To appear, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011.

Efficient Region Search for Object Detection

Sudheendra Vijayanarasimhan
University of Texas at Austin
svnaras@cs.utexas.edu

Kristen Grauman
University of Texas at Austin
grauman@cs.utexas.edu

Abstract

We propose a branch-and-cut strategy for efficient region-based object detection. Given an oversegmented image, our method determines the subset of spatially contiguous regions whose collective features will maximize a classifier's score. We formulate the objective as an instance of the prize-collecting Steiner tree problem, and show that for a family of additive classifiers this enables fast search for the optimal object region via a branch-and-cut algorithm. Unlike existing branch-and-bound detection methods designed for bounding boxes, our approach allows scoring of irregular shapes—which is especially critical for objects that do not conform to a rectangular window. We provide results on three challenging object detection datasets, and demonstrate the advantage of rapidly seeking best-scoring regions *rather than subwindow* rectangles.

1. Introduction

Object detectors determine whether a given object category is present in an image and estimate its spatial support. Many state-of-the-art approaches approximate the object's extent with a rectangular window: after training a classifier to distinguish objects from non-objects, a sliding window is used to exhaustively search a novel image for the subwindow that yields the best classifier score. In spite of its simplicity, this approach is responsible for a number of state-of-the-art results. However, sliding window detection has well-known fundamental weaknesses: (1) the computational expense of searching all windows is tremendous, which leads to sampling heuristics that may miss the object's best region of support, and (2) not all objects are box-shaped, which leads to representations polluted by features not belonging to the object.

Recent work offers various ways to avoid sliding windows and improve localization efficiency [1, 2, 3, 4]. Particularly relevant to our work, recently introduced *branch-and-bound* schemes show how to efficiently maximize certain classifier functions over candidate rectangles [5, 6, 7] and polygons [7]. Such approaches provide a significant speed-up over traditional sliding windows, without any loss in accuracy compared to an exhaustive search. However, they assume that the object fits well inside rectangular or coarse polygonal regions. This is rather restrictive, both because a box gives imprecise detection results for non-boxy objects (snakes, giraffes, cats, etc.), and also because "extra" features within a window may actually mislead the detector at test time, causing it to miss the object entirely (e.g., consider an object surrounded by clutter features that yield large negative classifier responses). See Figure 1 (a)-(c).

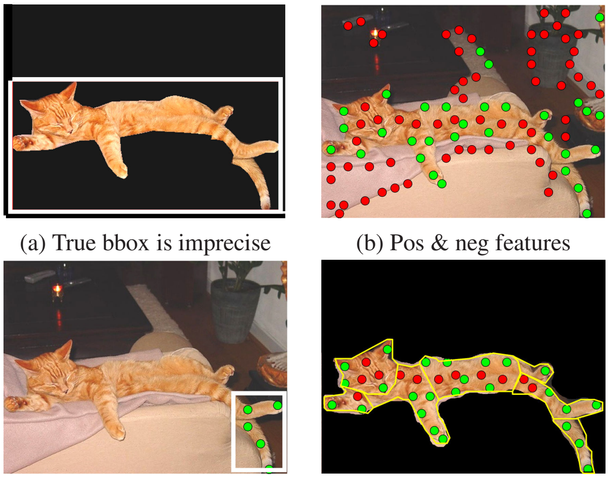
(a) True bbox is imprecise
(b) Pos & neg features
(c) Best subwindow
(d) Best connected subregions

Figure 1. Windowed detection vs. region subgraphs. (a) A box can give imprecise detections (e.g., only 30% of the cat's bounding box is actually foreground). (b) Image with local features mapped to a linear cat classifier's responses. Dark red dots denote negatively weighted features; light green dots denote positively weighted features. (c) A window detector misses most of the object, since the rectangle that would include the full object also includes a misleading number of negative features. (d) A region-based detector that sums classifier responses within *connected subregions* can more accurately detect non-boxy objects. Our main contribution is to show how to obtain this best-scoring region efficiently.

To address these limitations, we introduce a *branch-and-cut* approach that efficiently localizes the best-scoring *region* for an object category's classifier. Given a test image, we first divide it into segments and construct a region-graph:

the nodes are segments, and the edges link any two adjacent segments. Next we compute a score for each component region that indicates how much it could contribute to a possible detection of the object of interest (whether positively or negatively), where the scoring function is determined by a classifier having a specified form. Then, the goal is to identify the subset of connected regions (nodes) whose summed scores are maximal—or in other words, whose features collectively give the highest response for the classifier. See Figure 1 (d).

Obtaining the best-scoring region by exhaustively evaluating all possible subsets of connected nodes would require time exponential in the number of nodes in the graph. However, we show that for a certain family of additive functions (which includes a linear SVM on bags-of-features, nearest-neighbor local feature classifiers, or boosted classifiers), the problem of maximizing the detection score over all possible regions is equivalent to finding the *maximum-weight connected subgraph* (MWCS). While the MWCS problem is NP-complete [8], it can be transformed into an instance of the prize-collecting Steiner tree problem (PCST) [9], which can be efficiently solved in practice for our problem setting. We show this formulation is applicable for the "node-only" subgraph selection problem defined above, enabling fast and optimal solutions. We further show how to include edge weights *between* regions to favor connections according to learned category-specific models of inter-segment contours.

