The Middle Child Problem: Revisiting Parametric Min-cut and Seeds for Object Proposals

Ahmad Humayun † Fuxin Li ‡† James M. Rehg †
† Georgia Institute of Technology ‡ Oregon State University
http://cpl.cc.gatech.edu/projects/POISE/

Abstract

Object proposals have recently fueled the progress in detection performance. These proposals aim to provide category-agnostic localizations for all objects in an image. One way to generate proposals is to perform parametric min-cuts over seed locations. This paper demonstrates that standard parametric-cut models are ineffective in obtaining medium-sized objects, which we refer to as the middle child problem. We propose a new energy minimization framework incorporating geodesic distances between segments which solves this problem. In addition, we introduce a new superpixel merging algorithm which can generate a small set of seeds that reliably cover a large number of objects of all sizes. We call our method POISE— "Proposals for Objects from Improved Seeds and Energies." POISE enables parametric min-cuts to reach their full potential. On PASCAL *VOC it generates ∼2,640 segments with an average overlap of* 0.81*, whereas the closest competing methods require more than 4,200 proposals to reach the same accuracy [24, 30]. We show detailed quantitative comparisons against 5 state-of-the-art methods on* PASCAL *VOC and Microsoft COCO segmentation challenges.*

1. Introduction

Figure-ground object proposal algorithms [5, 10, 24, 32, 4, 30] have recently become popular due to their successful application in object detection and semantic segmentation [16]. These methods can find the location and, possibly, the shape of an object, helping to improve recognition.

Malisiewicz and Efros [28] were the first to suggest generating a large pool of proposals for recognition. The first widely-used method was CPMC [5], which generates proposals by selecting a few seed regions as priors for object support, and performing *parametric min-cut (PMC)* on the MRF graph generated from each seed. More recently, alternatives to CPMC have been developed. Some of them generate segments from energy minimization via graph-cuts [19, 9, 31, 24]. Other popular methods perform agglomerative clustering [32, 4], or employ edge-based techniques [23, 30] to generate proposals.

We believe that a discrete energy minimization approach has the potential to produce better object segments than the current CRF models, but only if the graphical model and its parameters are carefully designed. This fits in well with the long history of graphical models in obtaining elegant, yet effective solutions to hard vision problems [20].

It has been observed empirically that PMC tends to produce segments which are either comparable in size to the seed region or extend almost to the full image (see Fig. 1 for an example). In particular, segments which are in the middle and often correspond to particularly salient object candidates are frequently missing, and therefore do not get the attention that they deserve. We refer to the generation of these missing segments as the *middle child problem*. We will demonstrate that this problem is an intrinsic property of existing PMC formulations [5, 9, 19] and cannot be solved simply by tuning parameters or exploring breakpoints exhaustively. The middle child problem is a significant barrier to the use of PMC to generate effective object proposals.

We propose an algorithm to solve the middle child problem using PMCs. After obtaining a segment at a particular parameter λ, we adjust the unary potentials according to the geodesic distance of all superpixels in the image with respect to the current segment. This facilitates the generation of medium-sized segments by lowering their energy. The approach is a modification of the PMC framework, thereby maintaining the nesting property [22] of the segments produced. The resulting algorithm's run-time is $\sim$3.5 seconds when generating 1,000 proposals.[1]

We also introduce a new superpixel merging algorithm for generating seeds. It utilizes an adaptive appearance thresholding strategy to generate a hierarchy of superpixels of varying sizes, so that more superpixels are generated in regions that have more internal variation and less are gener-

‡ This work was conducted while the 2nd author was at Georgia Tech.

[1] Multi-threaded run-time on Intel i7-3930K. Code available online.

(a) Boundaries [7] with seed

Figure 1: This example demonstrates the *middle child problem*. The seed is placed on the child's jersey, as shown in (a). The probabilistic boundary map suggests that it should be possible to produce a segment containing just the child in red. Each row shows parametric min-cuts produced by a method from the displayed seed. The top row demonstrates that RIGOR [19] is incapable of finding this segment and the bottom row shows our results. Unlike RIGOR, we are able to capture the *middle child* (red outlined).

ated in regions with uniform color. This approach generates a small set of reliable seeds that cover objects of all sizes and diverse appearances, and improves on previous algorithms for small and less salient objects.

These improvements result in a state-of-art object proposal algorithm. Our method requires many fewer proposals than its competitors to obtain the same accuracy. The performance of our algorithm is validated on two segmentation benchmarks: PASCAL VOC and MS COCO [27].

In §2 we review earlier proposal generation methods, and their role in detection pipelines. §3 explains in detail the causes and effects of the middle child problem. §4 gives an efficient solution to the problem. Our superpixel seed generation method is explained in §5. This is followed by evaluation in §6, which quantitatively demonstrates the effects of each of our contributions. We conclude in §7.

2. Related Work

Convolutional neural networks (CNN) have been leading the progress in object detection [16, 15, 17], and part of their success can be attributed to their use of object proposals. Before proposal methods, it was common for classifiers to exhaustively test $\sim 10^6$ sliding window locations [33, 12]. Object proposal methods [32, 5, 30] provide a more manageable set of regions, which in most cases is < 5K. Given a smaller set of regions, it becomes feasible to apply more complex classifiers, increasing accuracy. Recent experiments have also shown that using proposals can reduce false positives in a class-specific object detector like DPM [2].

Proposal generation methods either produce bounding boxes [36, 2, 6], or segments [9, 5, 32]. Recent work [16, 8] argues for latter by demonstrating that segmentation-based features significantly increase the mean accuracy on both segmentation and detection challenges in Pascal VOC [11]. Their experiments indicate that both object shape and context are useful for recognition.

Encouraging results for detection have recently spurred new proposal methods. Selective Search [32] is one of the more popular methods and is based on grouping. It performs hierarchical merging of superpixels with different metrics, producing a diverse set of proposals. Yanulevskaya [35] and Bonev *et al*. [4] improve Selective Search by guiding the hierarchical grouping process. Instead of grouping by various metrics, our method segments objects by finding global minima of an energy function defined on superpixels. This is similar to other methods performing maximum a posteriori (MAP) inference by graphcuts for proposal generation [5, 19, 9, 31, 24].

