Learning to Propose Objects

Philipp Krähenbühl
UC Berkeley

Vladlen Koltun
Intel Labs

Abstract

We present an approach for highly accurate bottom-up object segmentation. Given an image, the approach rapidly generates a set of regions that delineate candidate objects in the image. The key idea is to train an ensemble of figure-ground segmentation models. The ensemble is trained jointly, enabling individual models to specialize and complement each other. We reduce ensemble training to a sequence of uncapacitated facility location problems and show that highly accurate segmentation ensembles can be trained by combinatorial optimization. The training procedure jointly optimizes the size of the ensemble, its composition, and the parameters of incorporated models, all for the same objective. The ensembles operate on elementary image features, enabling rapid image analysis. Extensive experiments demonstrate that the presented approach outperforms prior object proposal algorithms by a significant margin, while having the lowest running time. The trained ensembles generalize across datasets, indicating that the presented approach is capable of learning a generally applicable model of bottom-up segmentation.

1. Introduction

Object proposal algorithms aim to identify a small set of regions such that each object in the image is approximately delineated by at least one proposed region. Object proposals can be computed bottom-up, based only on low-level boundary detection and category-independent grouping [7, 12, 32]. They are used as a starting point for both object detection and semantic segmentation, and have become a standard first step in state-of-the-art image analysis pipelines [5, 6, 16, 17, 32].

To support diverse image parsing tasks, object proposal algorithms must have a number of characteristics. They need to provide region proposals with informative shape for semantic segmentation and instance segmentation [5, 6, 16, 17, 23]. They must have high recall, producing corresponding regions for as many genuine objects as possible. They must generate a manageable number of proposals to limit unnecessary workload. And they must be fast to support high-performance image parsing [16, 32].

Figure 1: Object proposals for three images from the Microsoft COCO dataset. From left to right: input images, ground-truth instance segmentations, region proposals generated by the presented approach. Note the accurate instance proposals in the top and middle rows, despite color and texture similarity across instances. In the bottom row, the trained ensemble correctly identifies the white surfboard as a single object with three connected components.

In this paper, we present an object proposal algorithm that has all of these characteristics. The key idea is to optimize an ensemble of figure-ground segmentation models. Given a new image, the algorithm simply applies each model and outputs all of the produced foreground segments. The algorithm is fast since each model is highly efficient and operates on elementary image features. Proposals produced by a trained ensemble are shown in Figure 1.

A number of prior object proposal techniques can be viewed as ensembles of binary segmentation models [7, 12, 19]. However, in each case all models used the same potentials and differed only in one or two hyperparameters, which were varied according to a predefined schedule. In some cases, diversity was achieved at test time by means of a computationally expensive classifier that was used to rank the proposals [7, 12].

In contrast, the presented approach optimizes a diverse ensemble of segmentation models globally during training. The training objective is the accuracy of the generated proposal set balanced by its size. We show that the training

objective can be expressed in terms of the uncapacitated facility location problem and optimized by combinatorial techniques. The training jointly optimizes the size of the ensemble, its composition, and the parameters of the incorporated models, all for the same objective. The number of generated proposals can be controlled at training time and there is no need for test-time ranking.

We conduct extensive experiments on the Pascal VOC2012 dataset and the recent Microsoft COCO dataset, comparing the performance of the presented approach to state-of-the-art object proposal algorithms. We evaluate both region proposal accuracy and bounding box proposal accuracy. In region proposal accuracy, our approach outperforms prior methods by a wide margin, while having the lowest running time. For example, the approach achieves 94% recall on the VOC 2012 dataset as measured by detailed shape overlap: the highest ever reported. Our approach also yields the highest bounding box proposal accuracy simply by taking the bounding boxes of the proposed regions. We also demonstrate that the segmentation ensembles trained by our approach generalize across datasets. This suggests that the presented approach is capable of learning a generally applicable model of accurate bottom-up object segmentation.

2. Related Work

Object detection pipelines based on sliding windows became widespread following the work of Viola and Jones [9, 14, 33]. Since performing detailed classification on all candidate windows induces unnecessary computational costs, a number of approaches have been developed to prune and rank rectangular windows, thus allocating the computational budget to the most promising candidates [1, 8, 22, 35]. The recent ranking method of Zitnick and Dollár demonstrates both high efficiency and high recall [35]. Unlike these works, we focus on generating region proposals with detailed shape cues, in order to support diverse image analysis tasks including semantic segmentation and instance segmentation [5, 6, 16, 17, 23]. Although not the primary focus of our work, simply taking the bounding boxes of the regions identified by our model yields state-of-the-art results in bounding box proposal generation.

