Category-Independent Object Proposals with Diverse Ranking

Ian Endres, *Student Member, IEEE,* and Derek Hoiem, *Member, IEEE*

Abstract—We propose a category-independent method to produce a bag of regions and rank them, such that top-ranked regions are likely to be good segmentations of different objects. Our key objectives are completeness and diversity: every object should have at least one good proposed region, and a diverse set should be top-ranked. Our approach is to generate a set of segmentations by performing graph cuts based on a seed region and a learned affinity function. Then, the regions are ranked using structured learning based on various cues. Our experiments on BSDS and Pascal VOC 2011 demonstrate our ability to find most objects within a small bag of proposed regions.

Index Terms—Vision and Scene Understanding, Segmentation, Object Recognition

1 Introduction

HUMANS have an amazing ability to localize objects without recognizing them. This ability is crucial because it enables us to quickly and accurately identify objects and to learn more about those we cannot recognize.

In this paper, we propose an approach to give computers this same ability for category-independent localization. Our goal is to automatically generate a small number of regions in an image, such that each object is well-represented by at least one region. If we succeed, object recognition algorithms would be able to focus on plausible regions in training and improve robustness to highly textured background regions. The recognition systems may also benefit from improved spatial support, possibly leading to more suitable coordinate frames than a simple bounding box. Methods are emerging that can provide descriptions for unknown objects [1], [2], but they rely on being provided the object's location. The ability to localize unknown objects in an image would be the first step toward having a vision system automatically discover new objects.

Clearly, the problem of category-independent object localization is extremely challenging. Objects are sometimes composed of heterogeneous colors and textures; they vary widely in shape and may be heavily occluded. Yet, we have some cause for hope. Studies of the human visual system suggest that a functioning object localization system can exist in the absence of a functioning object identification system. Humans with damage to temporal cortex frequently exhibit a profound inability to name objects presented to them, and yet perform similar to healthy controls in tasks that require them to spatially manipulate objects [3]. Many objects are roughly homogeneous in appearance, and recent work [4] demonstrates that estimated geometry and edges can often be used to recover occlusion boundaries for free-standing objects. While we cannot expect to localize every object, perhaps we can at least produce a small bag of proposed regions that include most of them.

Our strategy is to guide each step of the localization process with estimated boundaries, geometry, color, and texture. First, we create seed regions based on the hierarchical occlusion boundaries segmentation [4]. Then, using these seeds and varying parameters, we generate a diverse set of regions that are guided toward object segmentations by learned affinity functions. Finally, we take a structured learning approach to rank the regions so that the top-ranked regions are likely to correspond to different objects. We train our method on segmented objects from the Berkeley Segmentation Dataset (BSDS) [5], and test it on BSDS and the Pascal 2011 segmentation dataset [6], [7]. Our experiments demonstrate our system's ability for category-independent localization in a way that generalizes across datasets. We also evaluate the usefulness of various features for generating proposals and the effectiveness of our structured learning method for ranking.

2 Related Work

Category-Dependent Models: By far, the most common approach to object localization is to evaluate a large number of windows (e.g., [8], [9]), which are found by searching naively over position and scale or by voting from learned codewords [10], [11], distinctive keypoints [12], [13], or regions [14]. These methods tend to work well for objects that can be well-defined according to a bounding box coordinate frame when sufficient examples are present. However, this approach has some important drawbacks. First, it is applicable

• *I. Endres and D. Hoiem are at the Department of Computer Science, University of Illinois at Urbana-Champaign.*

only to trained categories, so it does not allow the computer to ask "What is this?" Second, each new detector must relearn to exclude a wide variety of textured background patches and, in evaluation, must repeatedly search through them. Third, these methods are less suited to highly deformable objects because efficient search requires a compact parameterization of the object. Finally, the proposed bounding boxes do not provide information about occlusion or which pixels belong to the object. These limitations of the category-based, window-based approach supply some of the motivation for our own work. We aim to find likely object candidates, independent of their category, which can then be used by many category models for recognition. Our proposed segmented regions provide more detail to any subsequent recognition process and are applicable for objects with arbitrary shapes.

Segmentation and Bags of Regions: Segmentation has long been proposed as a pre-process to image analysis. Current algorithms to provide a single bottom-up segmentation (e.g., [15], [16]) are not yet reliable. For this reason, many have proposed creating hierarchical segmentations (e.g., [17], [4], [18]) or multiple overlapping segmentations (e.g., [19], [20], [21], [22]). Even these tend not to reliably produce good object regions, so Malisiewicz et al. [20] propose to merge pairs and triplets of adjacent regions, at the cost of producing hundreds of thousands of regions. In our case, the goal is to segment only objects, such as cars, people, mugs, and animals, which may be easier than producing perceptually coherent or semantically valid partitionings of the entire image. This focus enables a learning approach, in which we guide segmentation and proposal ranking with trained classifiers.

An alternative approach is to attempt to segment pixels of foreground objects [23] or salient regions [24], [25]. However, these approaches may not be suitable for localizing individual objects in cluttered scenes, because a continuous foreground or salient region may contain many objects.

Two concurrent works have also considered generating object proposals as a preprocess for subsequent stages of object recognition. First, Alexe et al. [26] consider an "objectness" measure over bounding boxes, which they use to bias a sampling procedure for potential object bounding boxes. This method aims to be fast, on the order of several seconds per image, which restricts them to a less expressive bounding-box based representation. Alternatively, Carreira and Sminchisescu [27] consider a similar region proposal and ranking pipeline to ours. Regions are proposed by sampling points from a grid on the image which are used to seed the foreground color model of a segmentation. The border of the image is used to seed the background, and a per-pixel segmentation is generated with a graph-cut over simple color cues. The resulting regions are ranked through classification based on gestalt cues with a simple diversity model. Our approach instead guides segmentation with a learned affinity function, rather than setting the image border to background. We also differ in our structured learning approach to diverse ranking.

To summarize our contributions: 1) we incorporate boundary and shape cues, in addition to low-level cues to generate diverse *category-independent* object region proposals, and 2) introduce a trained ranking procedure that produces a small diverse set of proposals that aim to cover *all* objects in an image. We thoroughly evaluate each stage of the process, and demonstrate that it can generalize well across datasets for a variety of object categories.

3 Overview of Approach

Since our goal is to propose candidates for *any* object in an image, each stage of our process must encourage diversity among the proposals, while minimizing the number of candidates to consider. Our procedure is summarized in Figure 1. To generate proposals for objects of arbitrary shape and size, we adopt a segmentation based proposal mechanism that is encouraged to only propose regions from objects.

Rather than considering only local color, texture, and boundary cues, we include long range interactions between regions of an image. We do this by considering the affinity for pairs of regions to lie on the same object. This set of regions is chosen from a hierarchical segmentation computed over occlusion boundaries. To generate a proposal, we choose one of these regions to seed the segmentation, and compute the probability that each other region belongs to the same object as this seed. The affinities are then transferred to a graph over superpixels from which we compute segmentations with a variety of parameters. By computing the affinities over regions first and then transferring them to superpixels, we get the benefit of more reliable predictions from larger regions while maintaining the flexibility of a superpixel based segmentation. After repeating this process for all seed regions, we obtain an initial bag of proposals.

In our effort to discover a diverse set of objects, our proposal mechanism may generate many redundant or unlikely object candidates. In both cases, we would like to suppress undesirable proposals, allowing us to consider better candidates first. This motivates a ranking procedure that provides an ordering for a bag of proposals which simultaneously suppresses both redundant and unlikely candidates. We can then uncover a diverse set of the good object proposals with far fewer candidates.

Our ranker incrementally adds proposals, from best to worst, based on the combination of an object appearance score and a penalty for overlapping with previously added proposals. By taking into account the overlap with higher ranked proposals, our ranker ensures that redundant regions are suppressed, forcing the top ranked regions to be diverse. This is especially important in images with one dominant object and several "auxiliary" objects.

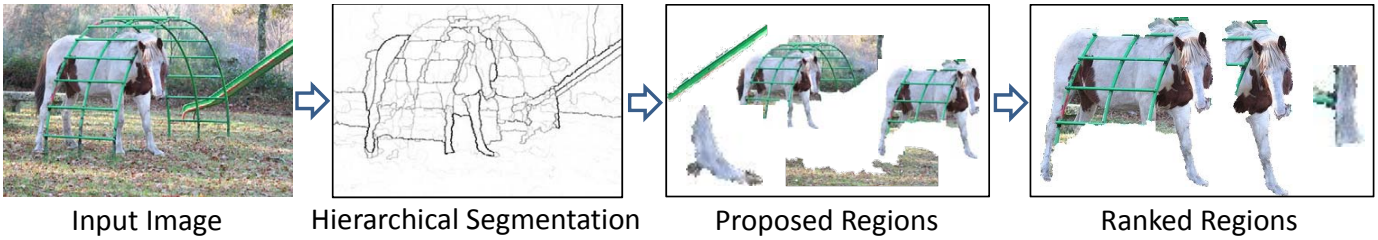


Fig. 1. Our pipeline: compute a hierarchical segmentation, generate proposals, and rank proposed regions. At each stage, we train classifiers to focus on likely object regions and encourage diversity among the proposals, enabling the system to localize many types of objects. See section 3 for a more detailed overview.

