Category Independent Object Proposals

Ian Endres and Derek Hoiem

Department of Computer Science
University of Illinois at Urbana-Champaign
{iendres2,dhoiem}@uiuc.edu

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 2008 demonstrate our ability to find most objects within a small bag of proposed regions.

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 2008 segmentation dataset [6]. 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

Here, we relate our work to category-dependent and category-independent methods for proposing object regions.

Category Dependent Models: By far, the most common approach to object localization is to evaluate a large number of windows (e.g., [7, 8]), which are found by searching naively over position and scale or by voting from learned codewords [9, 10], distinctive keypoints [11, 12], or regions [13]. 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 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., [14, 15] are not yet reliable. For this reason, many have proposed creating hierarchical segmentations (e.g., [16, 4, 17]) or multiple overlapping segmentations (e.g., [18–21]). Even these tend not to reliably produce good object regions, so Malisiewicz et al. [19] 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.

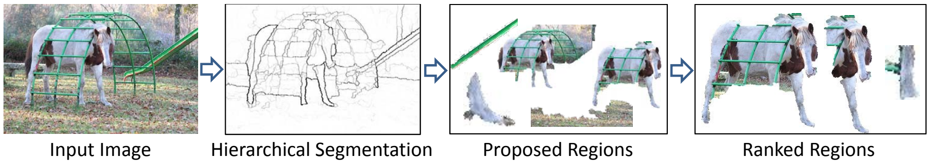

Fig. 1: Our pipeline: compute a hierarchical segmentation, generate proposals, and rank proposed regions. At each stage, we train classifiers to focus on likely object regions and encourage diversity among the proposals, enabling the system to localize many types of objects. See section 3 for a more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

An alternative approach is to attempt to segment pixels of foreground objects [22] or salient regions [23, 24]. 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 later stages of classification. First, Alexe et al. [25] consider an "objectness" measure over bounding boxes, which they use to bias a sampling procedure for potential object bounding boxes. However, they are limited to the restricted expressiveness of a bounding box. Alternatively, Carreira and Sminchisescu [26] consider a similar region proposal and ranking pipeline to ours. Segmentations are performed using graph cuts and simple color cues, and the regions are ranked through classification based on gestalt cues with a simple diversity model. Our approach 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 these 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.

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 details of this algorithm are not relevant to our paper). The occlusion algorithm outputs four successively coarser segmentations, with a probability of occlusion and of the figure/ground label 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 we average over the segmentations. Then, we create our hierarchical segmentation with agglomerative grouping based on boundary strength, as in [16], 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. Also, we remove regions with boundaries weaker than 0.01 , 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

CRF Segmentation: 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:

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

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 five γ values from between $[-2, 2]$, and five β values from $[0, 5]$.

Affinity: To compute the superpixel affinity $f(l_i; S, X, \gamma)$, we first compute each region R's affinity for lying on the same object as the seed S. We learn the foreground probability $P(l_R|S, X)$ 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.

The classifier uses features for cohesion, boundary, and layout cues, 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 ($P2$).

Feature Description	*Length*
P1. Color,Texture histogram intersection	2
P2. Sum,Max strength of boundary crossed 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

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.

We introduce a new layout feature. Given occlusion boundaries and figure/ground labels, we predict whether a particular region is on the left, right, top, bottom, or center of the object. These predictions are made by boosted decision tree classifiers based on histograms of occlusion boundaries, where the boundaries are separated based on figure/ground labels. 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 ($L1 - L3$).

Since the CRF is defined over superpixels, the region affinity probabilities are transfered 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:

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

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

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

The infinite cost ensures that superpixels belonging to the seed are labeled foreground.

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: $P(B_{i,j}|X)$.

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

This edge cost produces a submodular CRF, so exact inference can be computed quickly with a single graph-cut [27] for each seed and parameter combination. Proposals with disconnected components are split, and highly overlapping ($\geq$ 97%) 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.

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 can 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.

