occurrence information, spatial relationships between objects and scene labels. In addition, we combine various sources of object and scene information from object detectors [10], SIFT flow [9], scene recognition [2] and segment-based detectors [3]. The use of multiple information sources provides better results than any one source alone. Selecting a useful and compact set of contextual information can be difficult given the large number of possible rules. A feature pursuit algorithm [11] is adopted to automatically select contextual information in a greedy manner. Our system is evaluated on a subset of LabelMe database [12], demonstrating the performance gain due to the new formulation of object-level image prior.

This paper is organized as follows: Prior work is described in the next section. Section 3 describes our generic object-level image prior. A set of validation experiments are provided in Section 4, followed by several real applications in Section 5. Finally, sections7 contain conclusion of our work.

2 Prior Work

Many works have attempted to take advantage of contextual information for a variety of tasks. Scene level context has proven useful for narrowing the collection and location of objects that may be present in an image [1, 2]. Relative 3D location information has been used to reduce false positives in object detections such as cars and people [8]. Pair-wise relationships between objects have been explored in [3, 4] for co-occurrence and in [5–7] for spatial relationships such as relative location, support and surround. Contextual information may also be represented using a large dataset of labeled images [12]. In [9] optical flow methods are used to transfer object label information from large image collections that inherently encode contextual information. Finally, an study of various contextual information sources is studied in [13].

Recently, many works have demonstrated advances integrating contextual information inherently in images. One family uses a top-down hierarchical and generative approach and models scenes globally as containing objects, which at the same time can be decomposed into parts; finally, they attempt to learn the parameters of these hidden variables jointly [14–17]. These methods are useful as they discover and enforce implicit semantic relationships between different elements in a scene.

Maximum entropy models are widely used in areas like natural language processing [18], object recognition [19], and image annotation [20], amongst many others.

3 Object-level Image Prior

Our goal is to determine the most likely set of object and scene labelings given an image. This is commonly formulated in two parts, a likelihood term and a prior term. The likelihood term, also called the data term, computes the likelihood of an object or scene directly from pixel information. The prior term represents the contextual information by computing the probability of a configuration of objects and scenes. We refer discussion of the likelihood term to section 4. We now discuss the prior term.

Our prior is defined on a set of binary variables $\mathbf{x}$ that encode the set of possible configurations of objects and scenes in the image. In its simplest form $\mathbf{x}$ encodes the presence of objects. For instance, if a variable corresponds to the presence of a car it has a value of 1 if at least one car is present and a value of 0 otherwise. Additional variables can be added to encode scene types, such as indoor, street and forest scenes. Finally, $\mathbf{x}$ may include variables corresponding to spatial relationships between objects, such as supporting, surrounding and neighboring. That is, a single variable could represent whether at least one window is supported by a building. If information about whether a door is supported by a building was also desired another variable would be added. Variables may also be functions of the number of unique objects in an image, the number instances for a specific object, etc.

Given a specific instance of variables $\mathbf{x}$, we want to compute the prior probability $p(\mathbf{x})$. We compute the prior using a maximum entropy framework. The framework places a set of constraints on the joint distribution $p(\mathbf{x})$. The distribution with maximum entropy that satisfies the constraints is then found. A constraint is encoded using a binary function $f_i(\mathbf{x}) \in \{0, 1\}$ and enforces the following:

$$\sum_{\mathbf{x}} p(\mathbf{x}) f_i(\mathbf{x}) = \tilde{p}(f_i) \tag{1}$$

where $\tilde{p}(f_i)$ is the empirical probability of f_i:

$$\tilde{p}(f_i) = \frac{1}{m} \sum_{j=1}^{m} f_i(\mathbf{x}_j) \tag{2}$$

m is the number of training examples. We may interpret equation (2) as constraining certain marginal probabilities specified by f_i to be equal to their corresponding empirical probabilities. Constraints are typically used to constrain the co-occurrence probabilities of small sets of variables. For instance, a constraint function might have a value of 1 if a car and street street scene variable are both 1. As a result, we enforce the prior probability of observing a car in a street scene will be equal to the empirical probability.