4 Proposing Regions

We first generate a large and diverse bag of proposals that are directed to be more likely to be object regions. Each proposal is generated from a binary segmentation, which is seeded with a subregion of the image. This seed is assumed to be foreground, and a segmenter selects pixels likely to belong to the same foreground object as the seed.

4.1 Hierarchical Segmentation

We use regions and superpixels from a hierarchical segmentation as the building blocks for our proposal mechanism. To generate the hierarchical segmentation, we use the output of the occlusion boundary algorithm from Hoiem et al. [4]. The occlusion boundary algorithm outputs four successively coarser segmentations, with probabilities for occlusion and figure/ground for each boundary in the segmentation. From each segmentation, we compute a probability of boundary pixel map and a figure/ground probability pixel map, and then average over the segmentations. Then, we create our hierarchical segmentation with agglomerative grouping based on boundary strength, as in [17], and we use the boundary strength and figure/ground likelihoods as features.

4.2 Seeding

A seed serves as the starting point for an object proposal. The appearance and boundaries around the seed are used to identify other regions that might belong to the same object. Seeds are chosen from the hierarchical segmentation such that they are large enough to compute reliable color and texture distributions ($\geq 20 * 20$ pixels). This results in about 300 seed regions per image. Also, we remove regions with boundaries weaker than 0.005 , since these are likely to just be a portion of a larger region. Stronger boundaries also facilitate the use of boundary cues to determine the layout of the object with respect to the regions.

4.3 Generating Segmentations

To generate a proposal, we infer a foreground / background labeling $\mathbf{l}, l_i \in \{0, 1\}$ over superpixels. Given a seed region, defined by a set of superpixels S, we construct a CRF that takes into account each superpixel's affinity for the seed region and the probability of boundaries between adjacent superpixels:

$$P(\mathbf{l}|X, S, \gamma, \beta) \propto \exp\left(\sum_i f(l_i; S, X, \gamma) + \beta \sum_{\{i,j\} \in N} g(l_i, l_j; X)\right) \quad (1)$$

Here, $f(l_i; S, X, \gamma)$ is the superpixel affinity term, inferred from image features X, and $g(l_i, l_j; X)$ is the edge cost between adjacent superpixels (defined by set of neighbors N). This CRF is parametrized by the foreground bias γ and the affinity/edge trade-off β. By varying these parameters for each seed, we can produce a more diverse set of proposals. We choose seven γ values uniformly from $[-2, 1]$, and eight β values spaced logarithmically from $[0, 10]$. These ranges were selected on the training set to give the best tradeoff between maximizing recall and minimizing the number of proposals generated.

4.3.1 *Region Affinity*

To compute the superpixel affinity $f(l_i; S, \mathcal{I}, \gamma)$, we first compute the affinity between the seed S and each region R in the hierarchical segmentation, and then transfer these region predictions to individual superpixels. For examples of superpixel affinities for different seeds, see Figure 3. We learn the probability that region R and seed S lie on the same object ($P(l_R|S, \mathcal{I})$) with a boosted decision tree classifier. Positive training examples are generated from pairs of regions that lie on the same object. Negative examples use pairs with one region lying on an object, and the other region lying on another object or the background.

Features: The classifier uses features for cohesion, boundary, and layout, as summarized in Table 1. *Cohesion* is encoded by the histogram intersection distances of color and texture (P1). *Boundary cues* are encoded by considering the cost to pass across boundaries from one region to the other. This path across boundaries is the straight line between their centers of mass (P2).

We also introduce a new layout feature. Given occlusion boundaries and figure/ground labels, we predict

Precomputation

Occlusion Boundaries [4]
Geometric Context (non-planar vertical surface) [28]
Hierarchical Segmentation - gives set of regions H Section (4.1)
Probability of BG region classifier [28]

Train Classifiers

Homogeneous Region Classifier

- Predicts if a region is likely to be all foreground or all background.
- Binary boosted decision tree classifier trained over regions $R_i \in H$ from hierarchy.
- Positive examples are either all foreground or all background, negatives cover both foreground and background.

Region Affinity Classifer

- Predicts if two regions are likely to lie on the same object.
- Binary boosted decision tree classifier trained over pairs of regions from hierarchy.
- Positive examples are pairs of regions covering the same object, negatives are foreground/background pairs or pairs from two different objects.

Layout Classifier

- Predicts if a region lies on the left, right, top, or bottom of an object.
- Binary logistic regression classifier trained over regions from hierarchy.
- HOG features extracted on 4x4 grid over left/right occlusion boundary maps.

Ranking Model

- Ranks a set of proposals $\mathcal{P}$ by likelihood of being an object.
- Optimize latent structured objective over proposed regions and appearance features from training set. (Eq.10)

Region Proposal

- Select seeds: $\mathcal{S} = \{r \in H \mid \text{area}(r) \geq 20 \times 20 \text{ pixels} \wedge \text{boundary strength}(r) \geq 0.005\}$
- For each image $\mathcal{I}$, seed $S \in \mathcal{S}_\mathcal{I}$ and parameters $(\gamma, \beta) \in \mathcal{G} \times \mathcal{B}$:
  - Compute superpixel affinity map: $f(l_i; S, \mathcal{I}, \gamma)$ (Eq. 3)
  - Propose region: $p = \arg\min_{\mathbf{l}} P(\mathbf{l}|\mathcal{I}, S, \gamma, \beta)$ (Eq. 1)
- Split regions with disconnected components and add to set
- Remove redundant regions with $\geq 90\%$ overlap

Region Ranking

- For each proposal $p \in \mathcal{P}$, compute appearance features $\mathbf{x}_p$
- For each image I find (approximate) highest scoring ranking with greedy inference: $\mathbf{r}_I = \arg\max_{\mathbf{r}} S(\mathbf{x}, \mathbf{r}; \mathbf{w})$ (Eq. 5)

Fig. 2. System Overview

Fig. 3. Example superpixel affinity maps for three sample seeds, indicated by green shaded region. Lighter shading indicates stronger affinity for belonging to the same object as the seed

TABLE 1
Features computed for pairs of regions for predicting the likelihood that the pair belongs to the same object. These features can capture non-local interactions between regions, producing better segmentations.

Feature Description	*Length*
P1. Color,Texture histogram intersection	2
P2. Sum,Max boundary strength between centers of mass	2
L1. Left+Right layout agreement	1
L2. Top+Bottom layout agreement	1
L3. Left+Right+Top+Bottom layout agreement	1

whether a particular region is on the left, right, top, bottom, or center of the object. These predictions are made by logistic regression classifiers based on histograms of occlusion boundary orientations, weighted by the predicted probabilities. Separate histograms are computed for figure and ground predictions. As a feature, we measure whether the layout predictions for two regions are consistent with them being on the same object. For example, if one region predicts that it is on the left of the object and a second region to the right of the first predicts that it is on the right side of the object, those regions are consistent. We construct a *layout* score for horizontal, vertical, and overall agreement (L1-L3).

Computing Superpixel Scores: Since the CRF is defined over superpixels, the region affinity probabilities are transferred to each superpixel i by averaging over the regions that contain it. The terms of this average are weighted by the probability that each region R is homogeneous ($P(H_R)$), which is predicted from the appearance features in Table 2:

$$P(l_i = 1|S,\mathcal{I}) = \frac{\sum_{\{R|i\in R\}} P(H_R)\cdot P(l_R = 1|S,\mathcal{I})}{\sum_{\{R|i\in R\}} P(H_R)}. \quad (2)$$

Note that we now have labels for superpixels (l_i) and for regions (l_R). We use $P(l_i|S,\mathcal{I})$ to compute the final affinity term $f(l_i;S,\mathcal{I},\gamma)$:

$$f(l_i;S,\mathcal{I},\gamma) = \begin{cases} 0 & : l_i = 1, i \in S \\ \infty & : l_i = 0, i \in S \\ \ln\left(\frac{P(l_i=1|\mathcal{I})}{P(l_i=0|\mathcal{I})}\right) + \gamma & : l_i = 1, i \notin S \end{cases} \quad (3)$$

The first two terms ensure that superpixels belonging to the seed are labeled foreground.