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

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. $\varPhi_1(r)$ penalizes regions with high overlap with previously ranked proposals, and $\varPhi_2(r)$ 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:

$$\varPhi_1(r_i) = \max_{\{j|r_j<r_i\}} ov(i,j) \tag{6}$$

$$\varPhi_2(r_i) = \sum_{\{j|r_j<r_i\}} ov(i,j) \tag{7}$$

The overlap score is computed as the area of two regions' intersection divided by their union, 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 $\sigma = 150$.

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.

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 or occluding another region. We also encode the probability of interior boundaries, 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 [28], and characterize the response within the region. We also use the confidence of the vertical solid non-planar geometric class, using trained classifiers from [29], which is noted to often correspond to object classes. 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 and the entire background.

Learning: To solve the structured learning problem, we use the slack-rescaled method with loss penalty used in [30]. This method finds the highest scoring labeling, rather than the most violated constraint, and adds an additional cost to

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 of exterior boundary	2
B2. Mean,max probability of interior boundary	2
B3. Mean,max probability that exterior occludes	2
B4. Mean,max probability of exterior being occluded	2
S1. Min,mean,max,max-min probability of background	4
S2. Min,mean,max,max-min probability of vertical surface	4
S3. Color,texture histogram intersection with local background	2
S4. Color,texture histogram intersection with global background	2

the objective to penalize for high loss candidates:

$$\min_{\mathbf{w},\xi_n} \frac{1}{2}||\mathbf{w}||^2 + \frac{C_1}{N}\sum_n \xi_n + \frac{C_2}{N}\sum_n \mathcal{L}(\mathbf{r}^{(n)}, \hat{\mathbf{r}}^{(n)}) \tag{9}$$

$$\begin{aligned} &\text{s.t. } \forall \mathbf{r} \in P^{(n)} \backslash \mathbf{r}^{(n)}, \forall n \\ S(\mathbf{x}^{(n)}, \mathbf{r}^{(n)}; \mathbf{w}) - S(\mathbf{x}^{(n)}, \mathbf{r}; \mathbf{w}) &\geq 1 - \tfrac{\xi_n}{\mathcal{L}(\mathbf{r}^{(n)}, \mathbf{r})}, \\ \xi_n &\geq 0 \\ \mathbf{w}_p &\geq 0 \end{aligned}$$

where, for image n, $\mathbf{r}^{(n)}$ is the ground truth ranking, $\hat{\mathbf{r}}^{(n)} = \mathrm{argmax}_{\mathbf{r} \in P^{(n)}} S(\mathbf{x}^{(n)}, \mathbf{r}; \mathbf{w})$ is the highest scoring proposal, and $P^{(n)}$ is the set of valid labellings, in this case, the set of permutations over regions. The cutting plane approach avoids having to exhaustively enumerate the resulting intractable set of constraints.

The loss $\mathcal{L}$ must enforce two properties: higher quality proposals should have higher ranks ($\mathcal{L}_1$), and each object o in the set of objects O should have a highly ranked proposal ($\mathcal{L}_2$):

$$\begin{aligned} \mathcal{L}(\mathbf{r}, \hat{\mathbf{r}}) &= \tfrac{1}{2}\mathcal{L}_1(\mathbf{r}, \hat{\mathbf{r}}) + \tfrac{1}{2}\mathcal{L}_2(\mathbf{r}, \hat{\mathbf{r}}) \\ \mathcal{L}_1(\mathbf{r}, \hat{\mathbf{r}}) &= \tfrac{1}{|O|} \textstyle\sum_{o \in O} \sum_{\{(i,j) | r_i < r_j\}} I[ov(i,o) < ov(j,o)] \\ \mathcal{L}_2(\mathbf{r}, \hat{\mathbf{r}}) &= \tfrac{1}{|O|} \textstyle\sum_{o \in O} \min_{\{i | ov(i,o) \geq 50\%\}} r_i \end{aligned} \tag{10}$$

To learn this structured model, we iteratively find the highest scoring ranking for an image, update $\mathbf{w}$ with this new constraint, and repeat until the change in $\mathbf{w}$ is small.