Our approach offers several advantages over existing branch-and-bound object detection schemes. It is powerful enough to model objects of any shape within the region-graph, and thus is not limited to rigid, boxy categories. Because it can avoid including extra clutter features surrounding the true object of interest, it offers more precise localization. We demonstrate our method with three datasets, and show that it outperforms both state-of-the-art fast detection methods restricted to rectangles, as well as a related CRF-based approach with a global connectivity potential.

2. Related Work

Sliding window detection methods are well-suited for rigid objects with a fairly regular appearance pattern, and have been quite successful for detecting certain categories [1, 10]. However, the high computational cost of evaluating the classifier over a large set of windows is a severe limitation.

The *efficient subwindow search* (ESS) algorithm of [5], and subsequent extensions [6], provide branch-and-bound schemes that efficiently identify the rectangle in the image that maximizes certain classifier functions. While these methods provide significant speed-ups over sliding window search, they are best-suited for boxy objects as discussed above. An extension of ESS shows how to detect composite bounding boxes (composed of k boxes) or k-sided polygons [7]. However, the fixed k parameter must be pre-selected, and the value has considerable impact on the computational cost (the authors recommend values $k \leq 5$ for this reason), limiting the actual representable polygon shapes. Other sliding window alternatives include an approximation for a Steiner tree problem that selects a small number of image regions on which to run a classifier [11], and a ratio-contour algorithm to find polygonal regions with a strong classifier score [12]. In contrast to any existing branch-and-bound detection algorithm, we show how to efficiently compute the optimal best-scoring subset of connected *regions*, which can be of any shape.

Voting methods that use the Generalized Hough Transform avoid exhaustive search, efficiently aggregating evidence for an object's presence based on local parts [2, 3, 6, 13]. If training exemplars are segmented, they can also provide pixel-level detection hypotheses [2, 3]. On the other hand, they generally require roughly similar viewpoints, with minor pose variations between training and test images. By adopting a region-based bag of features representation, our framework provides greater flexibility to such transformations. Furthermore, unlike the Hough-based techniques, our method is guaranteed to return the region in the image that maximizes the object's classifier output.

Another class of techniques predicts pixel-level class labels—as a foreground-background map [14, 15, 16], or as a full multi-class labeling with random field models [17, 18, 19, 20]. Like our strategy, these methods can directly estimate the support for an object without resorting to rectangular bounds. However, the foreground methods often work best when the object permits a consistent 2d shape model. Conditional random field (CRF) models maximize the probability of the joint label assignment, and can be efficiently trained [19]. Among these models, most relevant to our approach is the global potential for log-linear CRFs that enforces that the output labeling be connected [16]; the authors give an approximate solution that relies on an LP relaxation. In contrast, our approach provides globally optimal solutions, it accommodates a family of additive classifiers, allows anytime solutions, and introduces class-specific models of inter-segment contours. Direct comparisons with this model (Section 4) show our method's optimal solutions have a clear advantage.

3. Approach

We first briefly overview the entire approach: Given training images in which the spatial extent of the object of interest is marked at the pixel level, we train an additive classifier to distinguish that object category from any other (Section 3.1). Given a new image, we oversegment it into subregions, extract the image features associated with each subregion, and pre-compute its resulting contribution to the classifier response. This yields a vertex-weighted region-graph, where the weights are the components of the classifier score (Section 3.2). We show that the problem of obtaining the best-scoring contiguous set of regions in this graph is equivalent to the MWCS problem; this in turn can be transformed into the PCST problem, which is efficiently solvable in practice with a branch-and-cut algorithm (Section 3.3). To incorporate inter-region contour cues, we show how to introduce learned edge costs into the graph (Section 3.4). The resulting detection method rapidly determines the region within the image that yields the maximal classifier response.

3.1. Objective and Classifiers

Suppose we have a classifier function $f : \mathrm{R} \rightarrow \mathbb{R}$ that scores a region $R \in \mathrm{R}$ according to how strongly it belongs to a particular object category. Assuming it is a reliable predictor, true object regions would yield high scores, whereas other objects, background, and partial object/non-object regions would yield lower scores. For detection, our goal is to determine the region within a novel image I that maximizes the score: $R^* = \arg\max_{R \in I} \; f(R)$.

Note that the best-scoring region can be of an arbitrary shape. While sliding window approaches also seek the portion of the image that maximizes the classifier response, they do so only over the restricted domain of rectangles.

We require the classifier to have the property that features computed within local regions of the image can be combined additively to obtain the classifier response for a larger region. This is what will allow us to decompose the classifier response spatially across the image into the region-graph. A linear kernel SVM applied to a bag-of-features representation has this property, as shown in [5].

In the bag-of-features representation, a vocabulary of K visual words is obtained by clustering a sample of local features (e.g., SIFT) from the training images. An image region with N local features is described by the set $R = \{(\boldsymbol{x}_i, \boldsymbol{v}_i)\}_{i=1}^N$, where each $\boldsymbol{x}_i$ refers to the feature position and $\boldsymbol{v}_i$ is the local descriptor. This set in turn can be mapped to a bag-of-features histogram $h_v(R)$ by mapping each feature $\boldsymbol{v}_i$ to its closest visual word c_i, and recording the frequency of words in a K-dimensional vector. The subscripts v reflect that these histograms are binning visual words. We use superscripts to index a vector.