4.3.2 Edge Cost

The edge cost enforces a penalty for assigning different labels to adjacent superpixels when their separating boundary is weak. This boundary strength is computed from the occlusion boundary estimates for each pair of adjacent superpixels i, j: $P(B_{i,j}|\mathcal{I})$.

$$g(l_i,l_j;\mathcal{I}) = \begin{cases} 0 & : l_i = l_j \\ -\ln P(B_{i,j}|\mathcal{I}) & : l_i \neq l_j \end{cases} \quad (4)$$

This edge cost produces a submodular CRF, so exact inference can be computed quickly with a single graph-cut [29] for each seed and parameter combination. Proposals with disconnected components are split and the new components are added to the set, and highly overlapping ($\geq 90\%$) proposals are pruned. Further non-maximum suppression is handled in the ranking stage.

5 Ranking Proposals

We now introduce a ranker that attempts to order proposals, such that each object has a highly ranked proposal. This ranker encourages diversity in the proposals allowing us to achieve our goal of discovering *all* of the objects in the image. Below, we detail our objective function, which encourages top-ranked regions to correspond to different objects and more accurate object segmentations to be ranked higher. Then, we explain the image features that we use to rank the regions. Finally, we describe the structured learning method for training the ranker.

5.1 Formulation

By writing a scoring function $S(\mathbf{x},\mathbf{r};\mathbf{w})$ over the set of proposals $\mathbf{x}$ and their ranking $\mathbf{r}$, we cast the ranking problem as a joint inference problem, allowing us to take advantage of structured learning. The goal is to find the parameters $\mathbf{w}$ such that $S(\mathbf{x},\mathbf{r};\mathbf{w})$ gives higher scores to rankings that place proposals for all objects in high ranks.

$$S(\mathbf{x},\mathbf{r};\mathbf{w}) = \sum_i \alpha(r_i)\cdot\left(\mathbf{w}_a^T\mathbf{\Psi}(x_i) - \mathbf{w}_p^T\mathbf{\Phi}(r_i)\right) \quad (5)$$

The score is a combination of appearance features $\mathbf{\Psi}(x)$ and overlap penalty terms $\mathbf{\Phi}(r)$, where r indicates the rank of a proposal, ranging from 1 to the number of proposals M. This allows us to jointly learn the appearance model and the trade-off for overlapping regions. $\Phi(r)$ is the concatenation of two vectors $\Phi_1(r), \Phi_2(r)$: $\Phi_1(r)$ penalizes regions with high overlap with previously ranked proposals, and $\Phi_2(r)$ further suppresses proposals that overlap with *multiple* higher ranked regions. The second penalty is necessary to continue to enforce diversity after many proposals have at least one overlapping proposal. Since the strength of the penalty should depend on the amount of overlap (regions with 90% overlap should be suppressed more than regions with 50%) we want to learn overlap specific weights. To do this, we quantize the overlaps into bins of 10% and map the values to a 10 dimensional vector $\mathbf{q}(ov)$ with 1 for the bin it falls into and 0 for all other bins.

$$\Phi_1(r_i) = \mathbf{q}\left(\max_{\{j|r_j<r_i\}} ov(i,j)\right) \quad (6)$$

$$\Phi_2(r_i) = \sum_{\{j|r_j<r_i\}} \mathbf{q}\left(ov(i,j)\right) \quad (7)$$

The overlap score between two regions is computed as the area of their intersection divided by their union,

TABLE 2
Features used to describe the appearance of a proposal region. It is important that each of these features generalize across all object categories, including ones never seen during training.

Feature Description	*Length*
B1. Mean,max probability that exterior occludes	2
B2. Mean,max probability of exterior being occluded	2
B3. Mean,max probability of exterior boundary	2
B4. Mean,max probability of interior boundary	2
S1. Min,mean,max,max-min background probability	4
S2. Min,mean,max,max-min geometric context probabilities	16
S3. Color,texture background hist. intersection (local)	2
S4. Color,texture background hist. intersection (global)	2

with A_i indicating the set of pixels belonging to region i:

$$ov(i,j) = \frac{|A_i \cap A_j|}{|A_i \cup A_j|} \tag{8}$$

Each proposal's score is weighted by $\alpha(r)$, a monotonically decreasing function. Because higher ranked proposals are given more weight, they are encouraged to have higher scores. We found that the specific choice of $\alpha(r)$ is not particularly important, as long as it falls to zero for a moderate rank value. We use $\alpha(r) = \exp\left(\frac{(r-1)^2}{\sigma^2}\right)$, with $\sigma = 100$.

Computing $\max_{\mathbf{r}} S(\mathbf{x}, \mathbf{r}; \mathbf{w})$ cannot be solved exactly, so we use a greedy approximation that incrementally adds the proposal with the maximum marginal gain. We found that this works well for a test problem where full enumeration is feasible, especially when $ov(\cdot,\cdot)$ is sparse, which is true for this ranking problem.

5.2 Region Representation

The appearance features $\mathbf{\Psi}(x)$ characterize general properties for typical object regions, as summarized in Table 2. Since this is a category-independent ranker, we cannot rely on finely tuned category-dependent shape and appearance models. However, we can expect object boundaries to respect occlusion boundaries, so we encode the probability that the exterior is occluded by (B1) or occluding another region (B2), and the overall boundary strength (B3). We also encode the probability of interior boundaries (B4), which we expect to be small.

Additionally, certain "stuff-like" regions can be quickly identified as background, such as grass and sidewalks, so we learn a pixel based probability of background classifier on LabelMe [30], and characterize the response within the region (S1). This is learned using the region based classifiers from [28]. We also use the confidence of the vertical, porous, solid, and sky geometric classes using trained classifiers from [28], which is noted to often correspond to object and background classes (S2).

Finally, we encode the differences between color and texture distributions between the object and background. We compute the difference in histograms between the object and two regions: the local background region surrounding the object (S3) and the entire background (S4). The local background is defined by any superpixels that are at most two superpixels away from the proposed region.

5.3 Learning

To solve the structured learning problem, we use the margin-rescaled formulation of latent max-margin structured learning [31]. Here the objective is find a linear weighting w such that the highest scoring zero-loss ranking for each image scores higher than every incorrect ranking by a margin defined by the loss $\mathcal{L}$. Below the objective is written in unconstrained form:

$$\begin{aligned} \min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||^2 + C\sum_n \max_{\hat{\mathbf{r}} \in P^{(n)}} S\left(\mathbf{x}^{(n)}, \hat{r}; \mathbf{w}\right) + \mathcal{L}(O^{(n)}, \hat{\mathbf{r}}) \\ - C\sum_n \max_{\substack{\mathbf{r} \in P^{(n)}: \\ \mathcal{L}(O^{(n)}, \mathbf{r})=0}} S\left(\mathbf{x}^{(n)}, \mathbf{r}; \mathbf{w}\right) \\ \text{s.t. } \mathbf{w}_p \geq 0 \end{aligned} \tag{9}$$

Here, for image n, $O^{(n)}$ defines the set of ground truth regions for each image, $P^{(n)}$ is the set of valid labelings (the set of permutations over regions), $\mathbf{r}$ defines the highest scoring correct (zero-loss) ranking, and $\hat{\mathbf{r}}$ is the highest scoring incorrect ranking.

Loss: The loss $\mathcal{L}$ requires that each object o in the set of objects O should have high overlap with a highly ranked proposal. The loss has penalties for several levels of overlap τ, ranging from 50% to 100% in intervals of 5%. Since this loss is cumulative, i.e. a proposal with 100% overlap will contribute to the loss for every τ, it encourages the highest quality region for each object to have the highest rank:

$$\mathcal{L}(O, \hat{\mathbf{r}}) = \frac{1}{|O||T|} \sum_{\tau \in T} \sum_{o \in O} \min_{\{i | ov(i,o) \geq \tau\}} r_i - K_O. \tag{10}$$

The constant K_O is subtracted so that the lowest possible loss for a given ground truth is zero.

To learn this latent structured model, we iterate between finding the highest scoring zero-loss ranking for each image, and solving the structured learning problem with the fixed ground truth structure. To learn the structured subproblem we use a cutting-plane based optimization with alternates between finding the most violate constraint and updating $\mathbf{w}$ with the new constraints, and repeat until the change in $\mathbf{w}$ is small.

