Factor Graphs for Region-based Whole-scene Classification

Matthew R. Boutell
CSSE Dept.
Rose-Hulman Inst. of Techn.
boutell@rose-hulman.edu

Jiebo Luo
Res. and Dev. Labs
Eastman Kodak Company
jiebo.luo@kodak.com

Christopher M. Brown
Dept. of Computer Science
University of Rochester
brown@cs.rochester.edu

Abstract

Semantic scene classification is still *a challenging problem in computer vision. In contrast to the common approach of using low-level features computed from the scene, our approach uses* explicit *semantic object detectors and scene configuration models. To overcome faulty semantic detectors, it is critical to develop a region-based, generative model of outdoor scenes based on characteristic objects in the scene and spatial relationships between them. Since a fully connected scene configuration model is intractable, we chose to model pairwise relationships between regions and estimate scene probabilities using loopy belief propagation on a factor graph. We demonstrate the promise of this approach on a set of over 2000 outdoor photographs, comparing it with existing discriminative approaches and those using low-level features.*

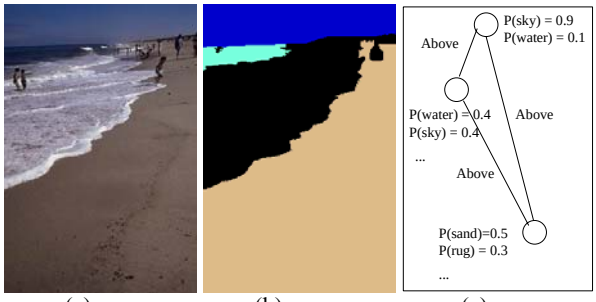
1. Introduction

Semantic (high-level) features have begun to bridge the "semantic gap" between pixels and image understanding. Scene classification using these features should be more accurate (given accurate detectors) than that using low-level features such as color or edges. While a scene classifier using reliable high-level features seems like "standing on the shoulders of giants," the key is how to handle computed semantic features that are imperfect.

Semantic feature detectors can be developed independently and employed selectively in relevant domains. Specifically, object and material (used interchangeably) detection features for natural environments are accurate enough to consider using in a practical system. Recent work using object detection for other tasks [2][5] has achieved some success using primarily object presence or absence alone as evidence. However, despite improvements, the detectors still make errors, presenting a continuing difficulty for this approach. The key question is how to handle imperfectly computed semantic features.

One way is to use a principled probabilistic inference system (vs. a rule-based one [5]) to classify a scene based on the presence or absence of certain semantic regions. Further, one could extract additional useful evidence from the input image, such as *spatial relationships between the detected semantic regions*, to improve scene classification.

Figure 1 shows an image; true material identities of key regions (color-coded); detector results, expressed as likelihoods that each region is labeled with a given material; and spatial relationships between the regions. The problem is how to determine which scene type best explains the observed, often imperfect, evidence. As humans, we can easily see that the evidence taken as a whole (Figure 1c), though ambiguous for each region individually, better fits a beach model than a field or city street model; our job is to train a system to do likewise.



(a) (b) (c)

Figure 1. (a) A beach scene. (b) Its manually-labeled materials. The true configuration includes *sky above water*, *water above sand*, and *sky above sand*. (c) The underlying graph showing detector results and spatial relations.

We present generative models for *scene configurations*, consisting of regions' identities and their spatial relations We show that such models can be used to classify medium to long-distance outdoor scenes, given semantic detectors and (pairwise) scene configurations. We compare this model with three other generative models and two discriminative models. The work's main novelty is its *explicit* use of spatial relations in building a *generative model* to parse a scene, distinguishing it from other work using semantic features [2][5].

2. Generative models for scenes

Our generative model is based on the concept of s*cene configurations,* consisting of two parts: the actual *spatial arrangement* of regions (edge labels in the graph of Figure 1c) and the *material configuration*, the identities of those regions (node labels in Figure 1c). We use the following

terminology to discuss each:

n: the number of distinct regions detected in the image

M: the small set of semantically critical materials for which detectors are used. M_j is the identity of region *j*.

R: the set of spatial relations (e.g., beside, below)

C: the set of configurations of materials in a scene

An upper bound on the number of scene configurations, |C|, is

$$|C| = |M|^n \cdot |R|^{\binom{n}{2}} \quad (1)$$

in a fully connected graph, although some of these enumerated spatial arrangements are inconsistent, e.g., *X above Y*, *Y above Z*, *X below Z*. In our experiments, for example, we have |M| = 10 materials of interest (the potential labels for a single region) and |*R*| = 7 spatial relations. An image with $n = 4$ regions has 10^4 material configurations and ${}_4C_2 = 6$ pairwise spatial relations yielding 7^6 spatial arrangements and a total of $10^4 \cdot 7^6 \approx 1.2$ trillion scene configurations.