The maximum entropy distribution $p_\Lambda(\mathbf{x})$ that satisfies the constraints $f_i \in \{f_1, \ldots, f_n\}$ has an exponential form:

$$p_\Lambda(\mathbf{x}) = \frac{1}{Z} \exp\left(\sum_{i=1}^{n} \lambda_i f_i(\mathbf{x})\right) \tag{3}$$

$$Z = \sum_{\mathbf{x} \in \mathbf{X}} \exp\left(\sum_{i=1}^{n} \lambda_i f_i(\mathbf{x})\right) \tag{4}$$

where λ_i is the scalar weight corresponding to constraint function f_i, Z is the partition function, and $\mathbf{X}$ is the set of all valid variable assignments. Some variable assignments may not be valid, such a assigning a value of 0 to the presence of a building, but a value of 1 to the value of window supported by a building. The weight vector $\Lambda = \{\lambda_1, ..., \lambda_n\}$ is learned using the Improved Iterative Scaling algorithm [21]. If efficiency is desired improved algorithms may also be used [22]. When updating the weights Λ, the partition function and marginals $p_\Lambda(f_i)$ must be computed. If the state space of $\mathbf{X}$ is too large, sampling techniques must be used. In our experiments we used 40,000 samples per iteration.

4 Model Validation

We can use our prior model in conjunction to a likelihood term to regularize noisy observations. In this section we will demonstrate the flexibility of our prior term for integrating multiple contextual rules in the area of multi-class object recognition. Specifically, the task will be, given an image, to associate class labels given object-level segments in an image. To simplify the framework, and fairly evaluate the role of the prior, we assume the case of ideal segmentations (*i.e.* use ground truth segmentations).

4.1 Implementation details

We model an image as an ensemble of objects interacting with each other according to a contextual dictionary. Let $\mathbf{c} = \{c_1.c_2, ...c_k\}$ be the vector of object class labels given to each of the segments $\mathbf{S} = \{S_1, .S_2, ..., S_k\}$ in the image $\mathbf{I}$ and the scene class label $d \in \mathbf{D}$. In this formulation, our likelihood term will model the generation of each independent element by a hidden variable while our prior term will model the contextual relationships in the scene. We decompose our model into a likelihood and a prior term following Bayes' rule.

$$\begin{aligned} p(\mathbf{c}, d|\mathbf{S}, \mathbf{I}) &= \frac{p(\mathbf{S}|\mathbf{c})p(\mathbf{I}|d)p(\mathbf{c}, d)}{p(\mathbf{S}, \mathbf{I})} \\ &\propto p(\mathbf{S}|\mathbf{c})p(\mathbf{I}|d)p(\mathbf{c}, d) \end{aligned}$$

We assume the likelihood of the segments and image are independent given the object class and scene information. In the above formulation our likelihood term is $p(\mathbf{S}|\mathbf{c})p(\mathbf{I}|d)$ and our prior is $p(\mathbf{c}, d)$.

We compute the likelihood of the segments given the class labels using a feature vector composed of a normalized histogram of visual words, color centers, and size. For simplicity, each class distribution is modeled as a mixture of Gaussians.

$$p(S_i|c_i) = \kappa(S_i; c_i, \theta_{color})\tau(S_i; c_i, \theta_{sift})\psi(S_i; c_i, \theta_{size})$$

where for each object class c_i, $\kappa(S_i; c_i, \theta_{color})$ denotes the mixture of Gaussians modeling the color component, $\tau(S_i; c_i, \theta_{sift})$ denotes the mixture of Gaussians modeling the texture component (dense SIFT), and $\psi(S_i; c_i, \theta_{size})$ denotes the mixture of Gaussians modeling the size feature component.