Initialization: Since the structured learning problem has latent variables (i.e. which zero loss ranking to use), the resulting objective function is non-convex and requires a strong initialization to perform well. To initialize, we first train a binary classifier over appearance features Ψ using a sampling of good regions ($\geq 65\%$ overlap) and bad regions ($\leq 33\%$ overlap). We then do a coordinate descent search for the weight of each bin of

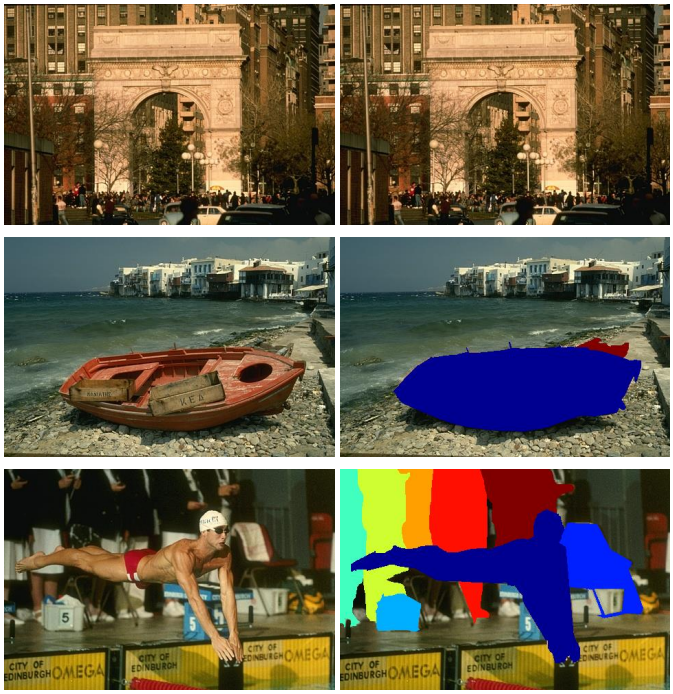


Fig. 4. Sample object annotations for BSDS. Each solid color corresponds to a distinct annotated object. All other pixels are considered background.

the penalty term that minimizes the loss. We do a single pass through the variables ordered in ascending bin size.

6 Experiments and Results

We perform experiments on the Berkeley Segmentation Dataset (BSDS) [5] and Pascal VOC 2011 [6]. All training and parameter selection is performed on the BSDS training set, unless otherwise noted, and results are evaluated on BSDS test and the Pascal validation set. Qualitative proposal results from both Pascal and BSDS are sampled in Figure 5.

Annotation: For both datasets, a ground truth segmentation is provided for each object. For BSDS, we label object regions by merging the original ground truth segments so that they correspond to objects. Object masks are non-overlapping subregions of the image that correspond to "things" with a definite shape, while "stuff"-type regions with indeterminate shape, such as sky and grass, are excluded. Regions such as buildings and trees are excluded since they are typically part of the background scene rather than distinct elements within it. There are an average of 2.6 annotated objects in each BSDS image. See Figure 4 for sample annotations. Note that since annotations are derived directly from the boundaries of BSDS, small objects without boundaries cannot be annotated, such as the cars in the street scene.

6.1 Baselines

We compare our method with two sets of baselines. First, we compare to the bottom-up **hierarchical segmentations** generated in Section 4.1. Second, we compare to the contemporary methods from [26] (Objectness) and [27] (CPMC). Since the Objectness method uses a bounding box representation, we repeat the comparison experiments using bounding box overlap on the larger VOC2011 Main val dataset.

TABLE 3
Comparison of features for generating proposals: affinity classification (AUC), recall @ 50% overlap, and best segment score (BSS).

	BSDS			Pascal 2011		
Feature	*AUC*	*Recall*	*BSS*	*AUC*	*Recall*	*BSS*
Color, Texture (P1)	75.0	77.0	65.7	71.5	74.9	63.8
C,T + Boundary (P1,P2)	79.8	80.2	66.3	**78.0**	75.7	64.5
C,T + Layout (P1,L1-L3)	77.5	**83.4**	**67.2**	72.6	**77.2**	**65.4**
All (P1,P2,L1,L2,L3)	**80.2**	79.7	66.2	77.2	76.2	64.9

6.2 Proposal Generation

To measure the quality of a set of proposals, we find the best segmentation overlap score for each object (BSS). From this, we can characterize the overall quality of segments with the mean BSS over objects, or compute the number of objects recalled with a BSS above some threshold. For our experiments, we set the threshold to 50% unless otherwise noted. A pixel-wise overlap threshold of 50% is typically, but not always, more stringent than a 50% bounding box overlap.

Features: The most commonly used features for segmentation are color and texture similarity, so we use this as a baseline. We then add the boundary crossing and layout features individually to see their impact. Finally, we combine all of the features to obtain our final model. To measure the performance of each feature, we consider the area under the ROC curve (AUC) for affinity classification, the best segment score, and recall at 50%. The results are shown in Table 3.

The first thing to note is that the addition of both the boundary and layout features are helpful for both datasets. In addition, we find that the affinity classification performance cannot fully predict a feature's impact on proposal performance. It is important to also consider how well the features facilitate producing a diverse set of proposals. Features that cause prediction to be more dependent on the seed region will produce a more diverse set of proposals. For the remainder of the experiments we use the color+layout features, since they create a more diverse set of proposals than color+boundaries+layout. The boundary cues are still captured with the pairwise term of the MRF.

Proposal Quality: We begin by considering similar baselines to [20]. The first baseline is to use each region from the hierarchical segmentation as an object proposal. The second baseline is to merge all pairs of adjacent regions, which achieves higher recall but with many more proposals. We can also measure the upper bound on performance by choosing the best set of superpixels for each object region.

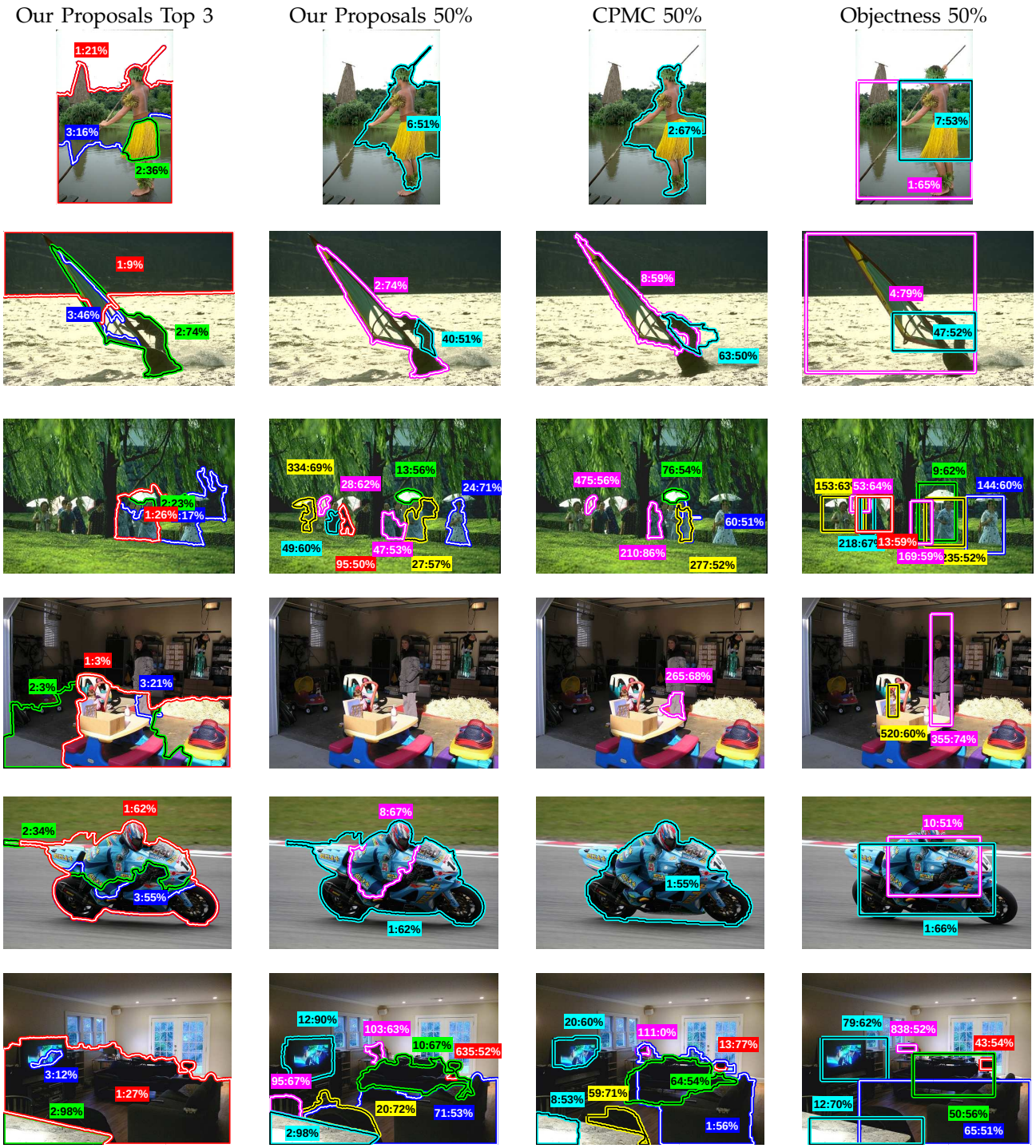
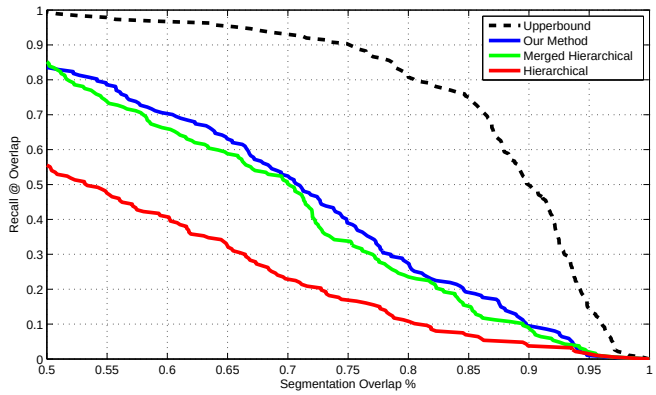
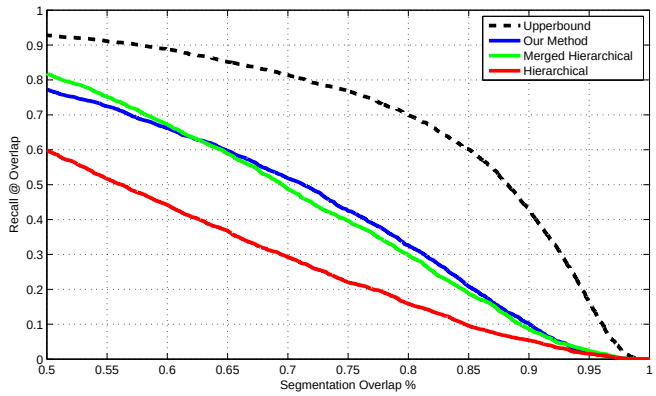