Using the histograms from the segmented training examples, we learn a linear SVM decision function: $f(R) = \beta_v + \sum_i \alpha_v^i \langle h_v(R), h_v(R_i) \rangle$, where i indexes the training examples, and α_v, β_v denote the learned weights and bias. Exploiting the linearity of the scalar product, Lampert et al. rewrite the expression for $f(R)$ as a sum over per-feature contributions [5]. Let $h_v^j(R)$ denote the count in the j-th bin of the region histogram $h_v(R)$. For every $1 \leq j \leq K$, the j-th word is associated with a weight w_v^j: $w_v^j = \sum_i \alpha_v^i h_v^j(R_i)$, and the classifier response for a region can thus be rewritten as:

$$f(R) \;=\; \beta_v + \sum_{j=1}^{K} w_v^j h_v^j(R) \;=\; \beta_v + \sum_{i=1}^{N} w_v^{c_i}, \quad (1)$$

where again c_i is the index of the visual word that feature $\boldsymbol{v}_i$ maps to, $c_i \in [1, K]$. Thus, the score of a region is the sum of its N features' word weights. The bias term β_v can be ignored for the purpose of maximizing $f(R)$. Bounds for this scoring function are shown in [5] for the case where R ranges over rectangles, and they enable an efficient branch-and-bound maximization procedure for subwindow search. In the following we show how to instead efficiently maximize $f(R)$ when R ranges over arbitrarily shaped regions.[1]

While we focus on linear SVMs and histogram features in this paper, our approach can also accommodate other models where the score of an object is additive in the scores of its individual features. This includes a boosted classifier linearly combining localized weak features, the nearest-neighbor recognition method of [21]—where an ROI is scored by the summed minimum distances from every local feature within it to the class-labeled training features—and, using recent work by [22], even some non-linear SVMs.

3.2. Constructing a Novel Image's Region-Graph

Next we define how a novel test image is mapped to a region-graph $G = (V, E)$, where V is a set of vertices and E are the edges. To define the vertices, we divide the novel image into a set of superpixels. We use superpixels as the smallest spatial tokens (rather than pixels) to enforce some local coherency. Each superpixel is thus a node in the image's region-graph, and we insert an edge between any two superpixels that share a boundary. Now a candidate region R is any subset of connected nodes in this graph, or in other words, a *connected subgraph*.

Each vertex $\mathrm{v} \in V$ has an associated weight, $\omega(\mathrm{v})$, which represents its contribution to the classifier score. Note that this weight can be positive or negative initially. We consider two ways to weight a superpixel vertex, which vary only in the type of feature used to construct the histogram $h_v(R)$:

Point features: For this representation, each descriptor $\boldsymbol{v}_i$ is a local point feature (we use SURF [23]), and the weight assigned to a superpixel vertex is the sum of the word-weights for all local features located within that superpixel: $\omega(\mathrm{v}) = \sum_{\boldsymbol{x}_i \in \mathrm{v}} w_v^{c_i}$. These descriptors are preferable for deformable objects and/or to maintain greater viewpoint invariance.

Shape features: Alternatively, each superpixel is mapped to a single shape descriptor. Specifically, we describe each region with the histogram of oriented responses

[1] More precisely, any connected set of segments.

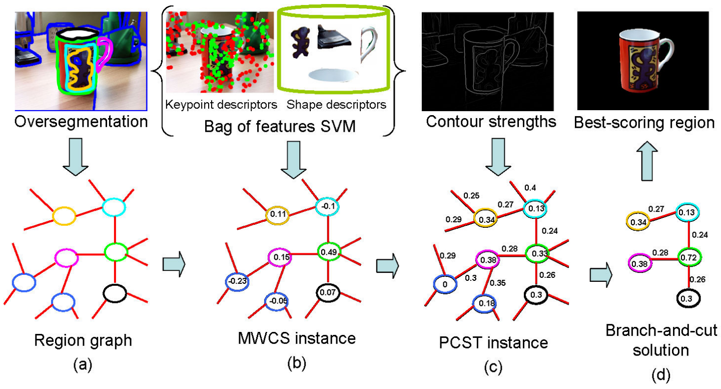

Figure 2. Approach summary. (a) We oversegment the test image, and construct a region-graph. (b) A region's node weight is its contribution to the classifier response. The optimal contiguous set of regions is equivalent to the MWCS problem on the vertex-weighted region-graph. (c) The MWCS is transformed into the PCST problem, and we incorporate class-specific inter-region contour cues by adding edge costs. (d) The best-scoring region is obtained by efficiently solving the PCST instance with a branch-and-cut algorithm.

of a contour and edge detector [24], binned into spatial grid cells, following [3]. Each $\boldsymbol{v}_i$ is then a concatenated contour+edge histogram, and the visual vocabulary is formed by quantizing this feature space. For a training region R, we form $h_v(R)$ by mapping the shape descriptors of all superpixels within it to their visual words. For a novel image, the vertex weight is $\omega(\mathrm{v}) = w_v^{c_i}$, where c_i is the single word associated with that superpixel's shape descriptor. These descriptors are preferable for objects with parts defined by their shape. We experiment with both features.