Recently, there have been some attempts to produce proposals by supervised learning. Krähenbühl and Koltun's LPO [24] generates regions from CRF models trained on VOC. We demonstrate better performance than LPO without training any models for proposal generation. Pinheiro *et al*. [29] introduced a CNN trained on COCO to generate segment proposals. POISE's segment boundaries appear to be qualitatively better than [29], which loses spatial accuracy due to the pooling layers. We refer the reader to Hosang *et al*. [18] for an excellent review of proposal methods. In §6 we evaluate several methods using the average recall metric which was introduced in their work.

One main contribution of this paper is the use of geodesically guided PMC to solve the *middle child problem*. Kolmogorov *et al*. [22] review PMC applications in vision. They demonstrate how PMCs can be used to solve some geometric functionals. Lim *et al*. [26] deal with more general constraints to produce accurate segments, when some ground-truth statistics are available. [25] discusses generating more solutions by decomposing the image. Certainly these methods could be useful for generating proposals, but they typically produce a segment in the order of seconds. Batra *et al*.'s work on Diverse M-Best [3] obtains highly probable solutions beyond MAP by Lagrangian relaxation in MRF models. This is related to our approach, since both methods change unary costs after obtaining the first optimal solution. On the other hand, exemplar-cut [34] changes energies to push solutions toward exemplars. Both these approaches [3, 34] adjust energies to direct solutions away

or towards existing solutions/exemplars, whereas we adjust the energy to encourage a more complete set of solutions.

3. The Middle Child Problem

This three part section defines and explains the middle child problem in PMC for segmentations. We start by introducing the PMC energy and the equivalent graph. In the second part, we illustrate why the problem exists using a simple model with 3 regions. We generalize this model in the third section, and show that the problem remains. Our example images are constructed from concentric regions which mimics the compositional nature of objects.

Generating Proposals by PMC: Our algorithm uses graph-cuts from multiple seeds to compute segments. For each seed, a directed graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ is created with nodes $\mathcal{V}$ and edges $\mathcal{E}$. Using this graph, we construct and minimize the Quadratic Pseudo-Boolean (QPB) function,

$$E(\mathbf{x}) = \sum_{i \in \mathcal{V}} \left(\theta_i^1 x_i + \theta_i^0 \overline{x}_i\right) + \sum_{(i,j) \in \mathcal{E}} \left(\psi_{ij}^{11} x_i x_j + \psi_{ij}^{01} \overline{x}_i x_j + \psi_{ij}^{10} x_i \overline{x}_j + \psi_{ij}^{00} \overline{x}_i \overline{x}_j\right) .$$

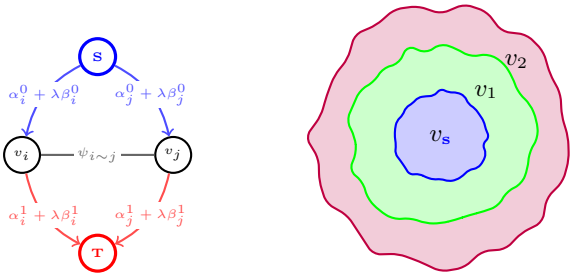
The solution is the boolean vector $\mathbf{x} = [x_1, \ldots, x_n]$. θ_i^ℓ is the unary potential associated with variable v_i when it takes the binary label ℓ. The pairwise potential, $\psi_{ij}^{\ell\eta}$, is used when variables v_i and v_j take binary labels ℓ and η respectively.

Since we use Potts energy, where $\psi_{ij}^{01} = \psi_{ij}^{10}$ (which we will denote as $\psi_{i \sim j}$), we can simplify the function to

$$E(\mathbf{x}) = \sum_{i \in \mathcal{V}} \left(\theta_i^1 x_i + \theta_i^0 \overline{x}_i\right) + \sum_{(i,j) \in \mathcal{E}} \left(\psi_{i \sim j} \left|x_i - x_j\right|\right) . \quad (1)$$

To generate object proposals, we convert this to the parametric pseudo-quadratic form, where $\theta_i^\ell = \alpha_i^\ell + \lambda \beta_i^\ell$, which can be represented as the graph given in Fig. 2(a). We denote the resulting parametric energy as $E_\lambda(\mathbf{x})$. The real-valued PMC parameter λ belongs to a sequence $\lambda_0 < \lambda_1 < \cdots < \lambda_L$. The unary potentials are defined by the values α_i^ℓ and β_i^ℓ. Given these parameters, the energy can be readily minimized by max-flow/min-cut. Min-cut produces two disjoint sets S and T, where node $v_i \in S$ iff $x_i = 1$, and $v_i \in T$ iff $x_i = 0$. The cut is defined by the sum of edge weights from S to T, which can be verified to equal the minimization of (1). We are interested in the monotonic case for PMC, where $\beta_i^1 < \beta_i^0$, which can be re-parameterized to get non-decreasing source capacities and non-increasing sink capacities with increasing λ [21]. The monotonic case gives solutions with the nesting property, where if $x_i = 1$ for λ_t, it is guaranteed that $x_i = 1$ for $\lambda_{t+1} > \lambda_t$ [22, 14].

We use PMC to produce multiple segments from each foreground seed at various image locations. For seed nodes, $v_{\mathbf{s}}$, we enforce $x_{\mathbf{s}} = 1$ by setting $\alpha_{\mathbf{s}}^0 = \infty$. All remaining nodes are $v_i \in \mathcal{V} \backslash \{v_{\mathbf{s}}\}$, each representing a superpixel. $\mathcal{E}$ is the set of all superpixel pairs which share a boundary.



(a) QPB function set up for graph-cut

(b) Example image to illustrate the middle child problem

Figure 2: (a) shows how a QPB function is represented as a graph, where a min-cut would minimize the function. (b) is an example image to demonstrate the middle child problem. In all graphs, **S** and **T** are the special source and sink nodes used by min-cut.

Simple Model with 3 Regions: To demonstrate the middle child problem, consider the image in Fig. 2(b) with three concentric regions. The center region, $v_{\mathbf{s}}$ is the seed. Following the formulation in [22], we set $\theta_i^0 = 0$. We assume that the unaries on all pixels are a constant, and set $\alpha_i^1 = C$ and $\beta_i^1 = -1$, where C is some constant. Translating unaries from a pixel to a superpixel graph incurs a constant multiplication factor of the size of the superpixel, z_i. The resulting unary potential is $\theta_i^1 = (C - \lambda) z_i$. This mimics a standard (uniform) graph used by CPMC [5], as well as RIGOR [19]. Suppose, $\psi_{s \sim 1}$ and $\psi_{1 \sim 2}$ are the costs associated to the outer boundaries of the blue and green regions respectively. Let us assume $z_{\mathbf{s}} < z_1 < z_2$ and $\psi_{s \sim 1} < \psi_{1 \sim 2}$ (longer boundaries typically have larger capacities).