Fig. 5. Results from the proposal and ranking stages on BSDS (first 3 rows) and Pascal 2011 (last 3 rows). The left column shows the 3 highest ranked proposals, The remaining columns show the highest ranked proposals with at least 50% overlap with each object for Our Proposals, CPMC, and Objectness. Note that we use region overlap for ours and CPMC, and bounding box ovelap for Objectness. The number pairs displayed on each proposal correspond to rank and overlap, respectively. As seen in row 3, CPMC tends to have more trouble finding small objects in cluttered scenes. Objectness provides less detailed bounding boxes and generally requires more candidates to achieve the same level of recall.

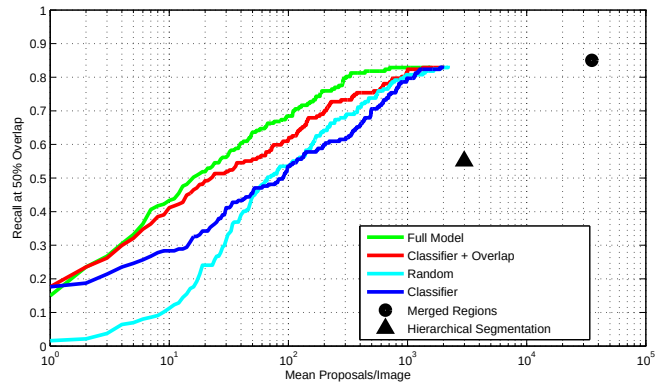


(a) Region: BSDS

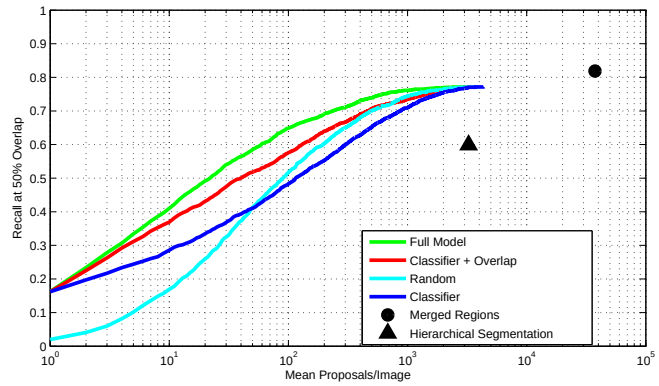


(b) Region: Pascal VOC2011 Segmentation val

Fig. 6. Recall vs. Region Overlap: The percentage of objects recalled as a function of best overlap with ground truth. For BSDS, we generate better proposals for all levels of overlap. For Pascal, we outperform the baselines for higher recall levels and are still comparable at 50% overlap. Note that we use 20-30 times fewer regions than the baselines.



(a) Region: BSDS



(b) Region: Pascal VOC 2011 Segmentation val

Fig. 7. Recall vs. number of proposals per image: When considering recall for more than 50 proposals per image, enforcing diversity (Random) is a more important than object appearance (Classifier). Combining diversity and appearance (Classifier + Overlap) improves performance further, and jointly learning both (Full model) gives even further gains.

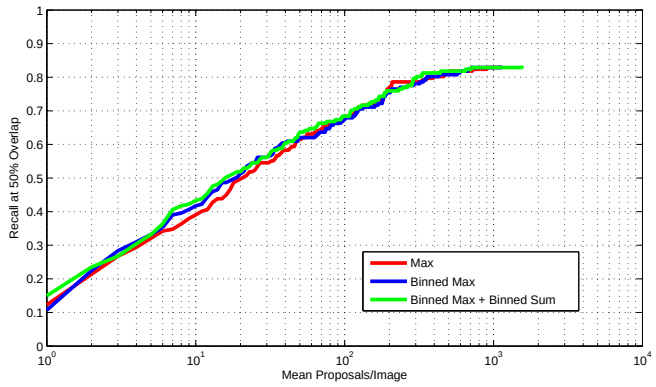
It is clear from Figure 6 that the initial hierarchical segmentation is not well suited for proposing object candidates. After merging proposals, the segmentation quality is comparable to our method, but as Figure 7 shows, it produces more than an order of magnitude more proposals. For both datasets, our method produces more high quality proposals for overlaps greater than 65%.

6.3 Ranking Performance

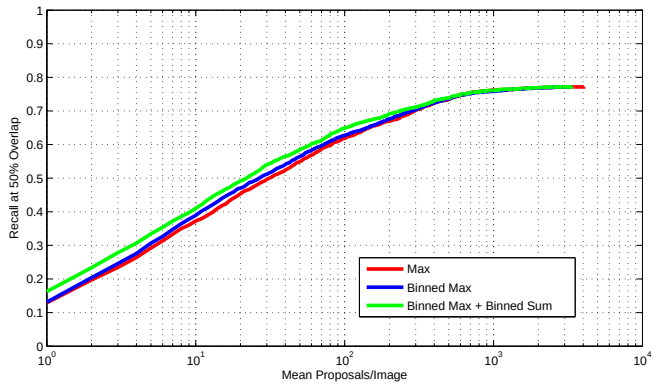
We compare our ranking method to three baselines. The first method scores each proposal independently, and the ranking is produced by sorting these scores from high to low. Positive examples are chosen from a pool proposals with at least 50% overlap with some object and negative examples have no more than 35% overlap with any object. The second baseline includes the overlap penalty of our method, but learns the appearance model and trade-off terms separately, as in [27]. The final baseline simply assigns random ranks to each proposal. This can be seen as encouraging diversity without taking into account appearance. To evaluate the quality of our ranker, we measure the number of objects recalled when we threshold each image's bag at a certain size. The results are presented in Figure 7.

We find that by jointly learning the appearance and suppression models, our method outperforms each of the baselines. Because the independent classifier does not encourage diversity, only the first object or object-like region is given a high rank, and the number of proposals required to recall the remaining objects can be quite high. In fact, when considering more than 50 proposals, the random ranker quickly outperforms the independent classifier. This emphasizes the importance of encouraging diversity. However, both models that include both appearance models and overlap terms outperform the random ranker. Finally, by learning with an appropriate loss and jointly learning all of the parameters of the model with structured learning, we achieve small but noticeable gains over the baseline with an overlap term.