After considering other alternatives for defining connectedness in the region-graph—such as four-connected neighborhoods on dense pixel-based features, or a Voronoi tesselation based on the feature positions—we settled on using segment adjacency because it can favorably enforce a compact structure. In particular, the segment-based graph helps avoid selecting subgraphs with short paths that connect only the positively-weighted nodes, which we found gave undesirable spindly detections.

In order to impose spatial constraints, we restrict edges between shape nodes whose visual words co-occurred in any of the training images, and whose predicted object extents overlap by at least 50%.

3.3. Efficient Region Search (ERS)

This section defines our Efficient Region Search (ERS) procedure to maximize $f(R)$. We show that the objective can be reduced to the maximum-weight connected subgraph problem (MWCS), which is defined as follows:

MWCS PROBLEM: *Given a connected undirected, vertex-weighted graph $G = (V, E)$ with weights $\omega : V \rightarrow \mathbb{R}$, find a connected subgraph $T = (V_T \subseteq V, E_T \subseteq E)$ of G, that maximizes the score $W(T) = \sum_{\mathrm{v} \in V_T} \omega(\mathrm{v})$.*

Here, the set of vertices V are the superpixels, and edges in E connect pairs of superpixels that share a boundary. The weight of a vertex $\omega(\mathrm{v})$ is the superpixel's classifier score, as defined above. The best-scoring subgraph identifies the most likely region for the object of interest. Note the two dual views of the representation: at training time, we think of the ground truth foreground region in terms of its total histogram of words, whereas at test time, we think of candidate connected subgraphs that aggregate the segment vertices' classifier scores.

Since the graph edges are unweighted under the above definition, any solution to the MWCS problem is equivalent to its spanning tree. If all vertices were weighted with positive values, then the best-scoring subgraph would contain all vertices, while if all vertices were negatively-weighted then the solution would simply be the single vertex with the highest weight. However, when we have both positive and negative node weights, as is the case for the SVM, the MWCS problem is NP-complete [8]. A brute force solution would enumerate all possible subsets of vertices, check if they are connected, and tally their scores—which would take time exponential in the number of nodes.

However, the MWCS problem can be transformed into an instance of the prize-collecting Steiner tree problem (PCST), as shown in [9]. The value in doing so is that the optimal solution may be efficiently obtained in practice, for some cases. The PCST is a well-known problem occurring frequently in operations research and networking, and various optimal and approximate solutions exist. Formally:

PCST PROBLEM: *Given a connected undirected vertex- and edge-weighted graph $G = (V, E, c, p)$ with vertex profits $p : V \rightarrow \mathbb{R}^{\geq 0}$ and edge costs $c : E \rightarrow \mathbb{R}^{\geq 0}$, find a connected subgraph $T = (V_T \subseteq V, E_T \subseteq E)$ of G that maximizes the profit:*

$$P(T) = \sum_{\mathrm{v} \in V_T} p(\mathrm{v}) - \sum_{\mathrm{e} \in E_T} c(\mathrm{e}). \tag{2}$$

Note that in the PCST, both vertex profits and edge costs must be positive real numbers. This, along with the fact that edge costs contribute negatively to the total profit in Eqn. 2, means that any solution of the PCST is a tree.

Let G be an instance of the MWCS with both positive and negative vertex weights, and let $\omega' = \min_{\mathrm{v} \in V} \omega(\mathrm{v})$ be its smallest vertex weight. One can then construct an equivalent instance of the PCST problem G', by first copying the set of vertices and edges from G, and then setting the vertex profits to $p(\mathrm{v}) = \omega(\mathrm{v}) - \omega'$ for all $\mathrm{v} \in V$, and setting the edge costs to $c(\mathrm{e}) = -\omega'$ for all $\mathrm{e} \in E$. This is a valid instance of the PCST because both the profits and the costs are positive; the optimal solutions of the PCST and MWCS instances are related by $W(T) = P(T) - \omega'$. A proof of this transformation is given in [9].

Once our region-graph is mapped to the PCST problem, we use the mathematical programming approach of [25] to identify the maximum weight connected subgraph. We use this algorithm because it provides solutions that are provably optimal, and we show it is very efficient in practice for most region-graphs of reasonable size (hundreds of superpixels). See Sec. 4 for empirical run times.

The algorithm first applies a series of pre-processing steps to simplify the input graph, such as removing edges between two nodes if the edge cost is larger than the shortest path between the two nodes. Then, the graph is transformed into a directed graph by duplicating every undirected edge and introducing a root node. An integer linear program (ILP) is built on the transformed graph by introducing a binary integer variable for every vertex and edge to model its presence or absence in the solution. Then, an exponential number of *cut* constraints or *connectivity inequalities* are used to model the connectedness of the solution. Finally, a branch-and-cut algorithm is applied to efficiently solve the PCST. An LP approximation of the problem provides tight bounds for each stage of the branching tree, and the connectivity constraints are iteratively added only when the current solution violates them. Violated cut constraints are efficiently obtained using a maximum flow algorithm in a support graph with arc-capacities given by the current formulation. See [25] for details.

The formulation also permits one to obtain "anytime" solutions, and multiple sub-optimal solutions along with the optimal one, which we can utilize to detect multiple instances of an object in the same image, or to sample among the classifier's most confident regions.

3.4. Efficient Region Search with Contours (ERS-C)

Our ERS approach as defined in the previous section uses only *vertex* weights to obtain the best-scoring subgraph. While this is sufficient to strictly capture the contiguous regions that maximize the classifier response, some undesirable effects may arise. Specifically, it may include background regions if the classifier maps their features incorrectly to positive weights.