We can compute the energy of each configuration of the vector $\mathbf{x}$. Since $\alpha_{\mathbf{s}}^0 = \infty$, $v_{\mathbf{s}}$ would always be in the foreground. We check the remaining four configurations:

		v_2	
		$x_2 = 0$	$x_2 = 1$
v_1	$x_1 = 0$	$E(\mathbf{x}) = \psi_{s \sim 1}$	$E(\mathbf{x}) = \psi_{s \sim 1} + \psi_{1 \sim 2} + (C - \lambda) z_2$
	$x_1 = 1$	$E(\mathbf{x}) = \psi_{1 \sim 2} + (C - \lambda) z_1$	$E(\mathbf{x}) = (C - \lambda)(z_1 + z_2)$

For simplicity, we will refer to the solution $x_1 = \ell, x_2 = \eta$ as $\langle \ell\eta \rangle$, and $\langle 10 \rangle$ is the middle child solution. When $\lambda \geq 0$, notice that the $\langle 01 \rangle$ solution will have higher energy than $\langle 10 \rangle$ because $z_2 > z_1$. Furthermore, when $\lambda = 0$, we will get the $\langle 00 \rangle$ solution, *i.e.* only $v_{\mathbf{s}}$ is in the foreground, as long as $\psi_{s \sim 1} < \psi_{1 \sim 2} + C z_1$ and $\psi_{s \sim 1} < (z_1 + z_2) C$. $\min E_\lambda(\mathbf{x}) = \psi_{s \sim 1}$ in this case. When $\lambda \geq C$, we will obtain the solution $\langle 11 \rangle$, *i.e.* all the regions are in the foreground, and $\min E_\lambda(\mathbf{x}) \leq 0$.

The key question is whether it is possible to obtain solution $\langle 10 \rangle$ from some real-valued λ? For this to be true, two conditions must hold for some λ:

1. $\psi_{1 \sim 2} + (C - \lambda) z_1 < \psi_{s \sim 1}$ 2. $\psi_{1 \sim 2} < (C - \lambda) z_2$

These two conditions imply that $E(\mathbf{x})$ for $\langle 10 \rangle$ should be

less than the energies of the $\langle 00 \rangle$ and $\langle 11 \rangle$ solutions at some λ. The first condition is discounted by our initial condition $\psi_{1\sim 2} > \psi_{s\sim 1}$, and will only be true if $C < \lambda$. The second condition can be true when $\lambda < C$, implying that we will never obtain the middle segment. In practice, one might obtain the segment in the middle if its boundaries have less total capacity than the boundaries it encloses, *i.e.* $\psi_{1\sim 2} < \psi_{s\sim 1}$. Since in a superpixel graph the image boundary/edge strength is inversely proportional to the pairwise potential $\psi_{i\sim j}$, this condition requires that a medium sized segment boundary must be stronger than its internal boundaries. This is not true in presence of strong internal structure (*e.g.* a striped shirt) in conjunction with weak object edges.

General PMC with $n+1$ Regions: We now demonstrate that the middle child problem also exists for graphs of more general form with $n+1$ regions (the seed, $v_{\mathbf{s}}$ contributes the $+1$), as illustrated in Fig. 3. We would use capacities from $\mathbf{S}$ and $\mathbf{T}$ as $e_i + \lambda f_i$ and $g_i - \lambda h_i$ respectively, as prescribed in Gallo *et al.* [14]. Here, e_i, f_i, g_i, h_i are all functions of vertex v_i, returning non-negative values. Moreover, $g_i \geq \lambda h_i$, $\forall \lambda$ to disallow negative capacities on sink arcs. We are interested in segments that form a single connected component, growing outward from $v_{\mathbf{s}}$. The aim is to produce all segments $\langle 1 \ldots 10 \ldots 0 \rangle$, where the last 1 happens at index t. This translates to the cut given in Fig. 3, which is equivalent to the whole region inside the solid green boundary belonging to v_t. In this section $\psi_t \equiv \psi_{t\sim t+1}$.

First, let us look at the energies of different solutions. For $\langle 0 \ldots 0 \rangle$, where only $v_{\mathbf{s}}$ is in the foreground,

$$E_\lambda(\mathbf{x}) = \psi_{s\sim 1} + \sum_{i=1}^{n} (e_i + \lambda f_i) \ . \tag{2}$$

For $\langle 1 \ldots 10 \ldots 0 \rangle$, where the cut passes through ψ_t, and $t \in \{1, \ldots, n-1\}$ (as illustrated in Fig. 3),

$$E_\lambda(\mathbf{x}) = \psi_t + \underbrace{\sum_{i=1}^{t} (g_i - \lambda h_i)}_{\text{Unaries cut from } \mathbf{T}} + \underbrace{\sum_{i=t+1}^{n} (e_i + \lambda f_i)}_{\text{Unaries cut from } \mathbf{S}} \ . \tag{3}$$

Similarly, for $\langle 1 \ldots 1 \rangle$, the full image solution is

$$E_\lambda(\mathbf{x}) = \sum_{i=1}^{n} (g_i - \lambda h_i) \ . \tag{4}$$

To get a middle segment (a segment enclosed by the solid green boundary), the following conditions need to hold:

1. Eq. 3 should be less than Eq. 2:
   $\psi_t + \sum_{i=1}^{t} (g_i - \lambda h_i) < \psi_{s\sim 1} + \sum_{i=1}^{t} (e_i + \lambda f_i)$
2. Eq. 3 should be less than energies of smaller segments, where $1 \leq k_1 < t$:
   $\psi_t + \sum_{i=k_1+1}^{t} (g_i - \lambda h_i) < \psi_{k_1} + \sum_{i=k_1+1}^{t} (e_i + \lambda f_i)$

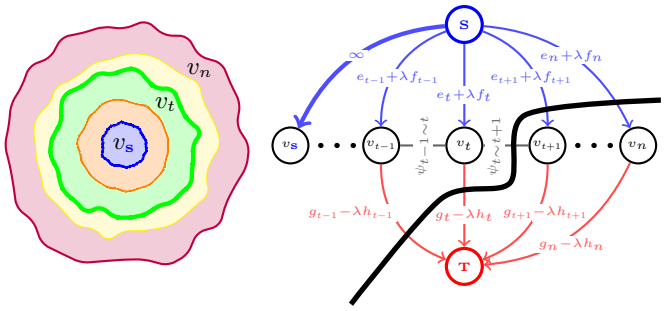


Figure 3: Generalization of Fig. 2(b) to multiple middle regions and unaries defined in Gallo *et al.* [14]. The image (left) and the corresponding graph (right) are given. The black curve on the graph shows the $\langle 1 \ldots 10 \ldots 0 \rangle$ cut, which is equivalent to the segment inside the thick green boundary on the left.