Actually, in the inference phase, constraints such as fixed spatial arrangement mean there would be many fewer possible configurations; still the above example would have 10,000 possible material configurations.

2.1. Formalizing the problem of scene classification from configurations

We formalize the scene classification problem as follows. Let $S = \{Si\}$ be the set of scene classes considered: we used S = {beach, field, mountain, street, suburban, open-water}. Let $E = \{ E_1, E_2, \ldots, E_n\}$ be the detector evidence, one for each of the *n* regions. Each $E_j = \{E_{j1}, E_{j2}, \ldots, E_{j|M|}\}$, in turn, is a belief vector for the identity of region *j*. These likelihoods are computed using a list of which material detectors fired on that region and with what belief, as described in Section 3.3.

In this framework, we want to find the scene with *maximum a posteriori* (MAP) likelihood, given the evidence from the detectors, or $\text{argmax}_i\ P(S_i|E)$. By Bayes' Rule, this expands to

$$\arg\max_i P(S_i)P(E \mid S_i)/P(E)\cdot \quad (2)$$

P(E) is fixed at inference time, and the value of the argmax does not change when multiplying by a constant. The joint distribution of $P(E|S_i)$ with the set of scene configurations *C* is

$$\arg\max_i P(S_i)\sum_{c\in C} P(E,c \mid S_i) \quad . \quad (3)$$

Conditioning on *c* yields

$$\arg\max_i P(S_i)\sum_{c\in C} P(c \mid S_i)P(E \mid c,S_i)\cdot \quad (4)$$

2.2. Learning the model parameters

Learning $P(E|c,S_i)$ is relatively easily. As is standard with probabilistic models used in low-level vision, we assume that a detector's output on a region depends only on the object present in that region and not on other objects in the scene. Furthermore, we assume that the detector's output is independent of the class of the scene (again, given the object present in that region). This allows us to factor the distribution as

$$P(E \mid c,S_i) = P(E \mid c) = \prod_{j=1}^{n} P(E_j \mid m_j), \quad (5)$$

in which each factor on the equation's right-hand side describes a single detector's characteristics. These characteristics can be *learned* by counting detection frequencies on a training set of regions or *fixed* using domain knowledge. This distribution is used in the belief calculations given in Section 3.3.

Learning $P(c|S_i)$ is more difficult. At this coarse level of segmentation, even distant (with respect to the underlying image) nodes may be strongly correlated, e.g., sky and pavement in urban scenes. Thus, we must assume that the underlying graph of regions is *fully connected*, prohibiting us from factorizing the distribution $P(c|S_i)$, as is typically done in low-level vision problems. Fortunately, for scene classification, and particularly for landscape images, the number of critical material regions of interest, *n*, is generally small ($n < 7$ in the hand-labeled images): over-segmentation is rare because the material detectors can be imbued with the ability to merge regions. Thus a brute-force approach to maximizing the likelihood in Eq. (4) can be tractable.

One difficulty with learning and inference in this approach is that each feature and relation is discrete. Disparate materials such as *grass*, *sand*, and *foliage* cannot be parameterized on a continuum. Relations such as *above* and *enclosing* are discrete as well. Therefore, we learn the distributions of scene configurations by counting instances from the training set and populating matrices; these become the scene configuration factors in our factor graph.

2.3. Computing the spatial relationships

Singhal, *et al.* found that |*R*| = 7 distinct spatial relations were sufficient to model the relationships between materials in outdoor scenes; we use the same relations (*above*, *far_above*, *below*, *far_below*, *beside*, *enclosed*, and *enclosing*) and the same techniques to compute these relations [8].

For simplicity, once we have computed the spatial relations between two regions, the current model ignores the shape and size of the regions, adjacency information about the regions, and occlusions that cause regions to be split. While any of these may be useful features in a full-scale system, we ignore them in this work.

2.4. Graphical model

While all graphical models may have the same representational power, not all are equally suitable for a given problem. Factor graphs [3] are a model that encodes functions, in particular, probability distributions, over a set of variables using a bipartitite graph, consisting of variable nodes and factor nodes. They are more convenient than MRF's, for example, for expressing higher-order interactions.