To address this issue, we introduce *edge* weights between pairs of adjacent superpixels based on the strength of their intervening contour. Our strategy must take into account two things—first, just as with the vertex weights, we want the best-scoring distribution of contours to be identifiable based on the sum of spatially distributed scores (the edge weights), and second, the internal contour properties may be class-specific (and thus should be learned).

To this end, we model an object region R as a subtree within the region-graph (as opposed to a subgraph in Sec. 3.3) whose score is the sum of its node and edge weights. Then, we propose a bag-of-*contour-strengths* histogram vector that captures the statistics of the internal contours within an object, and whose scores can be directly mapped to the edge costs $c(\mathrm{e})$ in the PCST instance. Intuitively, we expect the contours between an object and its background to be highly salient, and therefore we would like to learn weights such that the scores of segmentations that cross object boundaries are reduced.

Formally, the distribution of contour strengths $h_e(R)$ for an object region R is an L-bin histogram, where each bin represents a given range in the (scalar) contour saliency strength. To compute the contour saliency, we use the ultrametric associated with the hierarchical segmentation given by the method of [26]. Essentially, two adjacent regions more distant in the agglomerative segmentation tree will have a stronger contour between them.

Given these histograms, we define the Efficient Region Search-Contour (ERS-C) classifier score: $f'(R) = w_v h_v(R) - w_e h_e(R)$, and show below that it can be mapped to a PCST instance. Note that by subtracting the contour term, $f'(R)$ produces lower scores for regions crossing strong object boundaries, which means the optimal solution can better exclude background regions.

Analogous to the ERS classifier in Sec. 3.1 (Eqn. 1), we can decompose the ERS-C score as:

$$f'(R) = \sum_{i=1}^{N} w_v^{c_i} - \sum_{j=1}^{M} w_e^{s_j}, \tag{3}$$

where w_e are the edge weights (which we define below), $w_v^{c_i}$ are the vertex weights as defined in Sec. 3.1, $s_j \in [1, L]$ is the bin index of $h_e(R)$ into which the contour-strength of the j^{th} contour within the region falls, and M denotes the total number of contours in the region R.

We formulate the procedure for learning the contour histogram weights, w_e, in the structured SVM learning framework (see Algorithm 1 in [27]). In particular, we use the cutting plane algorithm with zero-one loss, and define the loss to be one for regions with an intersection score of less than 50% with the ground truth. The algorithm for the zero-one loss is a special case of the general cutting plane algorithm; at each iteration, we must identify the highest scoring region R that is incorrectly predicted, and add it to the current working set of constraints.

By virtue of our construction, the highest scoring region for f' can be efficiently obtained by constructing a PCST instance and using the same branch-and-cut procedure above to solve it (see Sec. 3.3). To form the PCST instance, we introduce an edge cost using the weighted contour strengths between the two superpixels: $c(\mathrm{e}) = w_e^{s_j}$ (see Eqn. 2). Then we apply the cutting plane algorithm together with our branch-and-cut procedure to learn the contour weights w_e.[2] See Figure 2 for a recap of the entire approach.

While the use of intervening contour cues bears some resemblance to CRF models developed for pixel labeling (e.g. [18, 20]), note that in our model the contribution of each contour strength level is learned at the level of the object category, rather than prescribed uniformly for all nodes and classes. We directly compare ERS-C to a CRF below.

4. Results

We compare to the two most relevant existing methods: (1) the state-of-the-art efficient subwindow search (ESS) method [5]—which efficiently provides the best result possible with bounding box search—and (2) the global connectivity CRF model of [16]—which efficiently provides an approximate region-based solution. Both methods permit the same form of histogram-based additive classifier. Relative to the former, our goal is to demonstrate the advantage of using our fast subgraph approach to search over *regions*, as opposed to searching over rectangular windows. Relative to the latter, our goal is to demonstrate the advantage of efficiently computing the *optimal* solution, as opposed to an approximate relaxation. For either baseline, we use the same data as selected by the authors.

Datasets: We use three datasets: the PASCAL VOC 2007, the ETHZ Shapes [13], and the PASCAL VOC 2008. We test with the PASCAL 2007 cats and dogs (659, 839 images resp.), following [5], since they are deformable objects with wide pose variation—aspects which make the linear SVM on bags of point features very effective. We use the provided *trainval* and *test* splits, and obtained ground truth segmentations. The ETHZ Shape dataset consists of 255 images of five shape categories (Applelogos, Bottles, Mugs, Giraffes, Swans). It is also a good testbed to emphasize the limitations of bounding boxes, since most objects are non-boxy. We use half the examples per category for training, the rest for testing. Finally, we test with all 20 categories in the PASCAL VOC 2008 segmentation dataset, a challenging benchmark [29].

In addition to the usual bounding box accuracy metric, we report *pixel-level* precision-recall and overlap scores with ground truth segmentations. These scores offer a more accurate evaluation of detections, particularly for non-rectangular objects.