3. Eq. 3 should be less than energies of larger segments, where $t < k_2 \leq n$:
   $\psi_t + \sum_{i=t+1}^{k_2} (e_i + \lambda f_i) < \psi_{k_2} + \sum_{i=t+1}^{k_2} (g_i - \lambda h_i)$
4. Eq. 3 should be less than Eq. 4:
   $\psi_t + \sum_{i=t+1}^{n} (e_i + \lambda f_i) < \sum_{i=t+1}^{n} (g_i - \lambda h_i)$

We can think of v_{k_1} as a variable between $v_{\mathbf{s}}$ and v_t. For instance, this could be the variable associated with the orange region surrounding $v_{\mathbf{s}}$ in the Fig. 3 image. Similarly, we can think of v_{k_2} as the yellow region surrounding v_t. To make it easier to analyze these constraints, we introduce some new variables. The first set of variables is for the sum $\sum_{i=l_b}^{l_u} (e_i - g_i)$. The following illustration gives variable symbols, each surrounded by two lines. Each variable is for the sum, where l_b and l_u is defined by the labels on the surrounding two lines. For instance, $C = \sum_{i=t+1}^{k_2} (e_i - g_i)$:

$$\begin{array}{ccccccccc} | & & | & & | & & | & & | \\ 1 & A & k_1 \ k_1+1 & B & t \ t+1 & C & k_2 \ k_2+1 & D & n \\ | & & | & & | & & | & & | \end{array}$$

Since, both e_i and g_i are non-negative, all variables A, B, C, D possibly could be negative. The next set of four variables, M, N, P, Q are defined for the sum $\sum_{i=l_b}^{l_u} (f_i + h_i)$. They are defined over the same limits, *e.g.* $N = \sum_{i=k_1+1}^{t} (f_i + h_i)$. Note that these variables can only have non-negative values. Furthermore, for simplicity, we will use $\Psi = \psi_{t\sim t+1}$.

After some simple algebra, and replacing variables, we can convert the four constraints to:

1. $\frac{\Psi - \psi_{s\sim 1}}{M+N} - \frac{A+B}{M+N} < \lambda$
2. $\frac{\Psi - \psi_{k_1\sim k_1+1}}{N} - \frac{B}{N} < \lambda$
3. $\lambda < \frac{\psi_{k_2\sim k_2+1} - \Psi}{P} - \frac{C}{P}$
4. $\lambda < -\frac{\Psi}{P+Q} - \frac{C+D}{P+Q}$

Let us suppose we have no control over the pairwise potentials, and we can only adjust unaries so that λ has a feasible non-negative value that satisfies these constraints. One way to achieve this is to make the L.H.S. in the first two conditions negative, and the R.H.S. in the last two condi-

tions positive. Then, there would be some non-negative λ which will satisfy these constraints. Following this strategy, condition 1 requires $A + B > \Psi - \psi_{s\sim 1}$, and condition 2 requires $B > \Psi - \psi_{k_1 \sim k_1+1}$. Moreover, to have positive R.H.S in conditions 3 and 4, we require $C < \psi_{k_2 \sim k_2+1} - \Psi$ and $C + D < -\Psi$ respectively.

There are certain conclusions one can draw from this setup. Firstly, functions f_i and h_i have no influence over the chances of obtaining the middle segments. On the contrary, e_i and g_i are essential in obtaining any middle segments. To increase the chances to have a feasible λ, we need $e_i \gg g_i$ where $1 \leq i \leq t$, and $g_i \gg e_i$ where $t + 1 \leq i \leq n$. Of course, this cannot be simultaneously true for all $t \in \{1, \ldots, n-1\}$, hence it needs to be adjusted for each individual t. This observation vouches for the geodesics based solution we give in the next section.

4. Biasing PMC for Obtaining Medium Sized Segment Proposals

In the previous section, we identified a problem with the structure of standard graph-cut energies that results in missing medium-sized segments. We propose to solve this problem by biasing the solutions in a sequence of optimizations to obtain segments which are close to the last cut. These optimizations are performed on a fixed set of PMC parameters $\lambda_0 < \lambda_1 < \cdots < \lambda_L$. The parameter λ_l is used in minimizing $E_{\lambda_l}(\mathbf{x})$ to produce $\mathbf{x}_l = [x_1^{(l)}, \ldots, x_n^{(l)}]$. Given the solution $\mathbf{x}_l$, we want to set the unaries in way that minimizing $E_{\lambda_{l+1}}(\mathbf{x})$ produces only a slightly larger segment $\mathbf{x}_{l+1}$. This requires the energy of Eq. 4 to be larger than Eq. 3.

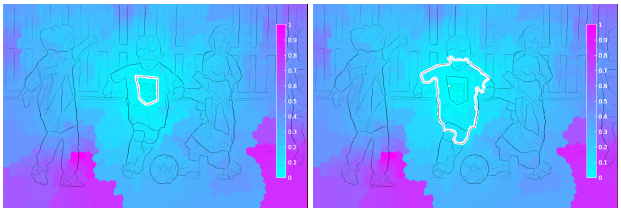
To enforce these constraints for obtaining segments $\mathbf{x}_{l+1}$ which are slightly larger than $\mathbf{x}_l$ (the last parametric solution), we change our unaries to the following form:

$$\alpha_i^\ell + \lambda_{l+1}\beta_i^\ell + f_i(\mathbf{x}_l) \tag{5}$$

This additional term $f_i(\mathbf{x}_l)$ guides the PMC to produce segments of all sizes. The function needs to be designed such that it raises the source unaries, θ_i^0, for superpixels which are spatially close to the last cut. Similarly we would like to raise the sink unaries, θ_i^1, for superpixels which are further away. Such a scheme would ensure that the energy of solutions that are slightly larger than the last cut decreases in comparison to segments which are much larger.