The data term assumes independence between each segment and each feature type (color, texture, and shape) and is computed as the product over all segments of the probability of each segment S_i being generated by its associated hidden variable c_i.

$$p(\mathbf{S}|\mathbf{c}) = \prod_{i=1}^{k} p(S_i|c_i)$$

The scene likelihood term $p(\mathbf{I}|d)$ is a binary vector corresponding to whether the gist descriptor [23] indicates the specified scene type.

The prior term is computed from equation (3), $p(\mathbf{c}, d) = p_\Lambda(\mathbf{x})$, by creating a set of variables $\mathbf{x}$ that indicate the occurrence of each object and scene type. That is, the number of variables in $\mathbf{x}$ is equal to the number of object types plus the number of scene types. For experiments using spatial relationships, additional corresponding variables are added. The values of $\mathbf{x}$ are computed directly from the label assignments $\mathbf{c}$ and d. The constraint values are found using the statistics of the training set.

We explore a variety of contextual constraints to form a dictionary of constraints on the prior distribution. In addition to the following high level constraints, we also use a set of baseline constraints on the occurrence of each individual variable. Many contextual rules can be integrated in this framework, here we describe some examples:

- **Co-occurrence** We consider all pairwise co-occurrences,as well as the triplets, and quadruplets that appear often (more than 5% of the time) in the training set.
- **Scene-to-object relationships** Images can also be classified into different scenes as a function of the objects present in the image. For each image, we construct a binary feature vector indicating the presence or absence of each object class. We further train SVM classifiers for each scene category using these image vectors to determine, given a vector of class occurrences, to which scene category the class might belong to, if any. Each SVM will represent a contextual rule for its associated scene class and its binary output will be the output its constraint function.
- **Gist-based scene co-occurrence with object classes** Another way to integrate scene information is using the gist descriptor [23], shown successful describing scenes in absence of the identity of the objects present. We build an SVM classifier to associate an image to a scene class. These gist-based constraint functions take the AND between a co-occurrence constraint and a gist-based scene classifier. Note that this evaluation function differs on the scene-to-objects relationship in that it uses the gist data as opposed to the object labels.
- **Attachment relationships** Aimed at scenes with objects that are part of a larger object or are composed by other objects (*e.g.* windows, doors, and awnings are attached to buildings, cars are supported by ground planes like roads, etc.). Given two segmentation masks, we can compute their overlapping region to check if one of the segments is inside the other one.

Finally, we perform inference using Gibbs sampling. As described in section 4, the posterior probability for a label assignment can be easily evaluated. However, it is intractable to evaluate the posterior probability for all possible label assignments when the number of segments becomes large. We perform 10,000 iterations of Gibbs sampling with 10 random restarts using the likelihood term labels as an initial guess.

4.2 Evaluation

We evaluate the contribution of the prior term on the MSRC data set containing 21 classes, and the *spatial envelope* subset of the LabelMe database using the 17 most frequently occurring categories. Each data set was split into training and testing portions following the splits published by [24] and [25].

We start by exploring class co-occurrence relations of different orders in the MSRC data set as well as scene-to-object relationships (since images in this set center mostly on the main object and gist-based scene constraints would not appear reliable). Figure 4.2 a shows the total pixel precision for objects in the database comparing the likelihood term alone and the result after context integration; notice the consistent improvement when considering context. Figure 4.2