We use a factor graph (Figure 2) so that we can model interactions between the scene type and various region configuration. There are variables for the scene class (1 variable, S, defined above) and the region identities (*n*; one for each region, collectively, they are the scene configuration C). After the evidence propagates through the network, we find the scene class by taking the value with the highest marginal probability at the scene node.

The factors in the graph encode the compatibilities between the scene type, the scene configurations, and the detector evidence, E, given in Eq. (4). The prior factor encodes *P(S)*, the prior distribution of scene types across the image population. We currently do not take advantage of prior information and simply use a flat prior, but priors could be calculated and modeled in the future. The detector factors shown at the bottom of Figure 2 encode the detector likelihoods (Section 3.3); there is one factor for each region. The number and type of these variables and factors do not change throughout our experiments.

We experiment with a few methods to enforce the compatibility, $P(c|S_i)$, between the scene and set of regions, as given in the set of *scene-to-region* factors. The exact topology is *dynamic*, depending on the number of regions in the image. Furthermore, for a given number of regions, we can change the network topology to enforce or relax independence assumptions and observe the effects of these assumptions. We have used four generative models of *between-region dependence, given the scene type*:

1. *Exact*: Generative model in which the full scene (material and spatial) configuration is taken as a single, dependent unit.
2. *Spatial Pairs:* Same as *Exact*, but approximate version using pairwise spatial relationships).

The next two are used as baselines for comparison:

3. *Material Pairs*: Dependent only on the pairwise co-occurrence of materials.
4. *Independent*: Each region is independent.

2.5. Factor graph variations for between-region dependence

We now discuss the scene-to-region factors and the factor graph topology for each option.

2.5.1 **Exact.** Recall that we cannot factor the distribution $P(c|S_i)$ into individual components because of the strong dependency between regions. We model it with a fully-connected structure. If we want an exact MAP solution to the distribution (Eq. 1), we must use the factor graph shown in Figure 2a.

Here, the single spatial configuration encodes the conditional probability, $P(c|S,)$, the distribution of all region identities for a given scene. This is implemented as a $(n+1)$-dimensional matrix. We first find a matrix of counts, $N(s,c), s \in S$ by counting instances of each in the training set, then normalize it such that $\forall s \in S, \sum_{c \in C} N(s,c) = 1$, to make it a conditional probability. This matrix has $|M|^n \cdot |S|$ elements, one for each material configuration and for each scene class.

The main benefits of this model are both due to its loopless topology: it can give an exact solution and it provides for efficient inference. However, it suffers from drawbacks. The distribution $P(c|S_i)$, is sparsely populated: the number of training images (call it $|T|$) is typically less than the number of entries $|M|^n \cdot |S|$, sometimes much less. (Consider that $|T|$=1000 is considered large in the literature, and that for M=10 and S = 6, a factor for a matrix with 5 regions has 600,000 entries.) The sparseness is exacerbated by correlation between objects and scenes, causing some entries to receive many counts and most entries to receive none.

Recall that each feature and relation is discrete, so we cannot interpolate between training examples to smooth the distribution (as can be done with parameterized distributions such as mixtures of Gaussians). In [1], Boutell *et al.* proposed a graph-based smoothing technique to deal with this sparse distribution problem.

2.5.2 **Spatial Pairs.** Even after smoothing, using the full joint distribution has a major limitation keeping it from being used in practice: the distribution of regions is also a function of the spatial arrangement of the regions in the image, which includes the number of regions in the image. For example, the distribution of regions in the arrangement: "*A beside B*, *B beside C*, and *A beside C*" (3 horizontally arranged regions) is distinct from those in the arrangement "*A above B*, *B above C*, and *A above C*" (3 vertically stacked regions), which is distinct from those in the arrangement of 4 vertically stacked regions A, B, C, and D.

However, approximating the joint distribution as the product of pairwise distributions, as shown in Figure 2b, means we only have to learn *pairwise* relationships. Each conditional probability, $P(m_i \text{ rel } m_j|S)$, for spatial relation *rel*, is only a 3D matrix with only $|M|^2 \cdot |S|$ elements. Furthermore, there are only |R|=7 of them, and because each training example contains multiple pairwise relations,