We find that the geodesic distance between superpixels is a good metric to guide our PMC. To construct $f_i(\mathbf{x}_l)$, we compute geodesics on an undirected graph with edge weights given by image edge strength - so two superpixels sharing a weak edge have a short geodesic distance. We precompute the $n \times n$ all-pairs shortest paths g_{ij}. In performing PMCs, for each variable we can retrieve the minimum shortest path to any superpixel in the last cut $\mathbf{x}_l$, i.e.

$$\phi_i(\mathbf{x}_l) = min_{j \in \mathcal{V} \,:\, x_j^{(l)}=1} g_{ij} \tag{6}$$



(a) $\mathbf{x}_l$ overlayed on $\phi_i(\mathbf{x}_{l-1})$ (b) $\mathbf{x}_{l+1}$ overlayed on $\phi_i(\mathbf{x}_l)$

Figure 4: This shows the geodesic distances $\phi_i(\mathbf{x}_*)$, which is used to bias unary potentials to produce medium sized segments. The resulting cut is overlayed on each figure as a white boundary. Note that all superpixels where $x_i^{(l)} = 1$ (cut in (a)), the next computed $\phi_i(\mathbf{x}_l) = 0$ (color in (b)), since now it is inside the previous cut.

This is visualized in Fig. 4 for two consecutive cuts. We finally compute $f_i(\mathbf{x}_l) = h(\phi_i(\mathbf{x}_l))$, where $h(\cdot)$ is a linear function, allowing us to raise source unaries if $\phi_i(\mathbf{x}_l) < \tau$, and raise sink unaries if $\phi_i(\mathbf{x}_l) > \tau$. We empirically tune the geodesic threshold, τ, on the VOC'12 training set.

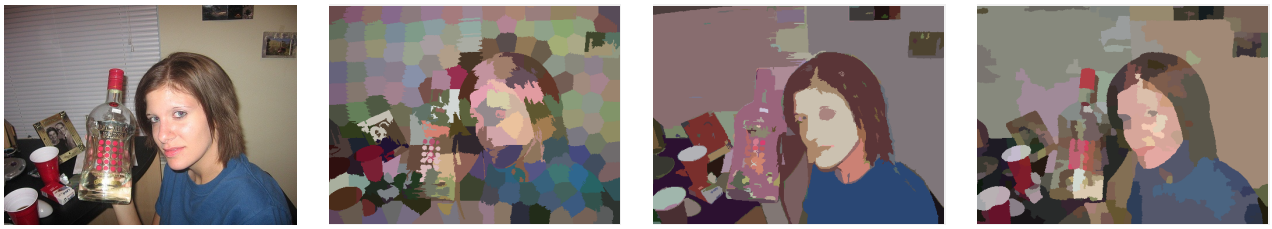
In our experiments, we noticed that the raw geodesic distance can be adversely affected by leaks in object boundaries. Ideally, one would like to compute the K shortest paths between any two superpixels, in order to avoid using erroneous boundaries. Since such a scheme would be expensive to compute, we resort to dropping 50% of the weakest edges in the superpixel graph before computing the geodesic distances. Since dropping the weakest edges can disconnect the graph, we avoid dropping edges in the graph which belong to a maximal spanning tree.

In practice, medium sized segments lie typically between 400 to 4,000 pixels. Our experiments demonstrate (Fig. 7) that *our solution is superior to all others in this regime*.

5. Segment Seeds from Merging Superpixels

Careful seed placement is important for good object proposal performance. In order to capture the majority of objects with a small number of proposals, it is preferable to place fewer seeds in regions with more uniform color and more seeds in regions that have more internal variation. Previously, seeds have been placed on all superpixels [10], a regular grid [5, 19], via diversified optimization [23], etc.

We propose seeding based on a hierarchical merging of watershed superpixels. Since watershed superpixels already combine areas with uniform color, it offers a nice starting point for obtaining different spatial resolutions in different areas. Our merging process is considerably faster than many optimization approaches, as only very simple operations are involved. In principle, any merging algorithm can be used, but we propose a new superpixel merging algorithm. The new algorithm is similar with the widely used Felzenszwalb-Huttenlocher (FH) algorithm [13], but with an adaptive thresholding scheme to improve the regularity of the superpixels in creating a hierarchy.



Original Image | SLIC Superpixels [1] | FH Superpixels [13] | Our Superpixels

Figure 5: Illustration of superpixel results. All algorithms produce 190 ± 1 superpixels (according to the default settings of FH [13]). SLIC [1] regularization is 0.05 (higher regularization would lose more detail). Superpixels are colored by their mean color plus a small random perturbation to reveal the differences among ones with similar colors. SLIC severely lacks detail by spending the budget evenly across the image. The FH algorithm produces many superpixels on very small textures and some superpixels are highly irregular in shape. Our merging method largely alleviated the problems of FH, and hence can represent more meaningful parts (e.g., the bottle cap, the mouth of the person) while preserving boundaries more effectively.

A basic idea, similar to FH, is that when generating superpixels of different sizes, smaller superpixels should be merged together unless they have very distinctive appearance. On the other hand, two large superpixels should not be merged when they have a moderate difference in appearance. We implement a novel adaptive thresholding scheme for this purpose. At each iteration, a "desired superpixel size" S_d is computed to set the adaptive threshold. S_d is initialized to $\frac{S}{N_d}$, where S is the number of pixels in the image and N_d is the user-specified desired number of superpixels. In subsequent iterations, S_d is chosen to satisfy:

$$S_d(N_d - \sum_{v_i} \mathbb{I}(|v_i| > S_d)) = S - \sum_{v_i, |v_i| > S_d} |v_i| \quad (7)$$

where $|v_i|$ represent the size of the superpixel v_i. In other words, S_d is equal to the average size of the remaining superpixels, after removing superpixels with sizes larger than S_d. This can be solved easily via an iterative procedure.

After obtaining the desired size, the adaptive threshold T_{ik} for superpixel v_i at iteration k is set to

$$T_{ik} = T_0 + kT_s \exp\left(-\sigma \frac{|v_i|}{S_d}\right) , \quad (8)$$

where T_0 is an initial threshold and T_s is the step size. $\sigma > 0$ is the parameter governing the tradeoff between large and small superpixels, so that T_{ik} is higher for smaller superpixels. The algorithm is not sensitive to T_0 and T_s which can be chosen simply to be sufficiently small. However, a larger T_s reduces computation time, hence is more desirable if there is no adverse impact on performance.

After obtaining the adaptive threshold, the edge and color distance between each connected superpixel pair are computed, and the pair is merged if both distances are smaller than the T_{ik} of the smaller superpixel in the pair. As the iteration advances, the threshold becomes larger and more small superpixels are merged since their relative penalty becomes larger after more iterations.