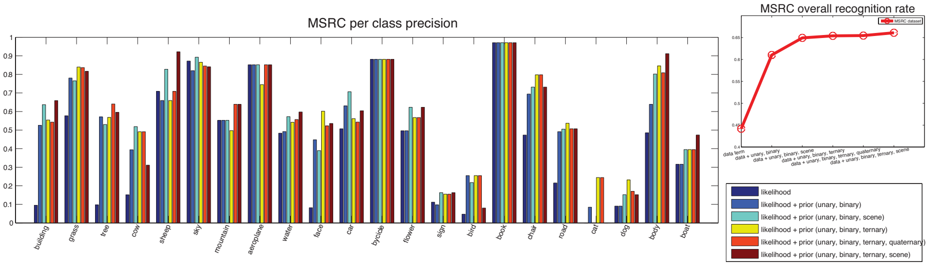

Fig. 1. Validation on MSRC toy set. We explore how much context can improve recognition given a simple likelihood term in the scenario of ideal segmentations in the MSRC and a subset of the LabelMe database. We observe a quick improvement in the per-pixel precision for several classes by only including pair-wise object co-occurrences. The addition of higher order constraints (ternary, quaternary, and scene-based) shows an asymptotic improvement (right) in the overall precision. This is because the dataset has few objects per image (between 1 and 5).

shows the overall pixel precision as a function of the different context terms used. Notice the great improvement form 45% recognition rate to 63% just by adding binary co-occurrences, the boost to 65% when considering ternary relationships, and the marginal improvement by adding higher order relationships. Higher order relationships are not necessary for the MSRC dataset since the number of objects in an image is typically below four.

Using the LabelMe dataset, we explore gist-based scene constraints together with basic co-occurrences and attachment relationships. Figure 4.2b shows the per-class recognition results. Most of the classes show considerable improvement compared to the likelihood term in isolation.

5 Application: enhancing scene parsings with object detectors

Sliding window object detectors are known to work well for objects with well defined parts such as people and cars, but have difficulty learning templates of polymophous regions such as roads and skies. Conversely, segmentation-based approaches perform well separating large and amorphous regions such as the sky and grass but have limitations segmenting small objects such as windows and cars. Therefore, we separate our object classes into *stuff*and *things*. We start by using a slight modification of SIFT-flow-based parsing system [25] by only transferring labels of *stuff*-type categories and generating N contextually sound candidate scene parsings. Furthermore, we train object detectors for *things*using a state-of-the-art sliding window detector [10]. SIFT-flow parsings consist of a pixel-level segmentation and labels for each pixel. A parsing for a query image is generated by retrieving the top nearest neighbors from the labeled database and warping the annotations of the neighbors to adapt to the query image. And

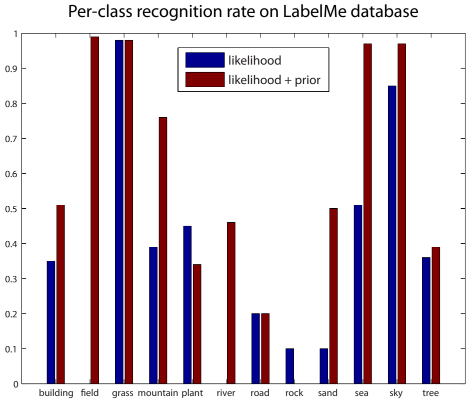

Fig. 2. Validation on LabelMe toy set. The number of images in this set lie between 2 and 30. Therefore, the prior we use for this set includes co-occurrences, gist-based scene relations to object classes, and attachment relationships.

advantage of SIFT-flow parsings is the inherent contextual coherence at a high level present in the transferred labels as they originate from the same image. In this setup, we will redefine our vector of label assignments **c** now as built by two label types for each kind of input (a scene parsing or an object detection of a particular class) and the vector **S** now containing the observed image information required for each kind of detection:

$$\begin{aligned}\mathbf{c} &= \left[m_{\mathit{stuff}}, \mathbf{c}_{\mathit{things}}\right]\\ \mathbf{S} &= \left[S_{\mathit{stuff}}, \mathbf{S}_{\mathit{things}}\right]\end{aligned}$$

where $m_{\mathit{stuff}} \in \{1, ..., N\}$ denotes the index of one of the top N scene parsings from the nearest neighbors of the sift flow match and $\mathbf{c}_{\mathit{things}} = \{c_{jk} | c_{jk} \in \{0, 1\}\}$ indicates the presence/absence of the kth detection for the jth object class of the *things*; S_{stuff} in this case is the SIFT flow matching score of the m_{stuff}th neighbor and $\mathbf{S}_{\mathit{things}}$ is a vector of containing the object detection scores for each of the detected windows.