6 Experiments and Results

We perform experiments on the Berkeley Segmentation Dataset (BSDS) [5] and the Segmentation Taster images from PASCAL VOC 2008 [6]. All training and parameter selection is performed on the BSDS training set, and results are evaluated on BSDS test and the PASCAL validation set. 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.

Qualitative results from both PASCAL and BSDS are sampled in Figure 2.

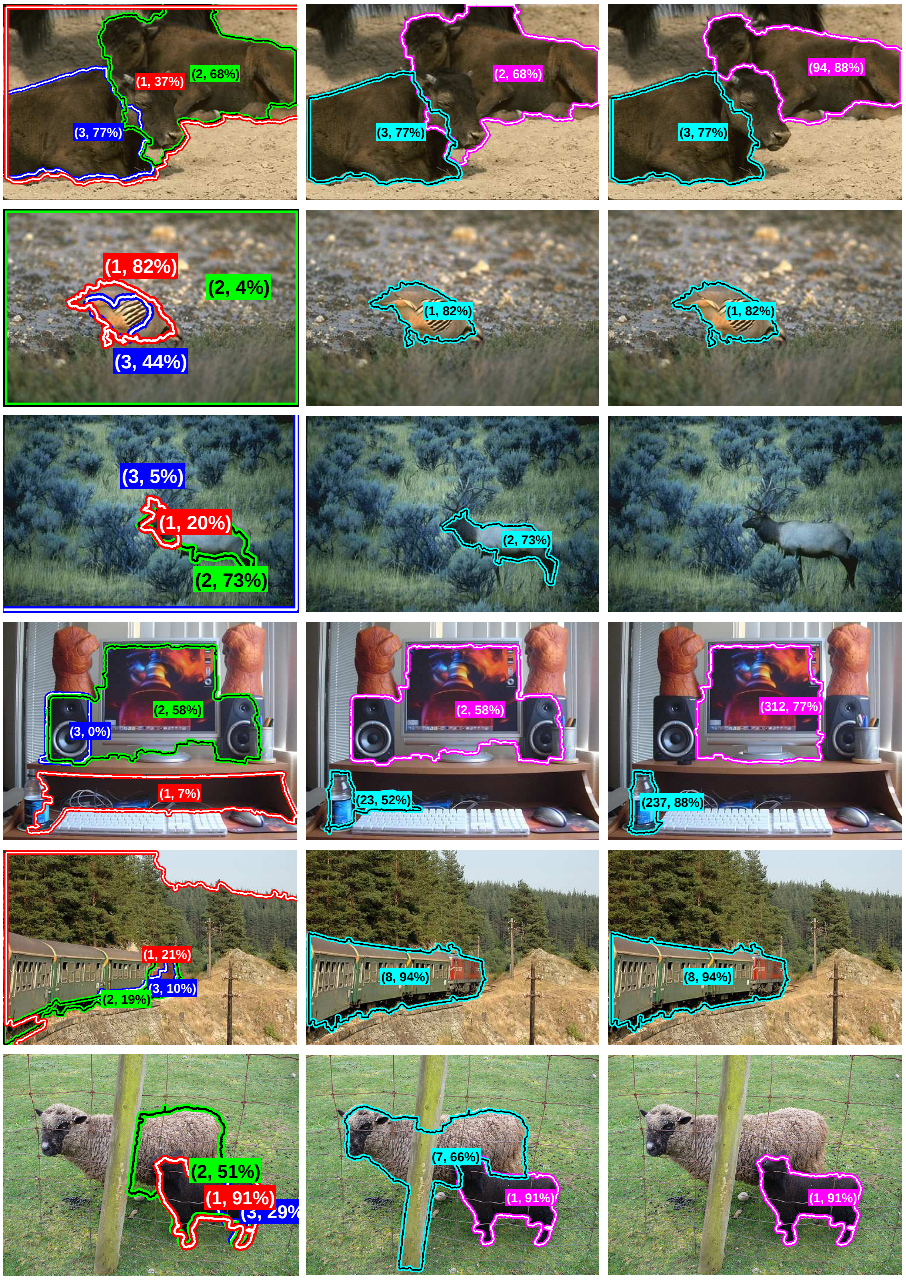

Fig. 2: Results from the proposal and ranking stages on BSDS (first 3 rows) and PASCAL 2008 (last 3 rows). The left column shows the 3 highest ranked proposals, The center column shows the highest ranked proposal with 50% overlap for each object. The right column shows the same for a 75% threshold. The number pairs displayed on each proposal correspond to rank and overlap, respectively. The desk scene demonstrates the diversity of our ranking. The train and deer demonstrate the high quality of proposals.