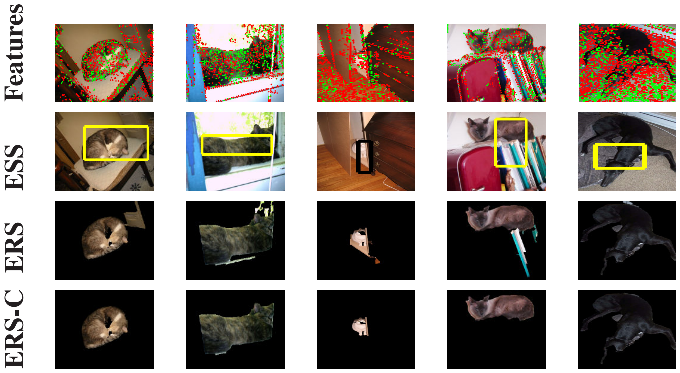

Figure 3. Example PASCAL07 detections. First row shows images with the sign of the point feature scores (sign$(w_v^{c_i})$) superimposed: red dots denote negatively weighted features, green dots denote positive features (*best viewed in color*). Remaining rows show detections returned by ESS and ERS. Both methods seek the region that will accumulate the most green points while avoiding including excessive red ones. However, since ESS is restricted to finding the max scoring *rectangle*, it often over/underestimates the object's extent. Our method provides precise arbitrarily shaped detections. Last row illustrates how ERS-C can exclude spurious background regions.

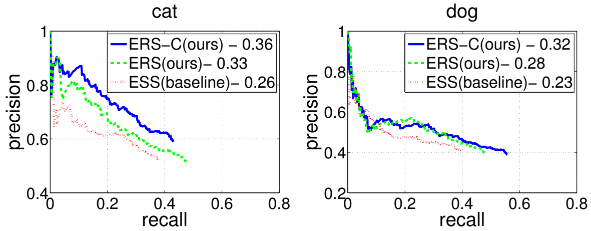

Figure 4. Detection accuracy on PASCAL07 objects. Numbers in legends report mAP. Both of our methods outperform the baseline.

Implementation details: For *point features*, we use SURF [23] extracted at Canny edge points, and quantize the training points into $K = 1000$ visual words using K-means. We collect negative examples from both other objects as well as random background regions. We train a linear SVM and map each visual word to its score w_v^j. We select the C parameter with cross-validation. To construct the region-graph, we obtain $\sim$100 regions per image using [26]. We compute per-pixel contour strengths from the resulting ultrametric contour map. We set $L = 10$ to coarsely bin the scalar contour strengths; we have not tried other values.

For *shape features*, we describe each region with a 4×4 spatial grid scaled to the region size, where each cell bins the normalized gPb and Canny edge responses according to their orientations. To collect responses at multiple scales, we also tally the responses from blurring the gPb map with Gaussians of two scales (σ=5,10). To form $h_v(R)$, we quantize to $K = 50$ shape descriptors per class and an additional $K = 750$ for background regions.

For the ESS baseline, we use the authors' publicly available code and evaluate both methods with exactly the same point features and classifiers, to enable a direct compari-

[2] In our implementation, we simply represent $w_v h_v(R)$ as the score of a pre-trained SVM classifier as done in [28] and other recent work. This simplifies and speeds up the learning procedure.

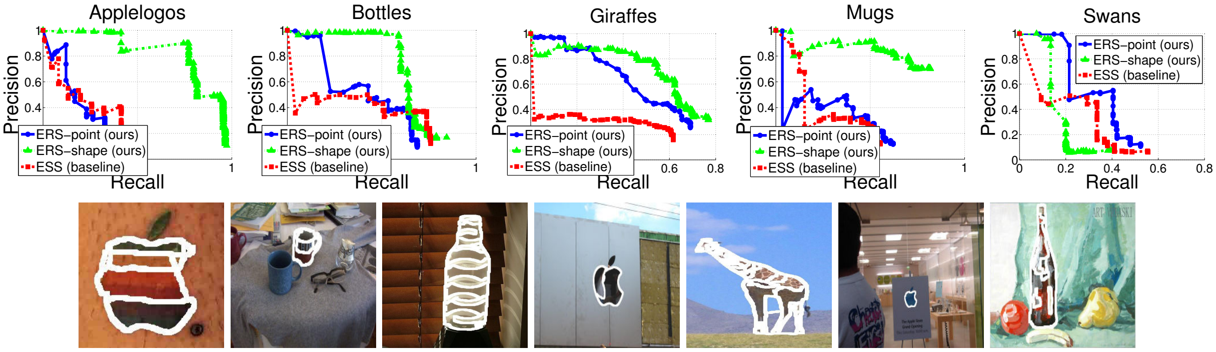

Figure 5. Detection accuracy (top row) and example detections by our method (bottom row) on the ETHZ categories.

son. For both, we evaluate the top-scoring region/rectangle per image, and obtain its confidence score using a χ^2-kernel SVM on the same bag-of-words features, following [5]. For comparisons with the CRF approach of [16], we use the features and superpixels kindly provided by the authors.

PASCAL 2007 results: Figure 4 compares the accuracy of our approaches versus ESS [5], and Figure 3 shows example detections. For this result we pose a pure *localization* task, following [5], where the detector is scored only on images where the object is present.[3] For both categories, maximizing the classifier response over the region-graph (ERS) yields much better accuracy than rectangular windows (ESS). In addition, including the edge costs (ERS-C) further boosts precision. Even under the PASCAL bounding box metric, our method is about 70% more accurate than ESS (mean AP of 27.2 for ERS, compared to 16.0 for ESS). This supports our claim that not only does a region-based detector allow precise localization, but it can also avoid errors due to including misleading features.