Finally, we assume independence between each element in the image and generate the likelihood for the image as a product of the likelihoods of each image element (object detections and scene parsing).

$$\begin{aligned}p(\mathbf{S}|\mathbf{c}) &= p(S_{\mathit{stuff}}|m_{\mathit{stuff}}{}_{\text{is a good match}})p(\mathbf{S}_{\mathit{things}}|\mathbf{c}_{\mathit{things}})\\ &= p(S_{\mathit{stuff}}|m_{\mathit{stuff}}{}_{\text{is a good match}}) \prod_{j=1}^{n_{\mathit{things}}} \prod_{k=1}^{m_j} p(S_{jk}|c_{jk})\end{aligned}$$

where $p(S_{stuff}|m_{stuff}\text{is a good match})$ is the probability of obtaining a matching score of S_{stuff} for the selected m_{stuff}th nearest SIFT-based neighbor match given that this match is a good one; for simplicity, we'll abbreviate abbreviate this term as $p(S_{stuff}|m_{stuff})$. Finally, $p(\mathbf{S}_{things}|\mathbf{c}_{things})$ is the probability of a current set of detections given their respective bounding boxes. This last term can be broken into multiple independent factors $p(S_{jk}|c_{jk})$ encoding the probability of observing the kth bounding box from the detection given the jth object class.

The prior term is computed with a set of variables $\mathbf{x}$ that now indicate the occurrence of each object, the detected scene type, and the attachment satisfaction information. Given some label configuration, the occurrence of each object can be easily extracted from the labels of the selected scene parsing and the categories of the detections that have been selected as active. The detected scene type is obtained from the scene category classifier that with the highest confidence given the image data. Finally, the attachment information is computed by checking for overlaps between pairs of segments given the current label configuration.

Notice that, due to the flexibility of our object prior and its decoupling from the likelihood term, we can apply the same prior learned in 4 directly to our new input data regardless of the change in the form of the input. The only variant will lie in the likelihood term and in reshaping the contextual rule evaluation functions to fit to the new data.

Generating candidate scene parsings The original SIFT-flow parsing method generates one parsing per query image by consolidating votes from the top nearest neighbor labelings. Our framework is designed to consider multiple candidates and choose the most contextually coherent one. We slightly modify the original SIFT-flow parsing framework to generate multiple parsings. One way is by directly taking the warped parsing of each N nearest neighbor. Another alternative consists of randomly sampling a subgroup of top nearest neighbors and merging their votes to generate multiple consolidated parsings. We explore both methods and show their respective results in figure 3

Inference We add a slight modification our original inference method to minimize the number of iterations required to find the correct configuration. Because of the potentially high number of detections per image, the vector $\mathbf{c}$ can have up to 100 dimensions. However, as many of the detections satisfy the same group of contextual rules in isolation, we can group the detections depending on the rules they would satisfy if they were selected. For example, we can group the car detections on top of the road in one cluster and the ones on top of the sky in another. This block sampling mode reduces the dimensionality of our input vector and requires much less time to find the correct configuration.

6 Experiments and results

Figure 3 shows the per-pixel precision for each *stuff* class. We consider two scenarios for generating candidate parsings: (1) the independent parsings from the top 10 nearest neihgbors and (2) different candidate parsings generated by randomly selecting 5 of the nearest neighbors and merging them to generate a consolidated parsing like in [25]. For the first case using independent parsings, our baseline is the parsing from the first nearest neighbor; for the second case, the result merging the top 5 nearest neighbors is our baseline. We observe a higher increase in overall performance using the prior in the first scenario (from 46.45% to 52.98%) in comparison with the second case (from 61.3% to 66.75%). This might be because the second type of parsing already integrates some low level context by considering the votes of more than a nearest neighbor.