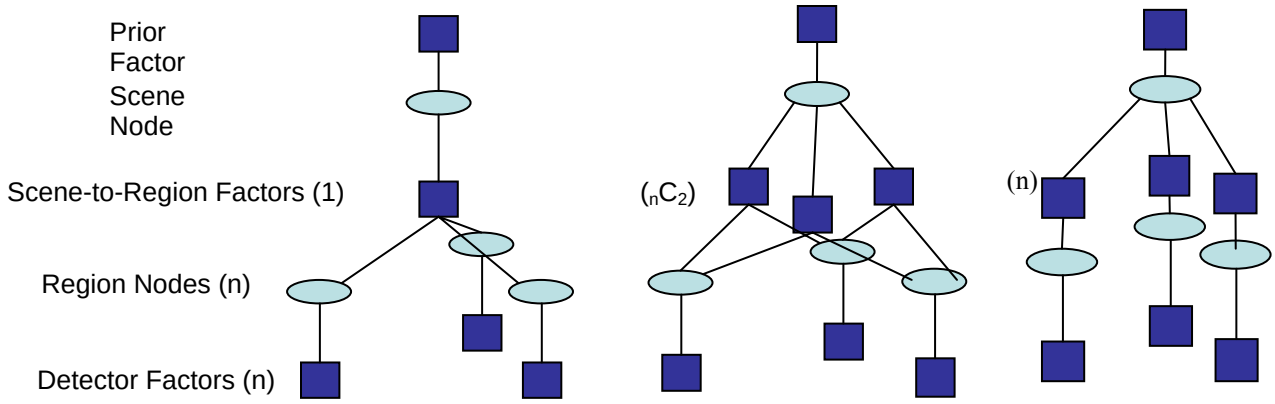


Figure 2. Factor graphs modeling various independence assumptions for $n = 3$ regions. The difference between the models is the number and type of Scene-to-Region factors, described in the text. (a) *Exact* model. Due to its tree structure, we can perform exact inference on it. However, the complexity of the model is hidden in the spatial configuration factor; learning it is problematic. (b) *Spatial pairs* model. While it is not exact due to the loops, each spatial factor's parameters are easier to learn than the exact model's joint one. Furthermore, its dynamic structure allows it to work on any image. (c) *Independent* model. This is equivalent to a tree-structured Bayesian network.

the pairwise distributions will be much more densely-populated. For example, pairwise relations such as "sky above grass" occur very often, compared to full configurations such as "sky above foliage above pavement enclosed by grass".

This model does contain loops, and so encounters the typical problems with loopy belief propagation: it can be slow and may not converge. However, we have found n tends to be small in practice, so speed is not an issue, and empirically it converges most of the time.

Furthermore, we can create the graph structure on the fly once the number of regions is known, and once each pairwise relationship is modeled (*X above Y, X beside Y, X enclosed by Y* and their inverse relationships), any spatial arrangement containing those relations can be modeled. Thus, for the set of 1400 images described above, we would need to learn only 7 factors and use one dynamically-created network using combinations of these densely-populated factors rather than needing 133 networks with sparsely-populated factors.

2.5.3 **Material Pairs.** One may argue that the spatial relationships do not provide information about relative position; perhaps they just encode which materials tend to co-occur. This model is designed to test this hypothesis. We use the same network topology as shown in Figure 2b, but replace the spatial relation factors with co-occurrence factors. We obtain these co-occurrence factors by counting the number of times a pair of materials i and j occur together in the training set for a scene S. Normalizing the matrix gives $P(c_i \text{ co-occur } c_j|S)$.

2.5.4 **Independent.** In our final model, we consider each of the n regions independently of each other region given the scene, $P(c \mid S_i) \approx \prod_n P(c_n \mid S_i)$. This yields the factor graph shown in Figure 2c. Each scene-to-region factor is just a normalized 2D matrix of counts of the number of times each material occurs in each scene type.

3. Semantic detectors

We define semantic, or *high-level*, features to be labeled regions. For outdoor scenes, we use 10 labels: *sky, cloud, grass, foliage, sand, snow, water, rocks, pavement*, and *buildings*. A region with ambiguous identity usually has a low belief value, and may also have multiple labels. In this study, we use high-level features generated from 3 types of detectors: (1) output from *actual* object and material detectors, (2) output from *simulated* detectors, and (3) output from *best-case* detectors (hand-labeled regions).

3.1. Best-case (hand-labeled) detectors

Images in which the semantically-critical regions have been hand-labeled form an integral part of this work. First, we use them for training. Specifically, we learn from them the distribution of which objects and materials appear in which scene and in what configurations. Second, we use them as an upper bound on material detectors' performance. Third, we can perturb the region labels assigned by a human to simulate faulty detectors with various error characteristics (Section 3.4).

Meaningful regions are produced by hand from the output of automatic segmentation routines (Figure 3). The region bounadaries are determined automatically, but hand-labeling can help compensate for oversegmenation.