Within each iteration, we compute a merge graph M, with an edge on each superpixel pair that ought to be merged. This merge graph is complemented by the conflict graph C, which has an edge on each superpixel pair that are incident to each other but should not be merged. We start with the superpixel with the highest degree on M and proceed to iteratively merge all its neighbors without conflicts. If there are conflicts, we choose the one with the highest degree on M among the conflicting superpixels to merge.

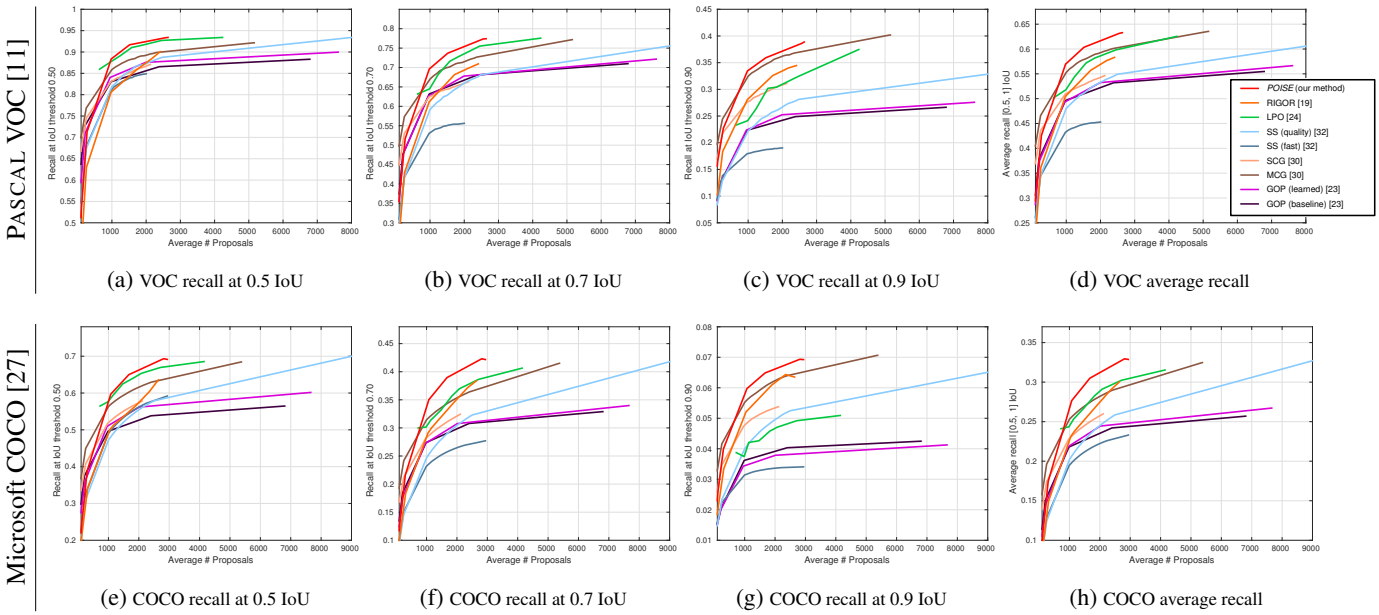
Most merging schemes have a clean-up routine for removing small superpixels. For our algorithm, every 5 iterations we run one "small superpixel merging" process, which is almost the same as normal merging, with the only difference being that the color difference from a large superpixel to a small one is only computed within a small vicinity of the latter. This is because the large superpixel might contain very distinct colors because of merging, and the mean color might have differed a lot from the smaller one. However if their colors are similar in the vicinity of the smaller superpixel, then the two should be merged.

We then generate one seed at the center of each merged superpixel, which has the capability of representing a complete picture of the scene with a moderate number of seeds.

6. Experiments

We conduct experiments on the validation sets of PASCAL VOC 2012 and Microsoft COCO [27]. Both have pixel-level annotations for certain object classes. There are $1,449$ images in VOC 2012, with $3,427$ ground-truth objects in 20 categories. COCO has $40,137$ images and $288,397$ ground-truth objects, with 80 categories that are currently available. Our algorithm is implemented in MATLAB with many crucial functions written in C++. We utilize StructEdges [7] for boundary detection and sticky superpixels [7] as nodes in the graph. Pairwise terms are computed from trained boosted regressors from RIGOR [19].

We report a number of metrics that have been widely used in previous evaluations. Suppose we want to evaluate



(a) VOC recall at 0.5 IoU (b) VOC recall at 0.7 IoU (c) VOC recall at 0.9 IoU (d) VOC average recall

(e) COCO recall at 0.5 IoU (f) COCO recall at 0.7 IoU (g) COCO recall at 0.9 IoU (h) COCO average recall

Figure 6: These graphs compare different object proposal methods based on recall against number of proposals at three IoU thresholds. For each segment ground-truth we select the proposal with the highest segmentation IoU. We use this to compute recall, which is the fraction of ground-truths having a corresponding proposal with an IoU score higher than the IoU threshold. [18] gives a similar comparison between methods for bounding box IoU. Note, the y-scale of each graph is different.

a segment pool $\mathbf{S} = \{S_1, \ldots, S_n\}$ against m ground-truth segments. First of all, each segment proposal $S_i \in \mathbf{S}$ is evaluated w.r.t. each ground-truth using the IoU overlap score

$$\text{IoU}(S_i, GT_j) = \frac{|S_i \cap GT_j|}{|S_i \cup GT_j|}, \qquad (9)$$

The best object overlap within the pool $\mathbf{S}$ is computed as

$$\text{IoU}(\mathbf{S}, GT_j) = \max_i \text{IoU}(S_i, GT_j)$$

We report the average best overlap (ABO), which is $\text{IoU}(\mathbf{S}, GT_j)$ averaged over all of the ground-truth objects in the dataset, as well as plotting recall under different IoU levels against the number of segments in the pool $|\mathbf{S}|$. In addition, we follow [18] in reporting the average recall under all IoU levels in $[0.5, 1]$. It is claimed that such an average recall measure correlates the best with downstream results on object detection [18]. Finally, we report the mean best covering over all images in the dataset:

$$\text{Cov}(\mathbf{S}, \mathbf{GT^I}) = \frac{\sum_j |GT_j| \text{IoU}(\mathbf{S}, GT_j)}{\sum_j |GT_j|}$$

where $\mathbf{GT^I}$ denotes all ground-truth objects in the same image. Covering measures the capability to extract larger segments and explain the scene as a whole.