Figure 6 also shows the precision-recall curves for the car and window detections. We observe a greater increase in performance in the car class because of the high false positive rate. Given this formulation, we will only be able to remove contextually incoherent results for detections, removing false positives, but will not find new detections. Notice how the remaining false positives **??**are still contextually coherent.

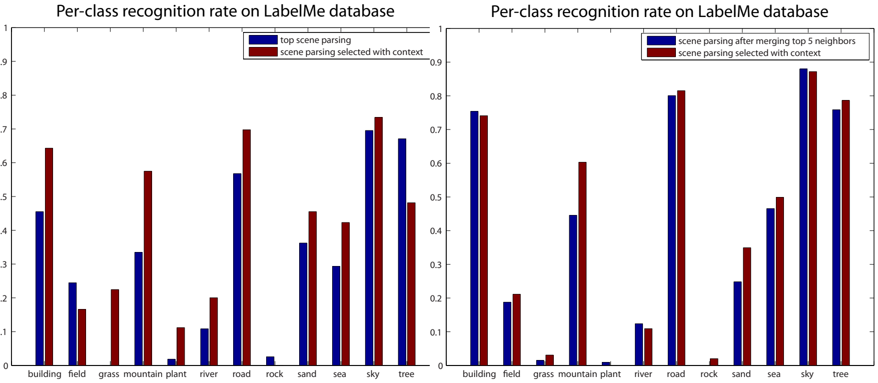

Fig. 3. Results for *stuff* on LabelMe data. Per-pixel precision rates for *stuff* categories when (1) directly selecting the scene parsing of the top nearest neighbor compared to using our prior to select from the top 10 nearest neighbor SIFT parsings (top), and (2) directly selecting the parsing after merging the labels from the top 5 results compared to using our prior to select from 10 candidates generated from merging groups of 5 results from the nearest neighbor pool (bottom).

Finally, figure 5 shows examples with the query image, the generated parsings and detections, and the selected results after integrating our prior. Notice how in many cases, the first scene parsing belongs to a scene category different from the query image. The gist-based scene detector is very helpful in this scenario to constraint the object classes present and pick more contextually coherent scene parsings. The gray segments indicate unlabeled regions (our database is no longer fully labeled since we restrict the objects to the most commonly occurring *stuff* classes.