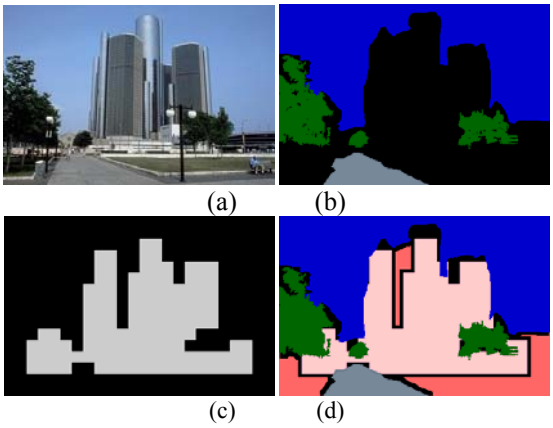


Figure 3. Process of hand-labeling images. (a) A street scene. (b) Output from the segmentation-based labeling tool. (c) Output from a manmade object detector. (d) Combined output, used for learning spatial relation statistics.

3.2. Actual detectors

Each of our actual detectors is based on color and texture features, similar to the common approach in the literature [6][9] (Figure 4).

All of the detectors are run independently. After this, the region maps are aggregated, inducing a segmentation map upon the image (Figure 5). Regions detected as multiple materials get multiple labels with likelihoods of each material calculated as in Section 3.3.

This technique for material detection is a bottom-up strategy because no spatial model is initially imposed and the detectors work for general outdoor scenes. . Detection is completely automatic in the testing phase. While some individual material detectors have very good accuracies (e.g., sky, 95%), others have substantially lower performance (e.g., water and rock below 70%), primarily due to high false positive detection rates.

3.3. Combining evidence for a region from multiple detectors

Each region is processed independently in the following manner. Let D be the set of detectors, $D = \{D_1, D_2, \ldots D_{|M|}\}$. Each detector D_i outputs a belief value $\lambda_{Di}(B)$ in [0,1] for region B, where $\lambda_{Di}(B)=0$ means the detector did not fire and $\lambda_{Di}(B)=1$ means it fired with full confidence. R's initial evidence is the vector $\lambda_D(B)$ of those belief values. Note we use non-binary beliefs, or "virtual evidence", as defined by Pearl [7].

When combining the evidence, we need to account for faulty detectors and detectors of different reliability. We use a two-level Bayesian network (Figure 6). Define the *characteristics* of detector D on a set of materials M to be the set of conditional probabilities $\{P(D \mid m_i) : m_i \in M\}$. They include both *true positive rates* (recall) and *false positive rates* (e.g., Table 1). Such characteristics can be learned by counting performance of detectors on a validation set.

Figure 4. Process of material detection, shown for the foliage detector. (a) Original image. (b) Pixel-level belief map. (c) Output of the detector. In (b) and (c), brightness corresponds to belief values.

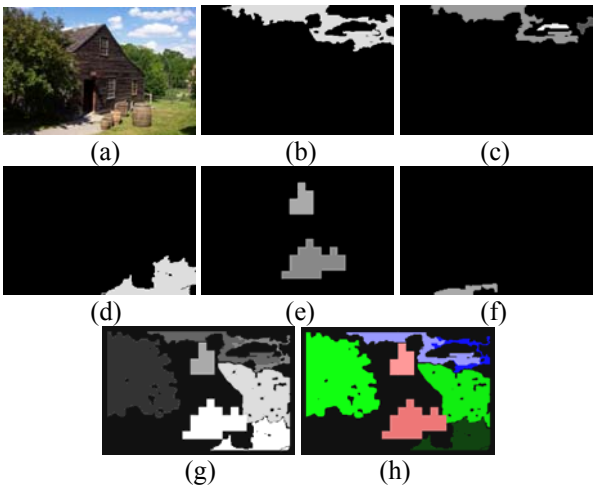


Figure 5. Aggregating results of individual material detectors for an image (a) Original image. (b) Blue-sky. (c) Cloudy sky. (d) Grass. (e) Manmade. (f) Sand. The foliage detection result from Figure 4 is also used. Other detectors gave no response. (g) The aggregate image with 7 detected regions. (h) Pseudo-colored aggregate segmentation map.

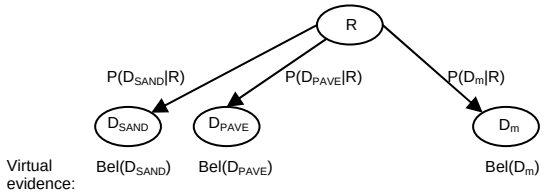


Figure 6. Bayesian network showing relationship between regions and detectors.