6.1 Proposal Generation

To measure the quality of a bag 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 recall by thresholding the BSS at some value, and counting the number of objects with a BSS of at least this threshold. For our experiments, we set the threshold to 50% unless otherwise noted. A pixel-wise overlap threshold of 50% is usually, 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.

	BSDS			PASCAL		
Feature	*AUC*	*Recall*	*BSS*	*AUC*	*Recall*	*BSS*
Color,Texture (P1)	0.72	75.4 %	0.655	0.68	78.8%	0.67
C,T + Boundary Crossing (P1,P2)	0.77	81.8%	0.671	0.76	79.7%	**0.68**
C,T + Layout (P1,L1,L2,L3)	0.74	82.9%	0.679	0.71	**81.1%**	**0.68**
All (P1,P2,L1,L2,L3)	**0.83**	**84.0%**	**0.69**	**0.80**	79.7%	**0.68**

Table 3: A comparison of how features impact affinity classification (AUC), recall @ 50% overlap, and best segment score (BSS). Both classification accuracy and diversity of proposals must be considered when choosing a set of features.

Proposal Quality: We define similar baselines to [19]. 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.

It is clear from Figure 3 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 6 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%.

Finally, we provide a breakdown of recall for individual categories of the PASCAL dataset in Figure 4. 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 5.

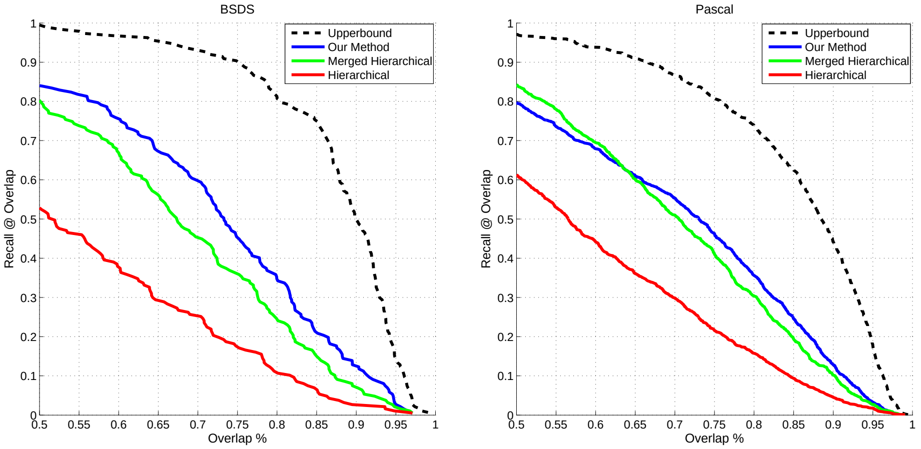

Fig. 3: These curves characterize the quality of proposals from each method, showing the percentage of objects recalled for a given overlap %. For BSDS, we generate better proposals for all levels of overlap. For PASCAL, we outperform the baselines for higher recall levels and are still comparable at 50% overlap. These results are impressive because we consider 20-30 times fewer regions.

6.2 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, as in [26]. 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. 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 6.

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 10 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 the model, we achieve small but noticeable gains over the baseline with an overlap term.