We compare against recent methods SS (Selective Search) [32], SCG and MCG [30], GOP [23], RIGOR [19] as well as the very recent LPO approach [24].

Method	Recall at 0.70 IoU	Avg. # Proposals	ABO	Cov	Average Recall
$\sim$69.0% recall at IoU threshold 0.70					
GOP (learned) [23]	0.678	1,992	0.748	0.814	0.532
RIGOR [19]	0.682	1,715	0.752	0.840	0.557
SS (quality) [32]	0.681	2,482	0.757	0.828	0.549
LPO [24]	0.682	1,237	0.759	0.822	0.544
MCG [30]	0.692	1,291	**0.768**	0.835	**0.570**
POISE	0.696	**979**	**0.768**	**0.842**	0.569
Limit performance at IoU threshold 0.70					
GOP (learned) [23]	0.722	7,609	0.769	0.829	0.566
RIGOR [19]	0.709	**2,411**	0.777	0.844	0.583
SS (quality) [32]	0.772	10,641	0.801	0.840	0.618
LPO [24]	**0.776**	4,233	0.805	0.859	0.626
MCG [30]	0.772	5,157	0.808	0.850	**0.635**
POISE	0.774	2,639	**0.809**	**0.864**	0.633

Table 1: Detailed PASCAL VOC results of different algorithms. We consider two scenarios, the first is to generate $\sim$69.0% recall at an IoU threshold of 0.70, the second is the limit performance by allowing all the algorithms to generate the maximal amount of proposals. Our method, known as POISE is able to obtain the same performance with much fewer proposals than the competitors.

Table 1 shows detailed performance of different algorithms under two settings: one where all algorithms generate about 69.0% recall at an IoU threshold of 0.70; and the second where algorithms are allowed to generate maximal number of proposals. One can see that our method generates much fewer proposals in any of the two scenarios while having comparable performance to the best com-

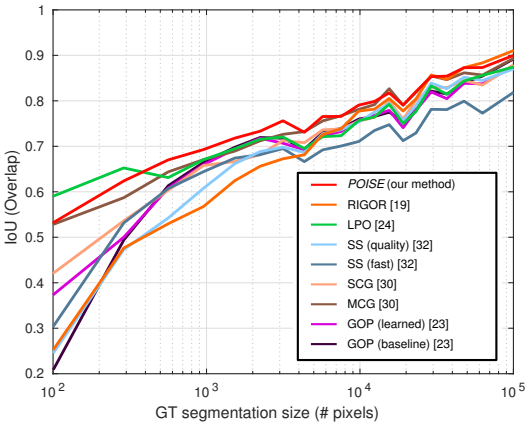


Figure 7: IoU comparison of various methods at different pixel sizes for Pascal VOC ground-truths, at $\sim 1,000$ # proposals.

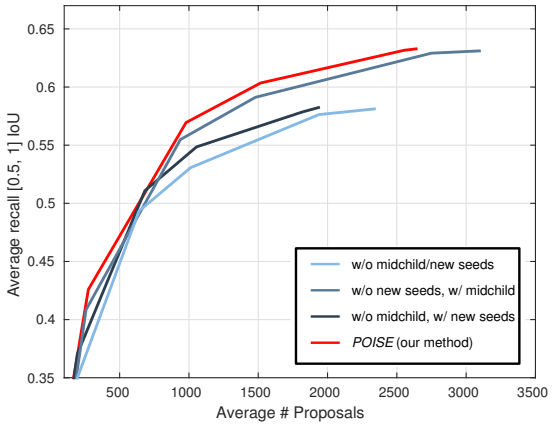


Figure 8: This graph shows results of an ablation study focused on average recall as in Fig. 6(d). We vary the number of seeds to generate results at different numbers of proposals.

petitors. Fig. 6(a)-6(d) shows the plots of segment recall at different overlap thresholds on the VOC dataset. Likewise, Fig. 6(e)-6(h) shows the results on the COCO dataset. We use linear instead of log scale [18, 30] to highlight that our method needs far fewer proposals to reach high recall. It can be seen that *our method consistently outperforms the competitors when the number of proposals is more than* 700, which is the range of settings most likely to be chosen users of proposal algorithms. POISE is superior to most other superpixel aggregation and edge-based approaches because it seeks solutions from a global energy function which solves the middle child problem.

Fig. 7 shows the IoU score broken down in terms of the size of the ground-truth segment. It can be seen that our method significantly outperforms all other approaches in objects with the sizes from 400 to $4,000$ pixels. This shows the effectiveness of our solution to the middle child problem, as well as the benefit of better seed placement. The only regime in which we are slightly worse than RIGOR is when the segment size grows to more than $60,000$ pixels, which is approaching the size of the entire image for a typical PASCAL VOC image. Even at that ground-truth size we still outperform all of the other competitors.

Ablation Study: Our paper has two contributions: a solution to the middle child problem; and a superpixel seeds generation method. In this section we will describe the results of an ablation study to identify the quantitative contribution of each of these two components. We compare four different variants of the algorithm: (1) "**w/o midchild/new seeds**" where neither the middle child solution in §4 or the new seeds in §5 are used; (2) "**w/o new seeds, w/ midchild**" where we use the geodesics middle child solution in §4, but not §5; (3) "**w/o midchild, w/ new seeds**" where we use the new seeds in §5, but not §4; and (4) the **POISE** method corresponding to the full algorithm in §4 and §5.

We generate results for all these variants individually over the complete validation set for PASCAL VOC 2012. The average recall results are plotted in Fig. 8. The general trend observable from these results is that the improved seeds (w/o midchild, w/ new seeds) help to move the graph left by reducing the number of proposals to reach the same recall. This is the result of requiring fewer number of seeds to localize most objects in the scene. On the other hand, the middle child solution (w/o new seeds, w/ midchild) moves the graph upward, indicating that adjusting the unaries by geodesics helps to obtain more accurate segmentations. Combining both improvements gives POISE the ability to increase recall while using fewer proposals.

7. Conclusion

In this paper we identify and solve the *middle child problem*— namely how to use parametric min-cuts to generate medium-sized segments for object proposals. We demonstrate that the problem arises from the intrinsic structure of the standard energy landscape and cannot be solved through parameter tuning. Our solution is an adaptive energy function which biases the min-cut solution in a sequence of proposals so that the next segment is close to the previous one from the standpoint of geodesic distance. In addition, we introduce a novel method for generating proposal seeds which is more effective than previous methods for small numbers of seeds. The resulting method, known as POISE (for "Proposals for Objects from Improved Seeds and Energies"), is demonstrated to outperform all competing methods in generating high-quality segments with a small proposal pool on the PASCAL VOC and Microsoft COCO datasets.