We have a detector for each material of interest. One node per detector is linked to the region, *B*, as shown in Figure 6. Input to the Bayesian network consists of virtual evidence ($\lambda_D(B)$ described above) at the leaf nodes. Note that this graph's root node corresponds to a specific *region* in the image.

The beliefs generated by the individual material detectors are fed into the leaf nodes of the network and propagated to the root node, which outputs a composite material belief for that region. We follow Pearl's

treatment and notation (Section 2.2.2 of [7]), using λ for beliefs.

Table 1. Characteristics of sand detector.

$P(D_{sand}\|M)$	True material (M)
0.10	Background (unmodeled)
0.01	Bluesky
0.05	Cloudysky
0.01	Foliage
0.05	Grass
0.10	Manmade
0.05	Pavement
0.05	Rock
0.90	Sand
0.05	Snow
0.05	Water

The combined belief of each material being the true material, given the set of detectors firing is given by:

$$\lambda(R) = \alpha \prod_i \lambda_{D_i}(R) = \alpha \prod_i M_{D|R_i} \lambda(D_i) \quad (6)$$

where α is a normalizing constant and A is a probability matrix as defined by Pearl [7] as $A_{y|x} \equiv P(y \mid x)$. More specifically, the $(i,j)^{th}$ position in $A_{y|x} \equiv P(y_j \mid x_i)$.

These beliefs, *λ(B)*, are passed on to the remainder of the network, i.e. by attaching the subgraph to each material leaf in the factor graphs shown in Figure 2).

3.4. Simulating Faulty Detectors for a Region

While we have actual detectors, we are also interested in determining the scene models' utility on a wider range of detector performance. Using the hand-labeled regions, we can simulate detector responses for each region.

We simulate detectors that have a range of accuracies and associated confidences by allowing them to (1) fire with certain probability, and (2) give the right answers with certain probability if fired. We anticipate that "better" detectors (via simulation) will outperform the actual ones.

4. Experimental results

The three image databases we used (Table 2) were chosen such that they could be described in terms of the objects or materials expected to be present in them. For example, an image with sky, water, and sand is a beach. Data set D2 included all of the images from D1 with a specific spatial arrangement: we ignored the *far* modifier and the enclosure relations (which occur rarely) and focused on the single spatial arrangement occurring most often in training: the 165 images with a vertically stacked structure, R_1 *above* R_2, R_2 *above* R_3, and R_1 *above* R_3.

Table 3 gives descriptions of various scene classes. In general, the images chosen did not have a dominant main subject. This is legitimate; for example, a close-up image of a horse in a field would be considered a "horse" image, not a "field" image. Future work includes investigating how to adapt the framework for such images. The images are taken from a mix of personal images, Corel photos, and images from the Internet. Each set has about the same number of images from each of the six classes.

Table 2. Image sets.

Name	**Size**	**Classes and comments**
D1	937	Regions are hand-labeled; we also ran actual detectors on D1.
D2	165	Subset of D1 having three vertically arranged regions.
D3	1153	Drawn from same sources, but non-overlapping with D1. Used actual detectors.

Table 3. Scene class descriptions.

Class	**Description**
Beach	At least 5% each of water, sand, and sky
Field	No aerial view, not cluttered with trees ("open")
Mountain	Open, whole mountains, mid-range view
Street	Urban, plus must contain pavement
Suburban	Includes pictures of houses and yards and recreation areas (parks)
Open-water	Picture taken primarily of water from a boat (vs. those taken from land, which are beaches or coastal scenes).

Data set D1 was automatically segmented and hand-labeled, as described in Section 3.1. To simulate actual detectors, which are faulty, we randomly perturbed the labels to create simulated detector responses (Section 3.4).

Pairwise spatial relations in our model are encoded as probability density functions of the two regions and the scene for each of the seven spatial relations; we learn these, not by using a maximum likelihood technique, but by counting occurrences directly from D1. The material detectors were trained on image sets not used in this study.

4.1. Discriminative approach

We compare our work against two discriminative approaches, in which a classifier is trained to learn the distribution *P(S|E)* directly. One direct way to encode the location of the semantic regions (objects) and the spatial relations between them is to use a grid-based approach. We divide the image into 49 blocks using a 7x7 grid and compute the belief in each material for each block by weighting each region's beliefs output by the combined detectors (λ(R) in Eq. 6) by the percentage of the area the region occupies in a block. For *m*=10 materials, this always yields a 490-dim. feature vector, regardless of the

number of image regions. We then train a set of support vector machines [11] for doing multi-class classification.