7 Discussion

We have introduced a procedure that generates a small, but diverse set of category-independent object proposals. By incorporating the affinity predictions,

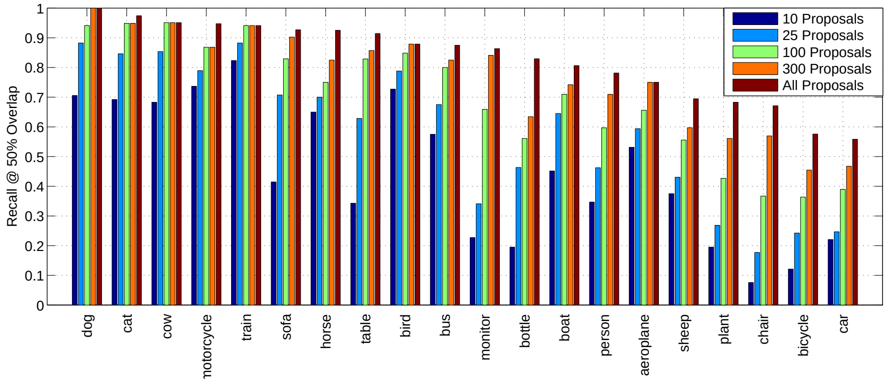

Fig. 4: Recall for each object category in PASCAL. These results are quite promising because many of the categories with high recall are difficult for standard object detectors to recognize. For many categories, most of the instances can be discovered in the first 100 proposals.

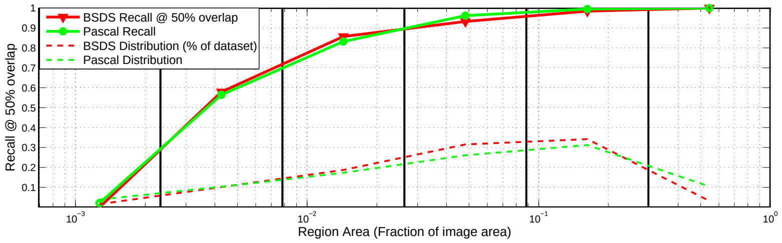

Fig. 5: Recall vs. object size: The plot shows the percentage of recalled objects based on their area, relative to the image size. Histogram bin edges are indicated by solid vertical lines. This demonstrates that uncovering smaller objects is more difficult than larger objects, but nearly 60% of objects between 0.3% and 0.8% 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.

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 [20]. Combined with the description based recognition methods [1, 2], we could locate and describe new objects.

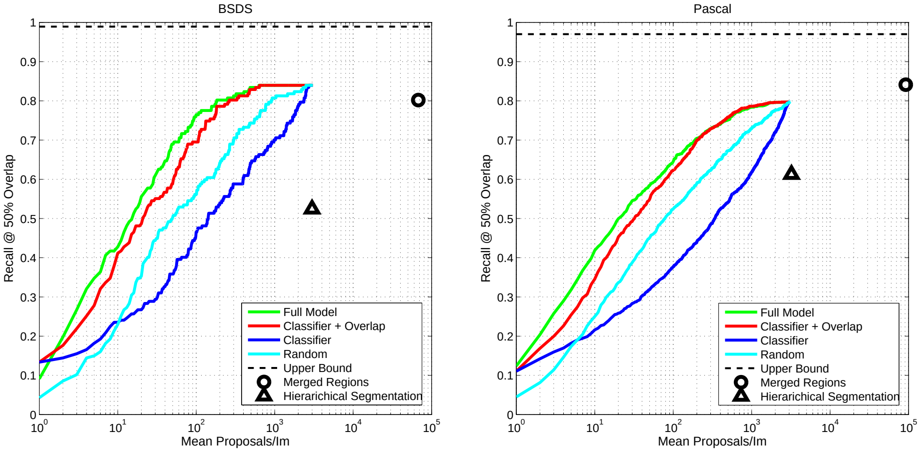

Fig. 6: Recall vs. number of proposals per image: When considering recall for more than 10 proposals per image, enforcing diversity (Random) is a more important than object appearance (Classifier). Combining diversity and appearance (Classifier + Overlap) improves performance further, and jointly learning both (Full model) gives further gains.

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 or category detectors that incorporate domain knowledge.

8 Acknowledgments

This work was supported in part by the National Science Foundation under IIS-0904209.