In Figure 8 we isolate the influence of each of the

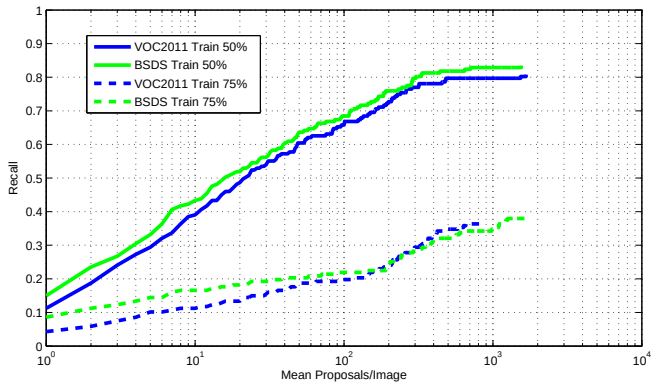


(a) Region: BSDS

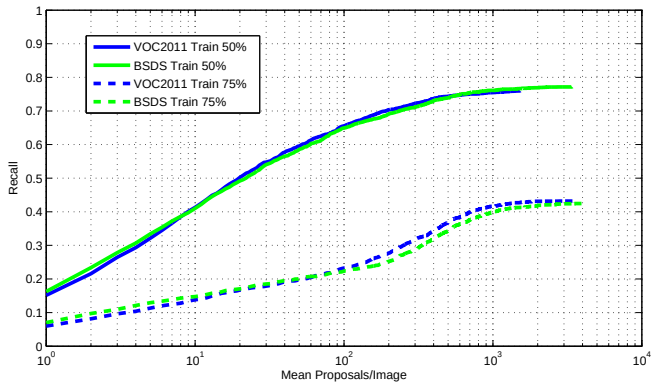


(b) Region: Pascal VOC 2011 Segmentation val

Fig. 8. Effects of Ranker Scoring Function: The baseline (Max) only uses the maximum overlap with a higher ranked proposal. Results are improved incrementally by both binning the overlaps (Binned Max) and adding the sum of higher ranked overlaps (Binned Max + Binned Sum). The latter is used in the final system.



(a) Region: BSDS



(b) Region: Pascal VOC 2011 Segmentation val

Fig. 10. Cross-Dataset comparison: Our method is trained on VOC2011 (blue) and trained on BSDS (green). Note that recall at both 50% and 75% overlap is quite comparable at all ranks. There is only a small advantage when training on the same set as testing, showing that our method generalizes well and need not be retrained for every new dataset.

components of the ranker's scoring function. First, we consider the most basic overlap penalty term as used in [27], consisting of the maximum overlap with a higher ranked proposal (**Max**, i.e. the unbinned version of Φ_1 in eq. 6). Next, we binning the output of the max function, withouth the sum (**Binned Max**). Although it is difficult to discern the benefit on BSDS, there is a clear improvment on Pascal. Finally, by adding the sum of the binned overlaps (**Binned Max + Binned Sum**), which is representative of the final ranking procedure used throughout the paper, we get further improvements on Pascal. Note that each method is trained using the full structured learning process.

Finally, we provide a breakdown of recall for individual categories of the Pascal VOC 2011 dataset in Figure 9. These results are especially promising, because many of the categories with high recall, such as dog and cat, are difficult for standard detectors to locate. The low performance for categories like car and sheep is mainly due to the difficulty of proposing small regions ($< 0.5\%$ of the image area, or < 1000 pixel area), especially when the objects are in crowded scenes. The dependence of recall on area is shown in Figure 13. The highly detailed ground truth pixel masks for bicycles makes them extremely difficult to recall for our method.

6.4 Cross-Dataset Comparison

To explore our method's ability to generalize to new datasets, we compare the overall proposal performance when trained and tested on the same set and across datasets. In Figure 10, we find that training on BSDS and testing on BSDS gives a slight gain over training on Pascal. The greater diversity of objects in BSDS may explain this advantage. In contrast, there is no significant difference between training on BSDS or Pascal when testing on Pascal. This result suggests that diversity of training examples, rather than quantity, is more important for our method to generalize. It also confirms that our method generalizes well and does not need to be retrained for each new dataset.

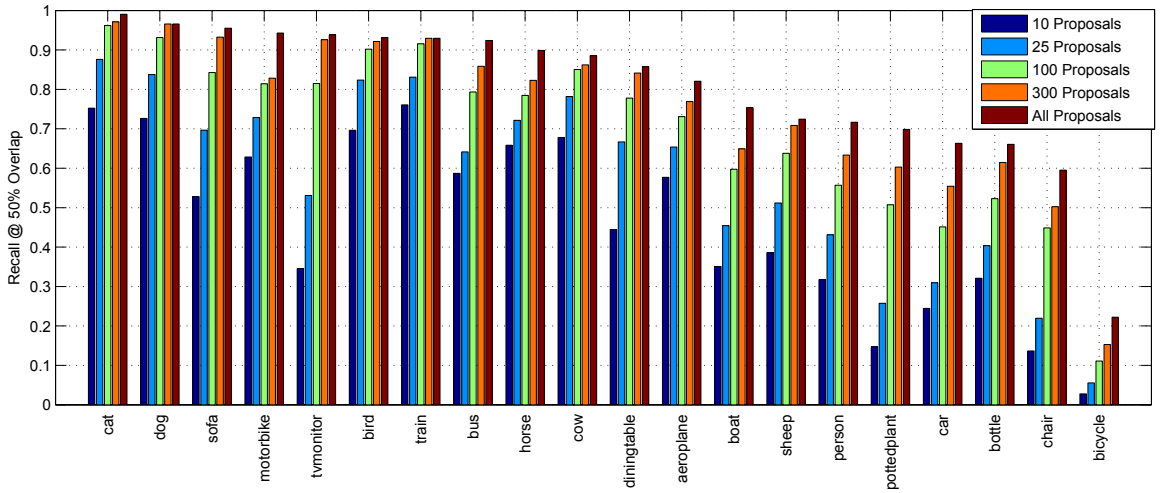


Fig. 9. Recall for each object category in Pascal with region overlap. These results are quite promising because many of the categories with high recall are difficult for standard object detectors to recognize. For many categories, most of the instances can be discovered in the first 100 proposals.

6.5 Comparison to Objectness,CPMC

Next, we compare to the Objectness [26] and CPMC [27] methods.

Proposal Quality: Figure 11 compares the recall at different overlap thresholds. With the region overlap criteria, CPMC recalls more objects at higher overlaps, especially when considering fewer proposals for each image. Their pixelwise segmentation is able to give more detailed segmentations. However, their proposals have less diversity which limits recall when using lower region overlap thresholds or the bounding box overlap criteria. The Objectness method gives comparable levels of recall at 50% bounding box overlap for both datasets, but their aggressive non-maximum suppression procedure causes recall to quickly drop for higher overlap thresholds.

Ranking: Figure 12 compares the quality of the ranking by showing the recall for different numbers of proposals for each image. At 50% region overlap, our method slightly outperforms CPMC at all ranks. However, at 75% overlap, their higher quality per-pixel masks have higher recall for more than 30 proposals per image. With bounding box overlap, our method and CPMC perform comparably on BSDS, and our method has 5% higher recall for most ranks on Pascal.

The Objectness ranking has a lower recall than both methods for less than a few hundred proposals per image. It performs comparably to our method at 500 proposals per image for BSDS and 2000 proposals for Pascal.

Area: Figure 13 show the dependence of each method on region or bounding-box area as a fraction of image pixels. All of the methods excel with $90\% - 100\%$ recall for regions which cover greater than 5% of the image. However, both CPMC and Objectness appear to be more sensitive to smaller objects. Our superpixel based representation appears to give a good balance between giving detailed segmentations while reducing the search space for candidate objects.