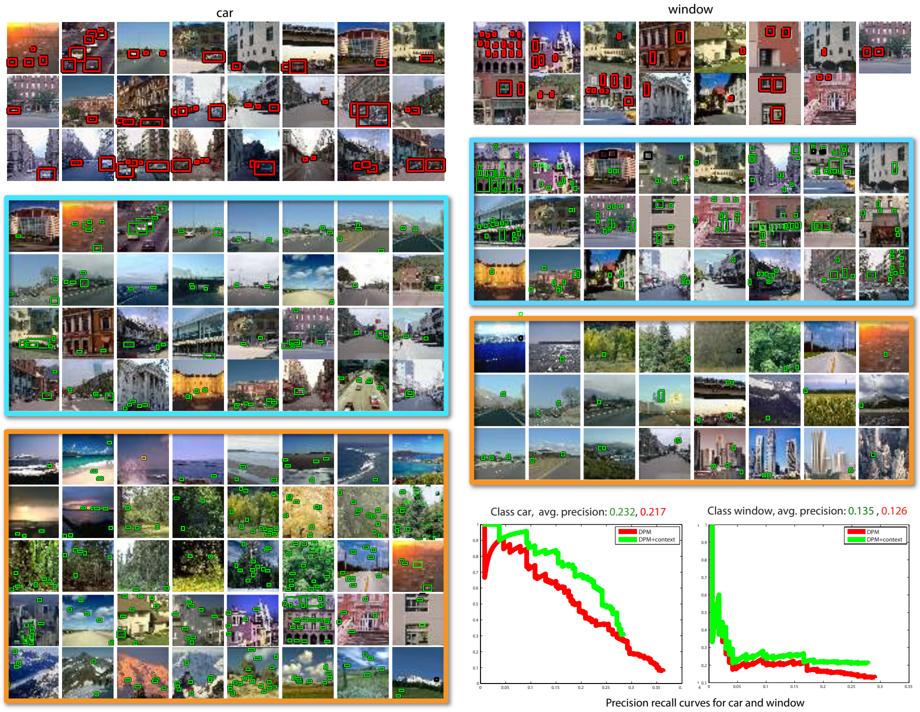

Fig. 4. Object detections returned by the DPM detector [10]. The correct detections for cars and windows (red) and the false alarms (enclosed in cyan) after integrating our prior. The orange box contains the false alarms that were ruled out using the prior. The detections are considered correct if more than 50% of the object region overlaps with the detected bounding box. The scene category and present object classes help rule out samples like cars on the water or in forest scenes, however, it is not powerful enough to rule out some false alarms on top of roads. Window detections also present an interesting phenomenon; they are not labeled in scenes of the class *tallbuildings* so many of the detected windows in this scene category are cleaned up. Context also has the drawback that it can only rule out false positives in this case, but not find missed object instances. The precision-recall curves (bottom right) show an increase precision when including the context.

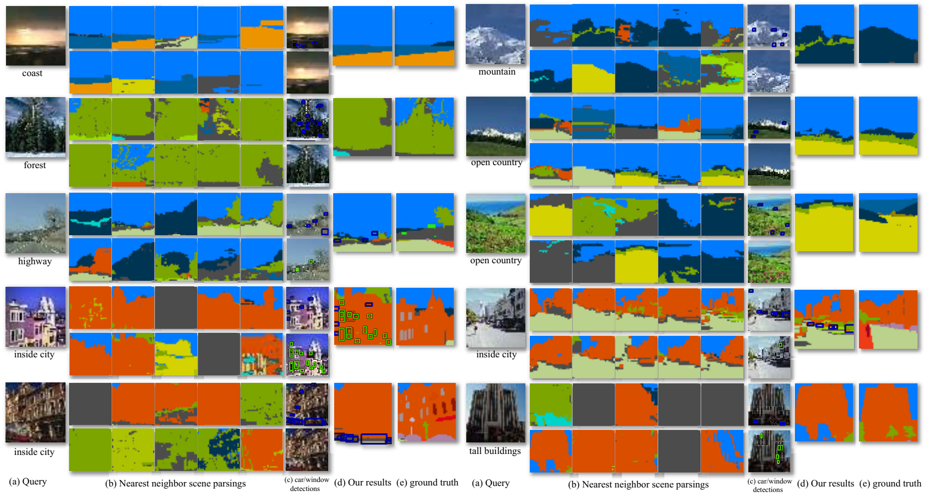

Fig. 5. Qualitative results after integrating scene parsings with object detections. Given query image (a) we infer its scene category, retrieve $N = 10$ scene parsings for the *stuff* categories present using our labeled training database, and detect candidate bounding (b) for the *thing* categories (c). Our algorithm picks the scene parsing/detections that in conjunction best explain the image as a whole using our prior (d). Finally, the ground truth segmentation (e).

7 Conclusion

We have presented a general framework to easily encode a rich variety of contextual rules to create object-level image priors ranging from simple object-class co-occurrences to higher order constraints at the scene level. We validated our framework connecting different priors to our simplified likelihood term and showed the role of different types of context regularizing noisy observations from the likelihood term. Despite using a very simple generative likelihood term, we observed considerable performance increases nearing that of CRF-based approaches in the MSRC set using co-occurrence relations. Furthermore, we demonstrate how to adapt our framework creating an end-to-end system that merges SIFT-flow scene parsings and object detections to eliminate contextually incoherent false detections as well as to pick the most contextually coherent parsing. With the advent of multiple algorithms to solve the problem of object recognition, a great number of solutions exist to understand images at the scene, object, segment, patch, and many other levels. Each feature and method presents advantages and disadvantages in different scenarios; our model serves as a catalyst to put these techniques to one unified framework.