We also compare the high-level features with a second discriminative approach using low-level features: spatial color moment features [10], also extracted using a 7x7 grid and classified using a set of SVMs [11].

4.2. Spatial Pairs Model

To show the benefit of spatial relations, we compare them against the baseline models mentioned earlier: *Material Pairs* and *Independent*. Table 4 shows classification accuracies for a number of scenarios. In each case, we learned from image set D1. We tested either on D1 using cross-validation (CV) to avoid biasing our results, or on D3. Figure 7 is an expansion of row 2, comparing performance across the range of simulated detectors.

Table 4. Accuracy (%) comparison between techniques.

Model:	Generative			Discriminative	
Test set	*Spatial Pairs*	*Mat. Pairs*	*Indep*	High-level features	Low-level features
D1 (best-case, CV)	94.5	91.7	92.2	93.1	78.0 (doesn't use detectors)
D1 (sim. det., CV)	80-93	70-89	67-90	74-91	
D1 (real det., CV)	51.0	47.2	45.6	75.6	
D3 (real detectors)	46.0	41.5	40.0	54.2	67.4

In our experiments, classification using the *Spatial Pairs* model always outperformed those using the other generative models, *Material Pairs* and *Independent*, showing that spatial information does help distinguish scenes. The marginal difference between the two increased as the accuracy of the detectors decreased. This is not surprising, given that spatial relations are expected to mitigate the effects of faulty detectors, more so when they are more faulty. Figure 8 contains example images (and their corresponding segmentations) for which the *Spatial Pairs* model yielded an improvement over one of the baseline generative models.

Spatial relations outperformed the discriminative models *when the simulated detectors were good enough* (including in the best-case). Of course, if the detectors are inaccurate enough, then no amount of spatial information will help due to the abundance of noise. When using actual detectors, the discriminative approach outperformed the generative approach using the same feature set. This could be due to the following reasons. First, the discriminative models, by their nature, are trained to classify, whereas generative models can perform classification, but only when the joint distribution between the class and the features is modeled. Second, while the generative and discriminative models both encode spatial information (explicitly and implicitly, respectively), only the discriminative model encodes relative region *size* through the number of blocks occupied by the region and to a lesser degree the *shape* of the region. For example, if buildings occupy a large portion of a street scene, but a smaller portion of a suburban scene, the discriminative model can differentiate the two, whereas the generative model, having no "region area" evidence factor, cannot.

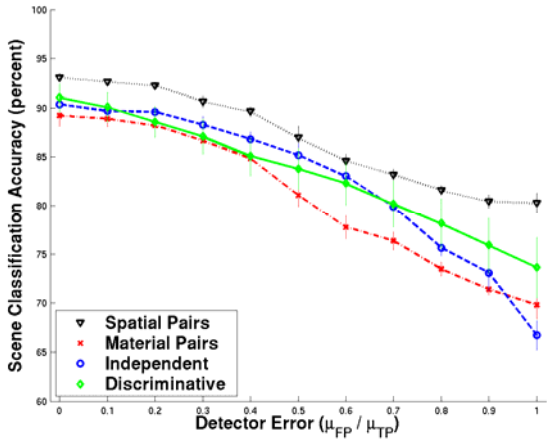


Figure 7. Classification accuracy of the methods as a function of simulated detector error rate. We repeated each simulation 10 times and report the mean accuracy. The error bars are for the standard error between test runs.

Figure 8. Examples of images and segmentations for which the *Spatial Pairs* model gave correct results while the baseline model failed. See text for details.

4.3. Comparison Between All Methods

As stated earlier, the *Exact* model is limited in that it can only be applied when the spatial arrangement of the regions is fixed. Therefore, when comparing this model with the others on a single data set, we use data set D2, which includes the 165 vertically-stacked images in D1. Furthermore, to ensure the model is populated densely enough, we must ignore the *far* modifier and the enclosing relations. We also compare it with the other generative models (*Material Pairs* and *Independent*) and the two discriminative models, using leave-one-out cross-

validation on D2 (Figure 9).