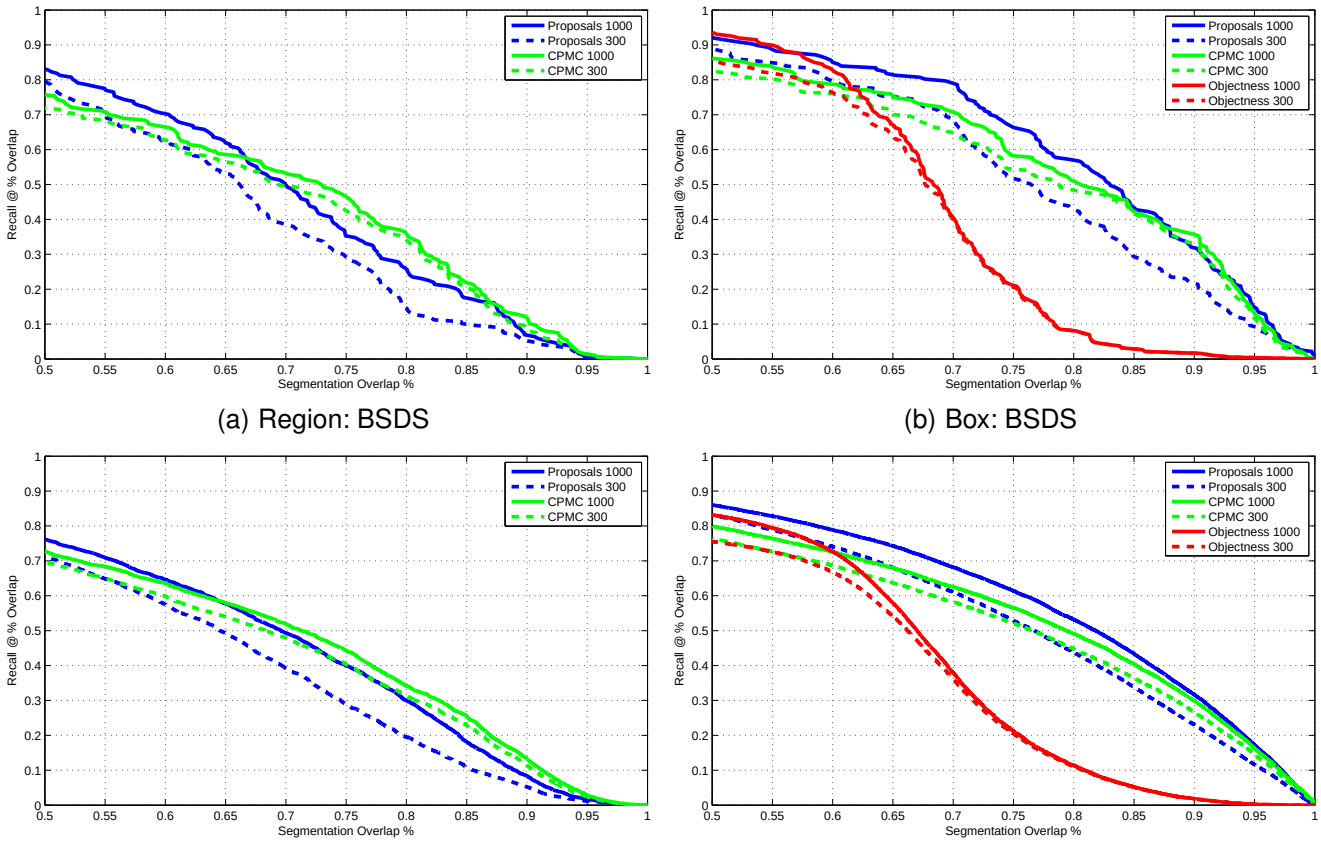
7 Conclusion

We have introduced a procedure that generates a small, but diverse set of category-independent object proposals. By incorporating the affinity predictions, we can direct the search for segmentations to produce good candidate regions with far fewer proposals than standard segmentations. Our ranking can further reduce the number of proposals, while still maintaining high diversity. Our experiments show that this procedure generalizes well and can be applied for many categories.

The results on Pascal are especially encouraging, because with as few as 100 proposals per image, we can obtain high recall for many categories that standard scanning window detectors find difficult. This is quite amazing, considering that the system had never seen most of the Pascal categories during training!

Beyond categorization, our proposal mechanism can be incorporated in applications where category models are not available. When presented with images of new objects, our proposals can be used in an active learning framework to learn about unfamiliar objects. Alternatively, they can be used for automatic object discovery methods such as [21]. Combined with the description based recognition methods [1], [2], we could locate and describe new objects.

While this method performs well in general, it has difficulty in cases where the occlusion boundary predictions fail and for small objects. These are cases where having some domain knowledge, such as appearance or shape models can complement a generic proposal mechanism. This suggests a joint approach in which bottom-up region proposals are complemented by part



(a) Region: BSDS

(b) Box: BSDS

(c) Region: Pascal VOC 2011 Segmentation val

(d) Box: Pascal VOC 2011 Main val

Fig. 11. Recall at different overlap thresholds for our method, Objectness, and CPMC. Solid lines indicate recall with 1000 regions per image, dashed lines for 300 regions.

or category detectors that incorporate domain knowledge.

REFERENCES

[1] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, "Describing objects by their attributes," in *CVPR*, 2009.

[2] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *CVPR*, 2009.

[3] M. A. Goodale, A. D. Milner, L. S. Jakobson, and D. P. Carey., "A neurological dissociation between perceiving objects and grasping them," *Nature*, vol. 349, pp. 154–156, Jan 2000.

[4] D. Hoiem, A. A. Efros, and M. Hebert, "Recovering occlusion boundaries from an image," *Int. J. Comput. Vision*, vol. 91, no. 3, pp. 328–346, Feb. 2011. [Online]. Available: http://dx.doi.org/10.1007/s11263-010-0400-4

[5] D. Martin, C. Fowlkes, and J. Malik, "Learning to find brightness and texture boundaries in natural images," *NIPS*, 2002.

[6] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2011 (VOC2011) Results," http://www.pascal-network.org/challenges/VOC/voc2011/workshop/index.html.

[7] ——, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[8] P. Viola and M. J. Jones, "Robust real-time face detection," *IJCV*, vol. 57, no. 2, 2004.

[9] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *PAMI*, 2010.

[10] B. Leibe, A. Leonardis, and B. Schiele, "Robust object detection with interleaved categorization and segmentation," *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 259–289, May 2008.

[11] S. Maji and J. Malik, "Object detection using a max-margin hough transform," *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, vol. 0, pp. 1038–1045, 2009.

[12] O. Chum and A. Zisserman, "An exemplar model for learning object classes," in *CVPR*, 2007.

[13] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman, "Multiple kernels for object detection," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2009.

[14] C. Gu, J. Lim, P. Arbelaez, and J. Malik, "Recognition using regions," *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, pp. 1030–1037, 2009.

[15] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. PAMI*, vol. 22, no. 8, August 2000.

[16] P. Felzenszwalb and D. Huttenlocher, "Efficient graph-based image segmentation," *IJCV*, vol. 59, no. 2, 2004.

[17] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 5, pp. 898–916, 2011.

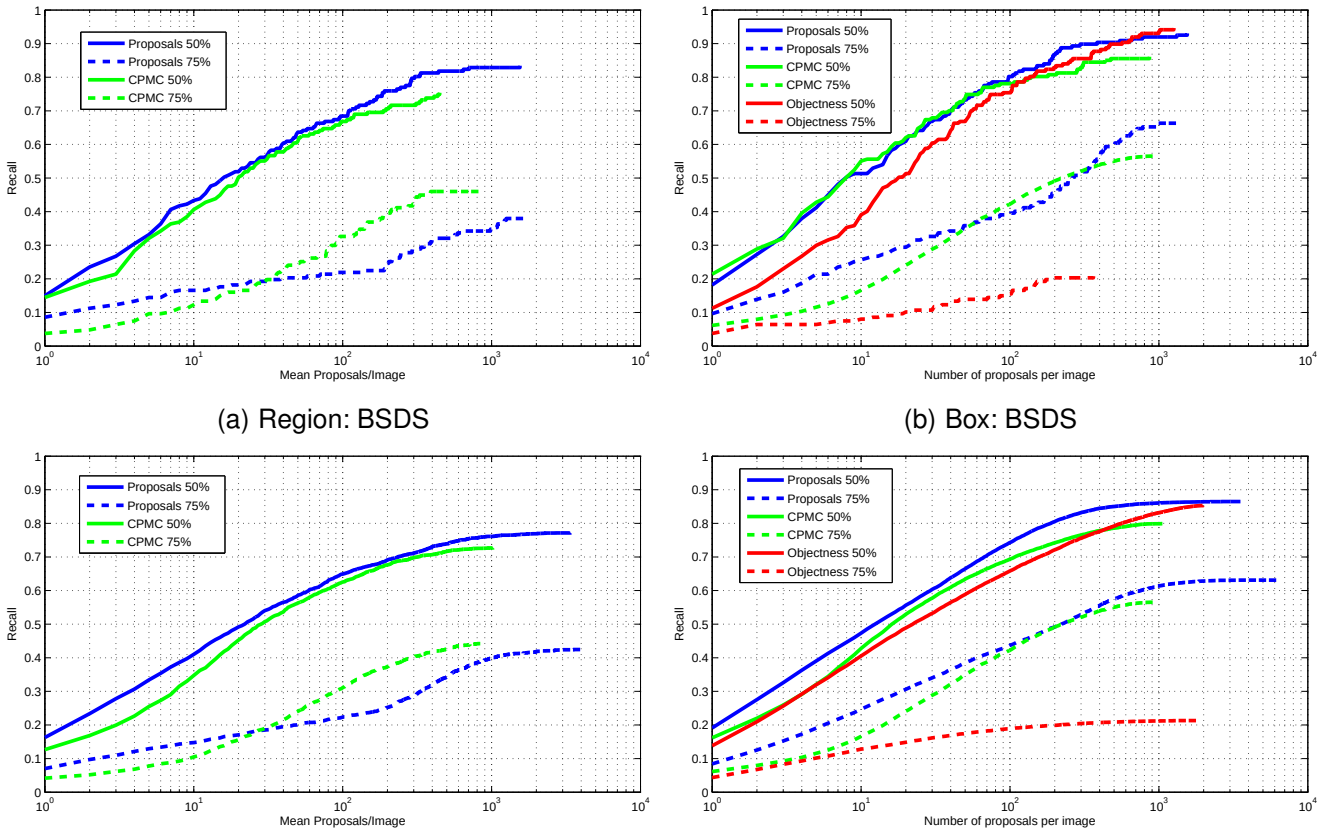
[18] E. Sharon, M. Galun, D. Sharon, R. Basri, and A. Brandt., "Hierarchy and adaptivity in segmenting visual cues," *Nature*, June 2006.