The mean overlap scores for our approach and ESS are 31.9%, 22.5% resp. for cats and 29.4%, 23.1% resp. for dogs (this measure normalizes for per-object area). Our improvements are most pronounced for the object instances that do not fit well in a window.

ETHZ results: Figure 5 shows the results on ETHZ, scored for the full *detection* task where we run on all images (positive or negative). We apply our ERS method using either the point features (blue curves), or the shape features (green curves). ESS is only applicable to the point features (red curves). Again we see the clear accuracy advantage our strategy offers. Comparing both methods using the point features, we see the most significant gains for the Giraffes and Swans, both of which are poorly captured with a rectangular window. For Applelogos and Mugs, ERS-point's accuracy is close to ESS's, which makes sense as these objects are more boxy. Overall, the mAP of ERS is from 9% to 90% better than ESS using the same features. Using shape features, we see dramatic gains for almost all objects, since the ETHZ objects have parts well-described by their shape (e.g., mug handle, bottle neck). Again, even with the bounding box metric our approach is 19% better than ESS (30.1 vs. 25.3 mAP).

To study the importance of learning category-specific contour strength weights, we conducted an experiment where the contour weights were swapped among the five categories (weights learned for Mugs used for Bottles, Swans for Applelogos, etc.). When swapped to use the wrong model, the mAP scores of Applelogos, Bottles, and Mugs drop to 19.9, 33.5, and 25.0 respectively (versus 25.5, 41.2, 30.4). The Giraffe and Swan results remained close to the ERS scores. This confirms the importance of our learning procedure in Sec. 3.4; the distribution of internal contour strengths varies in informative ways per category.

PASCAL 2008 results: Finally, we compare our approach against the method of [16], which provides an approximate algorithm for using a global connectivity potential within a standard CRF. Again, to offer the most direct comparison, we use the exact same features, and follow the setup defined in [16], which computes the 3-fold cross-validation accuracy of the foreground segmentation using the PASCAL overlap criterion [29].

Figure 6 shows the 20-class results and some example detections. Our ERS approach outperforms the baseline on 17 out of 20 categories, and improves the mean overlap accuracy significantly (0.274 vs. 0.228). On average we obtain a 24% gain per class, with more than 50% increases in some cases (e.g. *pottedplant, sheep, tvmonitor*). Our ERS-C variant also improves four classes relative to ERS. Given that we are using identical features, this result indicates the value of obtaining the optimal solution with our approach as opposed to the approximate inference procedure in [16].

Computation time: Our approach is very efficient in practice for all the datasets tested, showing the suitability of this PCST reduction for our problem setting. On average it converges in 0.29 seconds, which is similar to ESS. The longest time taken over any single test image was 5.8 secs.

[3] Note that the curves we show for ESS are not identical to those in [5] because we use slightly different features, and score at the pixel-level.

	aerop.	bicyc.	bird	boat	bottle	bus	car	cat	chair	cow	dinin.	dog	horse	motor.	person	potte.	sheep	sofa	train	tvmon.	mean
ERS	0.324	**0.109**	**0.268**	0.262	**0.121**	**0.405**	**0.244**	0.389	0.120	**0.324**	**0.300**	**0.288**	**0.280**	**0.337**	**0.257**	**0.119**	0.394	**0.224**	**0.453**	0.259	**0.274**
ERS-C	0.325	0.058	0.257	0.262	0.104	**0.405**	0.240	**0.399**	0.097	0.319	0.300	0.249	0.261	0.280	0.249	0.107	**0.404**	0.210	0.445	**0.272**	0.262
CRF [16]	**0.380**	0.091	0.202	**0.275**	0.115	0.391	0.185	0.311	**0.121**	0.236	0.269	0.244	0.209	0.268	0.194	0.075	0.249	0.200	0.393	0.152	0.228

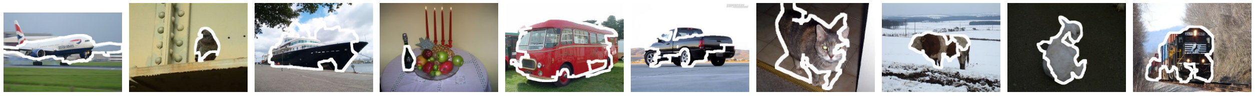

Figure 6. Detection overlap accuracy compared to the global connectivity CRF [16] on the PASCAL 2008 (top), and example detections by our method (bottom). Our optimal solution leads to significantly more accurate results on this challenging dataset.

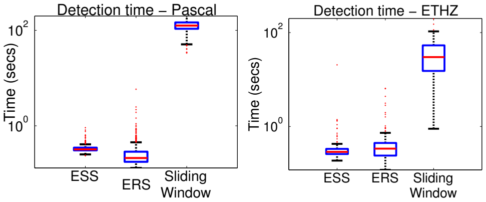

Figure 7. Our search times on both PASCAL and ETHZ are similar to ESS's, and both are orders of magnitude faster than sliding window search. (Note time is on log scale.)

Figure 7 shows the detection times on a log scale for ETHZ and PASCAL, as compared to a sliding window that searches across 30 scales. Note that brute force search for the max subgraph would take time exponential in the number of nodes, $\sim 2^{100}$, and cannot be practically tested. Our method offers two orders of magnitude speed-up, and (unlike ESS) it permits pixel-level detections of any shape.