References

1. Farhadi, A., Endres, I., Hoiem, D., Forsyth, D.: Describing objects by their attributes. In: CVPR. (2009)
2. Lampert, C.H., Nickisch, H., Harmeling, S.: Learning to detect unseen object classes by between-class attribute transfer. In: CVPR. (2009)
3. Goodale, M.A., Milner, A.D., Jakobson, L.S., Carey., D.P.: A neurological dissociation between perceiving objects and grasping them. Nature **349** (2000) 154–156
4. Hoiem, D., Stein, A.N., Efros, A.A., Hebert, M.: Recovering occlusion boundaries from an image. In: ICCV. (2007)
5. Martin, D., Fowlkes, C., Malik, J.: Learning to find brightness and texture boundaries in natural images. NIPS (2002)
6. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results.

http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html (2008)
7. Viola, P., Jones, M.J.: Robust real-time face detection. IJCV **57** (2004)
8. Felzenszwalb, P., McAllester, D., Ramanan, D.: A discriminatively trained, multiscale, deformable part model. In: CVPR. (2008)
9. Leibe, B., Leonardis, A., Schiele, B.: Robust object detection with interleaved categorization and segmentation. IJCV **77** (2008) 259–289
10. Maji, S., Malik, J.: Object detection using a max-margin hough transform. In: CVPR, Los Alamitos, CA, USA, IEEE Computer Society (2009) 1038–1045
11. Chum, O., Zisserman, A.: An exemplar model for learning object classes. In: CVPR. (2007)
12. Vedaldi, A., Gulshan, V., Varma, M., Zisserman, A.: Multiple kernels for object detection. In: ICCV. (2009)
13. Gu, C., Lim, J., Arbelaez, P., Malik, J.: Recognition using regions. CVPR (2009) 1030–1037
14. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. PAMI **22** (2000)
15. Felzenszwalb, P., Huttenlocher, D.: Efficient graph-based image segmentation. IJCV **59** (2004)
16. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: From contours to regions: An empirical evaluation. CVPR (2009) 2294–2301
17. Sharon, E., Galun, M., Sharon, D., Basri, R., Brandt., A.: Hierarchy and adaptivity in segmenting visual cues. Nature (2006)
18. Hoiem, D., Efros, A.A., Hebert, M.: Geometric context from a single image. In: ICCV. (2005)
19. Malisiewicz, T., Efros, A.A.: Improving spatial support for objects via multiple segmentations. In: BMVC. (2007)
20. Russell, B.C., Efros, A.A., Sivic, J., Freeman, W.T., Zisserman, A.: Using multiple segmentations to discover objects and their extent in image collections. In: CVPR. (2006)
21. Stein, A., Stepleton, T., Hebert, M.: Towards unsupervised whole-object segmentation: Combining automated matting with boundary detection. In: CVPR. (2008)
22. Gould, S., Fulton, R., Koller, D.: Decomposing a scene into geometric and semantically consistent regions. In: ICCV. (2009)
23. Walther, D., Koch, C.: 2006 special issue: Modeling attention to salient proto-objects. Neural Networks **19** (2006) 1395–1407
24. Liu, T., Sun, J., Zheng, N., Tang, X., Shum, H.: Learning to detect a salient object. In: CVPR. (2007) 1–8
25. Alexe, B., Deselaers, T., Ferrari, V.: What is an object? In: CVPR, CVPR (2010)
26. Carreira, J., Sminchisescu, C.: Constrained parametric min cuts for automatic object segmentation. CVPR (2010)
27. Rother, C., Kolmogorov, V., Blake, A.: "grabcut": interactive foreground extraction using iterated graph cuts. ACM Trans. Graph. **23** (2004) 309–314
28. Russell, B.C., Torralba, A., Murphy, K.P., Freeman, W.T.: LabelMe: a database and web-based tool for image annotation. Technical report, MIT (2005)
29. Hoiem, D., Efros, A.A., Hebert, M.: Recovering surface layout from an image. IJCV **75** (2007) 151–172
30. Szummer, M., Kohli, P., Hoiem, D.: Learning crfs using graph cuts. In: ECCV. (2008)