[19] D. Hoiem, A. A. Efros, and M. Hebert, "Geometric context from a single image," in *ICCV*, 2005.

[20] T. Malisiewicz and A. A. Efros, "Improving spatial support for objects via multiple segmentations," in *BMVC*, 2007.

[21] B. C. Russell, A. A. Efros, J. Sivic, W. T. Freeman, and A. Zisserman, "Using multiple segmentations to discover objects and their extent in image collections," in *CVPR*, 2006.

[22] A. Stein, T. Stepleton, and M. Hebert, "Towards unsupervised whole-object segmentation: Combining automated matting with boundary detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2008.



(a) Region: BSDS

(b) Box: BSDS

(c) Region: Pascal VOC 2011 Segmentation val

(d) Box: Pascal VOC 2011 Main val

Fig. 12. Recall at different numbers of proposals per image for our method, Objectness, and CPMC. Solid lines indicate 50% overlap, dashed lines for 75% overlap.

[23] S. Gould, R. Fulton, and D. Koller, "Decomposing a scene into geometric and semantically consistent regions," in *ICCV*, 2009.

[24] D. Walther and C. Koch, "2006 special issue: Modeling attention to salient proto-objects," *Neural Networks*, vol. 19, no. 9, pp. 1395–1407, 2006.

[25] T. Liu, J. Sun, N. ning Zheng, X. Tang, and H. yeung Shum, "Learning to detect a salient object," in *in: Proceedings of IEEE Computer Society Conference on Computer and Vision Pattern Recognition (CVPR*. CVPR, 2007, pp. 1–8.

[26] B. Alexe, T. Deselaers, and V. Ferrari, "Measuring the objectness of image windows," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 11, pp. 2189–2202, 2012.

[27] J. Carreira and C. Sminchisescu, "Cpmc: Automatic object segmentation using constrained parametric min-cuts," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 7, pp. 1312–1328, 2012.

[28] D. Hoiem, A. A. Efros, and M. Hebert, "Recovering surface layout from an image," *IJCV*, vol. 75, no. 1, pp. 151–172, 2007.

[29] C. Rother, V. Kolmogorov, and A. Blake, ""grabcut": interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 309–314, 2004.

[30] B. C. Russell, A. Torralba, K. Murphy, and W. T. Freeman, "Labelme: A database and web-based tool for image annotation," *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 157–173, 2008.

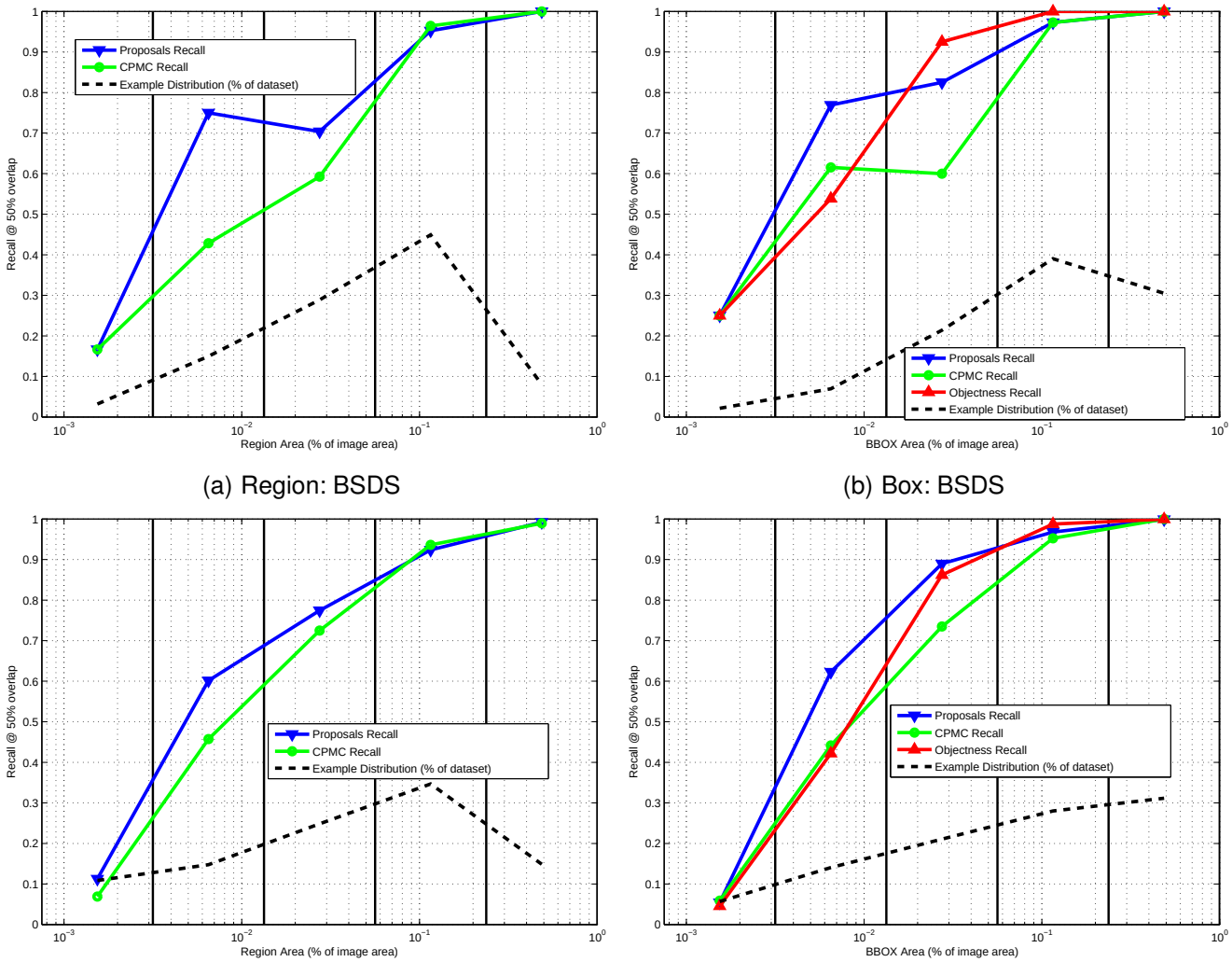
[31] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, "Large margin methods for structured and interdependent output variables," *Journal of Machine Learning Research*, vol. 6, pp. 1453–1484, 2005.

PLACE PHOTO HERE

Ian Endres received the BS degree in computer science from University of Illinois at Urbana-Champaign in 2008. He is currently pursuing the Ph.D. degree in at University of Illinois at Urbana-Champaign. He is the recipient of the Richard T. Cheng Fellowship.

PLACE PHOTO HERE

Derek Hoiem is an assistant professor in Computer Science at the University of Illinois at Urbana-Champaign. He received his PhD in Robotics from Carnegie Mellon University in 2007. Dereks research in visual scene understanding and object recognition has been recognized with an ACM Doctoral Dissertation Award honorable mention, CVPR best paper award, NSF CAREER grant, Intel Early Career Faculty award, and Sloan Fellowship.



(a) Region: BSDS

(b) Box: BSDS

(c) Region: Pascal VOC 2011 Segmentation val

(d) Box: Pascal VOC 2011 Main val

Fig. 13. Recall vs. object size: The plot shows the percentage of recalled objects based on their area, relative to the image size. Histogram bin edges are indicated by solid vertical lines. This demonstrates that uncovering smaller objects is more difficult than larger objects, but for each dataset, more than 60% of objects between 0.3% and 1.1% of the image are still recovered. This is due to weaker object cues and because the region overlap criteria is more sensitive to individual pixel errors for smaller objects. The dashed lines also show the proportions of the dataset for each object size.