Acknowledgments: This work was supported in part by NSF grant IIS-1320348.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. SLIC Superpixels Compared to State-of-the-Art Superpixel Methods. *PAMI*, 34(11):2274–2282, Nov 2012. 6

[2] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the Objectness of Image Windows. *PAMI*, 34(11):2189–2202, Nov 2012. 2

[3] D. Batra, P. Yadollahpour, A. Guzman-Rivera, and G. Shakhnarovich. Diverse M-Best Solutions in Markov Random Fields. In *ECCV*, volume 7576, pages 1–16. 2012. 2

[4] B. Bonev and A. Yuille. A Fast and Simple Algorithm for Producing Candidate Regions. In *ECCV*, volume 8691, pages 535–549. 2014. 1, 2

[5] J. Carreira and C. Sminchisescu. CPMC: Automatic Object Segmentation Using Constrained Parametric Min-Cuts. *PAMI*, 34(7):1312–1328, 2012. 1, 2, 3, 5

[6] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr. BING: Binarized Normed Gradients for Objectness Estimation at 300fps. In *CVPR*, pages 3286–3293, June 2014. 2

[7] P. Dollár and C. L. Zitnick. Structured Forests for Fast Edge Detection. In *ICCV*, pages 1841–1848, Dec 2013. 2, 6

[8] J. Dong, Q. Chen, S. Yan, and A. Yuille. Towards Unified Object Detection and Semantic Segmentation. In *ECCV*, volume 8693, pages 299–314. 2014. 2

[9] I. Endres and D. Hoiem. Category Independent Object Proposals. In *ECCV*, volume 6315, pages 575–588. 2010. 1, 2

[10] I. Endres and D. Hoiem. Category-Independent Object Proposals with Diverse Ranking. *PAMI*, 36(2):222–234, Feb 2014. 1, 5

[11] M. Everingham, S. Eslami, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge: A Retrospective. *IJCV*, 111(1):98–136, 2015. 2, 7

[12] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object Detection with Discriminatively Trained Part-Based Models. *PAMI*, 32(9):1627–1645, sept. 2010. 2

[13] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient Graph-based Image Segmentation. *IJCV*, 59(2):167–181, 2004. 5, 6

[14] G. Gallo, M. D. Grigoriadis, and R. E. Tarjan. A Fast Parametric Maximum Flow Algorithm and Applications. *SIAM Journal on Computing*, 18(1):30–55, 1989. 3, 4

[15] R. Girshick. Fast R-CNN. In *ICCV*, Dec 2015. 2

[16] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *CVPR*, pages 580–587, June 2014. 1, 2

[17] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous Detection and Segmentation. In *ECCV*, volume 8695, pages 297–312. 2014. 2

[18] J. H. Hosang, R. Benenson, P. Dollár, and B. Schiele. What Makes for Effective Detection Proposals? *CoRR*, abs/1502.05082, 2015. 2, 7, 8

[19] A. Humayun, F. Li, and J. M. Rehg. RIGOR: Reusing Inference in Graph Cuts for Generating Object Regions. In *CVPR*, pages 336–343, 2014. 1, 2, 3, 5, 6, 7

[20] J. Kappes, B. Andres, F. A. Hamprecht, C. Schnörr, S. Nowozin, D. Batra, S. Kim, B. X. Kausler, T. Kröger, J. Lellmann, N. Komodakis, B. Savchynskyy, and C. Rother. A Comparative Study of Modern Inference Techniques for Structured Discrete Energy Minimization Problems. *IJCV*, pages 1–30, 2015. 1

[21] P. Kohli and P. H. Torr. Dynamic Graph Cuts for Efficient Inference in Markov Random Fields. *PAMI*, 29(12):2079–2088, 2007. 3

[22] V. Kolmogorov, Y. Boykov, and C. Rother. Applications of parametric maxflow in computer vision. In *ICCV*, 2007. 1, 2, 3

[23] P. Krähenbühl and V. Koltun. Geodesic Object Proposals. In *ECCV*, volume 8693, pages 725–739. 2014. 1, 5, 7

[24] P. Krähenbühl and V. Koltun. Learning to Propose Objects. In *CVPR*, June 2015. 1, 2, 7

[25] Y. Lim, K. Jung, and P. Kohli. Energy Minimization Under Constraints on Label Counts. In *ECCV*, pages 535–551, 2010. 2

[26] Y. Lim, K. Jung, and P. Kohli. Efficient Energy Minimization for Enforcing Label Statistics. *PAMI*, 36(9):1893–1899, Sept 2014. 2

[27] T. Lin, M. Maire, S. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common Objects in Context. *CoRR*, abs/1405.0312, 2014. 2, 6, 7

[28] T. Malisiewicz and A. A. Efros. Improving Spatial Support for Objects via Multiple Segmentations. In *BMVC*, pages 55.1–55.10. BMVA Press, 2007. 1

[29] P. Pinheiro, R. Collobert, and P. Dollár. Learning to Segment Object Candidates. In *NIPS*, 2015. 2

[30] J. Pont-Tuset, P. Arbeláez, J. T. Barron, F. Marqués, and J. Malik. Multiscale Combinatorial Grouping for Image Segmentation and Object Proposal Generation. *CoRR*, abs/1503.00848, 2015. 1, 2, 7, 8

[31] P. Rantalankila, J. Kannala, and E. Rahtu. Generating Object Segmentation Proposals Using Global and Local Search. In *CVPR*, pages 2417–2424, June 2014. 1, 2

[32] J. Uijlings, K. Sande, T. Gevers, and A. Smeulders. Selective Search for Object Recognition. *IJCV*, 104(2):154–171, 2013. 1, 2, 7

[33] P. Viola and M. J. Jones. Robust Real-Time Face Detection. *IJCV*, 57(2):137–154, 2004. 2

[34] J. Yang, Y.-H. Tsai, and M.-H. Yang. Exemplar Cut. In *ICCV*, pages 857–864, Dec 2013. 2

[35] V. Yanulevskaya, J. Uijlings, and N. Sebe. Learning to Group Objects. In *CVPR*, pages 3134–3141, June 2014. 2

[36] C. Zitnick and P. Dollár. Edge Boxes: Locating Object Proposals from Edges. In *ECCV*, pages 391–405. 2014. 2