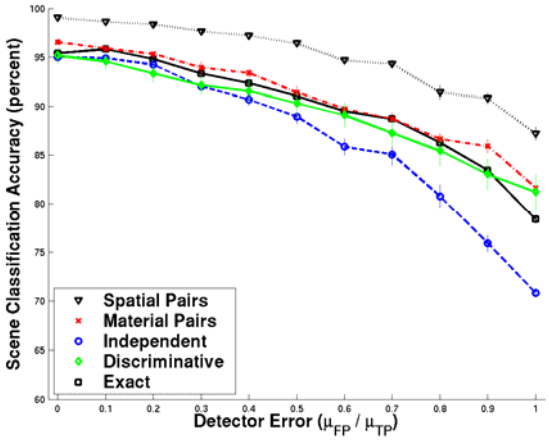


Figure 9. Comparison between accuracy obtained using the *Spatial Pairs* model, the *Exact* model, and the discriminative model using high level features for the range of simulated detector error rates.

On this small data set, when high-level features are used, the *Spatial Pairs* model outperforms the discriminative and the baseline generative models. The reduction in the discriminative model's accuracy is because the discriminative model is undertrained given its dimensionality and the available training data. High-level features also outperform the low-level features (75.2% accuracy, not shown on graph) except when the simulated detectors are highly inaccurate, which is expected, given our previous results on the larger data set (Table 4).

Figure 9 shows that the approximate model also *outperforms* the exact model. This is counter-intuitive because when we approximate, we are losing information. Two types of images are involved. The first type, those images classified correctly by the *Exact* model, but not by *Spatial Pairs*, include images with full configurations that are repeated, though rarely, in the training set, but consist of pairwise relations seen more often in other scenes.

The second type, those images classified correctly by the approximate model, but not by the exact model, include those "singleton" images containing a unique configuration of materials. This is due to the fact that the model is undertrained, i.e., the distributions of scene configurations are too sparsely populated. Because there are more images of the second type, the accuracy of the exact model is lower on the training set.

5. Discussion and conclusions

We use material detectors and generative scene configuration models to parse the scene context, showing experimental comparison with discriminative approaches using both semantic and low-level features. When the semantic detectors are not good enough, generative models do not yield high classification accuracy.

Generative models do offer a number of other advantages. First, the systems can be highly modular. Second, generative models usually offer much insight to the relationship between various factors involved in the problem. Third, they operate as well as discriminative models when there is a shortage of labeled training data., which is often the case with semantic scene classification. Fourth, they can express confidence in their judgment. Finally, generative models can operate in the face of missing cues (e.g., allowing an option of turning off some object detectors on the fly to gain speed).

Others, e.g. Kumar and Hebert [4], have used factor graphs to model pairwise relations. However, their model is used to improve the region labeling only, while ours also yields a classification of the scene as a whole.

Future work should improve the accuracy of the semantic detectors, since our simulations show that our approach yields higher accuracy, given good enough detectors. An interesting idea is to encode area and shape in the graphical model. Finally, we would like to apply our model to classify indoor scenes using appropriate detectors for typical indoor objects.

6. References

[1] M. Boutell, J. Luo, and C. M. Brown, Learning spatial configuration models using modified Dirichlet priors. In *Proc. Workshop on Statistical Relational Learning (in conjunction with ICML)*, 2004.

[2] J. Fan, H. Luo, and Y. Gao. Learning the semantics of images by using unlabeled samples. In *Proc. Conf. on Computer Vision and Pattern Recognition,* June 2005.

[3] F. R. Kschischang, B. H. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47, 2, (Feb. 2001).

[4] S. Kumar and M. Hebert. A hierarchical field framework for unified context-based classification. In *Proc. IEEE Int'l Conf on Comp. Vis*, 2005.

[5] P. Mulhem, W. K. Leow, and Y. K. Lee. Fuzzy conceptual graphs for matching images of natural scenes, In *Proc. Int'l Joint Conf. on Artific. Intell.*, (2001), 1397-1404.

[6] M.R. Naphade and T.S. Huang. A probabilistic framework for semantic video indexing, filtering, and retrieval. *IEEE Transactions on Multimedia*, 3, 1 (2001), 141-151.

[7] J. Pearl, editor. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann Publishers, 1988.

[8] A. Singhal, J. Luo, and W. Zhu. Probabilistic spatial context models for scene content understanding. In *Proc. Conf. on Computer Vision and Pattern Recognition*, 2003.

[9] J. R. Smith and C.-S. Li. Image classification and querying using composite region templates. *Computer Vision and Image Understanding*, 75, 1/2 (1999), 165–174.

[10] A.Vailaya, M.Figueiredo, A.Jain, and H.-J. Zhang. Content-based hierarchical classification of vacation images. In *Proc. Conf. Multimedia Systems 1999.*

[11] N. Vapnik. The Nature of Statistical Learning Theory. Springer, 1995.