References

1. Russell, B.C., Torralba, A., Liu, C., Fergus, R., Freeman, W.T.: Object recognition by scene alignment. In: Advances in Neural Info. Proc. Systems. (2007)
2. Torralba, A.: Contextual priming for object detection. IJCV **53** (2003) 153–167
3. Rabinovich, A., Vedaldi, A., Galleguillos, C., Wiewiora, E., Belongie, S.: Objects in context. In: ICCV. (2007)
4. Kumar, S., Hebert, M.: Discriminative random fields: A discriminative framework for contextual interaction in classification. In: ICCV. (2003)
5. Galleguillos, C., Rabinovich, A., Belongie, S.: Object Categorization using Co-Ocurrence, Location and Appearance. In: CVPR. (2008)
6. Desai, C., Ramanan, D., Fowlkes, C.: Discriminative models for multi-class object layout. In: ICCV. (2009)
7. Parikh, D., Zitnick, C.L., Chen, T.: From appearance to context-based recognition: Dense labeling in small images. In: CVPR. (2008)
8. Hoiem, D., Efros, A., Hebert, M.: Putting objects in perspective. In: CVPR. (2006)
9. Liu, C., Yuen, J., Torralba, A., Sivic, J., Freeman, W.T.: SIFT flow: dense correspondence across different scenes. In: ECCV. (2008)
10. Felzenszwalb, P., McAllester, D., Ramanan, D.: A discriminatively trained, multiscale, deformable part model. In: CVPR. (2008)
11. Zhu, C.S., Wu, N.Y., Mumford, D.: Minimax entropy principle and its application to texture modeling. Neural Computation **9** (1997)
12. Russell, B.C., Torralba, A., Murphy, K.P., Freeman, W.T.: LabelMe: a database and web-based tool for image annotation. IJCV **77** (2008) 157–173
13. Divvala, S., Hoiem, D., Hays, J., Efros, A., Hebert, M.: An empirical study of context in object detection. In: CVPR. (2009)
14. Sudderth, E., Torralba, A., Freeman, W.T., Willsky, W.: Learning hierarchical models of scenes, objects, and parts. In: ICCV. (2005)

15. Li, L.J., Fei-Fei, L.: What, where and who? classifying event by scene and object recognition. In: ICCV. (2007)
16. Li, L.J., Socher, R., Fei-Fei, L.: Towards total scene understanding: Classification, annotation and segmentation in an automatic framework. In: CVPR. (2009)
17. Parikh, D., Zitnick, C.L., Chen, T.: Unsupervised learning of hierarchical spatial structures in images. In: CVPR. (2009)
18. Berger, A.L., Della Pietra, S.D., Della Pietra, V.J.D.: A maximum entropy approach to natural language processing. Computational Linguistics **22** (1996) 39–71
19. Lazebnik, S., Schmid, C., , Ponce, J.: A maximum entropy framework for part-based texture and object recognition. In: ICCV. (2005)
20. Jeon, J., Manmatha, R.: Using maximum entropy for automatic image annotation. In: In Proc. CIVR. (2004) 24–32
21. Berger, A.: The improved iterative scaling algorithm: A gentle introduction. Technical report, CMU (1997)
22. Malouf, R.: A comparison of algorithms for maximum entropy parameter estimation. In: Proc. Sixth Conference on Natural Language Learning (CoNLL-2002). (2002) 49–55
23. Oliva, A., Torralba, A.: Modeling the shape of the scene: a holistic representation of the spatial envelope. IJCV **42** (2001) 145–175
24. 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)
25. Liu, C., Yuen, J., Torralba, A.: Nonparametric scene parsing: Label transfer via dense scene alignment. In: CVPR. (2009)