5. Conclusions

We introduced an efficient branch-and-cut method for region-based detection, and with three challenging datasets we demonstrated its advantages over both existing branch-and-bound methods that are limited to searching rectangles and a CRF model. Our approach is the first that can efficiently identify the subregion that maximizes the additive detector's scoring function. In future work we will examine the alternate classifiers accepted by our model.

Acknowledgements Many thanks to Ivana Ljubic and Gunnar Klau for sharing code for solving the PCST problem, and Sebastian Nowozin for sharing features for PASCAL 2008. This research was supported in part by DARPA CSSG N10AP20018 and the Luce Foundation.

References

[1] Viola, P., Jones, M.: Rapid Object Detection using a Boosted Cascade of Simple Features. In: CVPR. (2001)

[2] Leibe, B., Leonardis, A., Schiele, B.: Combined Object Categorization and Segmentation with an Implicit Shape Model. In: Workshop on Statistical Learning in Computer Vision. (2004)

[3] Gu, C., Lim, J., Arbelaez, P., Malik, J.: Recognition Using Regions. In: CVPR. (2009)

[4] Murphy, K., Torralba, A., Eaton, D., Freeman, W.: Object Detection and Localization Using Local and Global Features. In: Towards Category-Level Object Recognition. LNCS (2006)

[5] Lampert, C., Blaschko, M., Hofmann, T.: Beyond Sliding Windows: Object Localization by Efficient Subwindow Search. In: CVPR. (2008)

[6] Lehmann, A., Leibe, B., van Gool, L.: Feature-Centric Efficient Subwindow Search. In: ICCV. (2009)

[7] Yeh, T., Lee, J., Darrell, T.: Fast Concurrent Object Localization and Recognition. In: CVPR. (2009)

[8] Ideker, T., Ozier, O., Schwikowski, B., Siegel, A.F.: Discovering Regulatory and Signalling Circuits in Molecular Interaction Networks. Bioinformatics (2002)

[9] Dittrich, M.T., Klau, G.W., Rosenwald, A., Dandekar, T., Müller, T.: Identifying Functional Modules in Protein–Protein Interaction Networks. Bioinformatics (2008)

[10] Felzenszwalb, P., McAllester, D., Ramanan, D.: A Discriminatively Trained, Multiscale, Deformable Part Model. In: CVPR. (2008)

[11] Russakovsky, O., Ng, A.: A Steiner Tree Approach to Efficient Object Detection. In: CVPR. (2010)

[12] Zhang, Z., Cao, Y., Salvi, D., Oliver, K., Waggoner, J., Wang, S.: Free-Shape Subwindow Search for Object Localization. In: CVPR. (2010)

[13] Ferrari, V., Jurie, F., Schmid, C.: From Images to Shape Models for Object Detection. IJCV (2009)

[14] Borenstein, E., Ullman, S.: Class-Specific, Top-Down Segmentation. In: ECCV. (2002)

[15] Winn, J., Jojic, N.: LOCUS: Learning Object Classes with Unsupervised Segmentation. In: ICCV. (2005)

[16] Nowozin, S., Lampert, C.: Global Connectivity Potentials for Random Field Models. In: CVPR. (2009)

[17] He, X., Zemel, R., Carreira-Perpinan, M.: Multi-scale Conditional Random Fields for Image Labeling. In: CVPR. (2004)

[18] Shotton, J., Winn, J., Rother, C., Criminisi, A.: Textonboost: Joint Appearance, Shape and Context Modeling for Multi-class Object Recognition and Segmentation. In: ECCV. (2006)

[19] Szummer, M., Kohli, P., Hoiem, D.: Learning CRFs using Graph Cuts. In: ECCV. (2008)

[20] Kohli, P., Ladicky, L., Torr, P.: Robust Higher Order Potentials for Enforcing Label Consistency. In: CVPR. (2008)

[21] Boiman, O., Shechtman, E., Irani, M.: In Defense of Nearest-Neighbor Based Image Classification. In: CVPR. (2008)

[22] Vedaldi, A., Zisserman, A.: Efficient Additive Kernels via Explicit Feature Maps. In: CVPR. (2010)

[23] Bay, H., Tuytelaars, T., Van Gool, L.J.: Surf: Speeded Up Robust Features. ECCV (2006)

[24] Maire, M., Arbelaez, P., Fowlkes, C., Malik, J.: Using Contours to Detect and Localize Junctions in Natural Images. In: CVPR. (2008)

[25] Ljubic, I., Weiskircher, R., Pferschy, U., Klau, G., Mutzel, P., Fischetti, M.: An Algorithmic Framework for the Exact Solution of the Prize-Collecting Steiner Tree Problem. Mathematical Prog. (2006)

[26] Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: From Contours to Regions: An Empirical Evaluation. In: CVPR. (2009)

[27] Tsochantaridis, I., Joachims, T., Hofmann, T., Altun, Y., Singer, Y.: Large Margin Methods for Structured and Interdependent Output Variables. Journal of Machine Learning Research (2005)

[28] Wang, Y., Mori, G.: A Discriminative Latent Model of Object Classes and Attributes. In: ECCV. (2010)

[29] Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge (2008)