The use of bottom-up segmentation to generate candidate regions for object detection was advocated by Malisiewicz and Efros [25, 26], who obtained a pool of candidate regions by applying multiple segmentation algorithms with varying parameters, collecting the resulting segments, and adding regions obtained by merging adjacent segments. This built on the work of Russell et al. [29], who used a similar approach for unsupervised object discovery in image collections. Using multiple segmentations and grouping adjacent segments have become common ingredients in subsequent proposal algorithms [2, 3, 27, 32].

An alternative approach to region candidate generation is to compute many figure-ground segmentations and add each computed foreground region to the candidate set [7, 12, 19, 21]. Proposals are generated by applying a specified set of segmentation models to different locations in the image. The recent algorithm of Krähenbühl and Koltun achieves state-of-the-art accuracy using this approach [21]. Our method also uses figure-ground segmentations, but the proposals are generated by a diverse ensemble that comprises multiple model types. The size and composition of the ensemble are optimized during training to maximize the accuracy of the candidate set relative to its size.

3. Model

Our approach optimizes an ensemble $\mathcal{M} = \{M_1, \ldots, M_K\}$ of binary segmentation models. We primarily use two types of models. The first type is a global CRF that produces a single segmentation for a given image. The second type is a localized CRF that takes a given image location into account. The specified location serves as an optional attention mechanism. The application of a trained ensemble to an image is illustrated in Figure 2.

Let τ_k be the type of model M_k (e.g, global or localized) and let θ_k be its parameter vector. Let $\mathbf{X}_k^i$ be the set of proposals produced by M_k for image I^i. If M_k is a global CRF, $|\mathbf{X}_k^i| = 1$. For a localized CRF, the number of proposals equals the number of specified locations in image I^i.

Both types of models operate on a superpixel segmentation of a given image. Each CRF M_k parameterizes a probability distribution over binary partitions of the locally connected superpixel graph $(\mathcal{V}, \mathcal{E})$. The Gibbs energy of a partition $\mathbf{x} \in \{0,1\}^n$ is

$$E(\mathbf{x}|I^i;\theta_k) = \sum_{u\in\mathcal{V}} \psi_u(x_u|I^i;\theta_k) + \sum_{(u,v)\in\mathcal{E}} \psi_{u,v}(x_u,x_v|I^i;\theta_k),$$

where ψ_u and $\psi_{u,v}$ are unary and pairwise potentials, respectively.

We use binary log-linear CRFs with submodular pairwise potentials. Submodular binary CRFs can efficiently generate proposals via exact maximum a posteriori (MAP) inference [4]. The log-linear structure enables parameter estimation via large-margin learning [31].

Global CRF. For a global CRF model M_k, each unary potential has the form

$$\psi_u(x_u|I;\theta_k) = 1_{[x_u]} f_u^\top \theta_k,$$

where f_u is the unary feature vector evaluated at superpixel u. The pairwise terms are

$$\psi_{u,v}(x_u,x_v|I;\theta_k) = 1_{[x_u \neq x_v]} f_{u,v}^\top \theta_k,$$

where $f_{u,v}$ is the pairwise feature vector evaluated at the edge (u,v). All pairwise features are strictly positive and

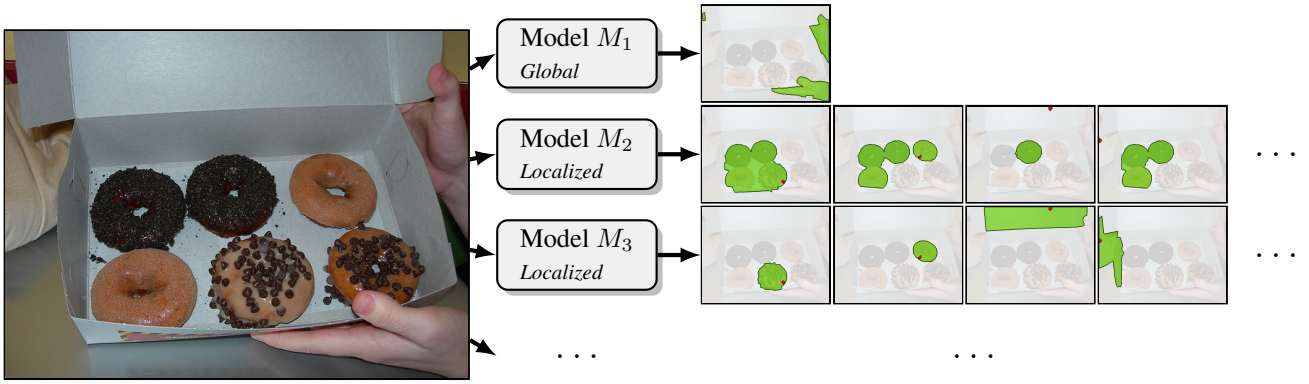


Figure 2: Our approach jointly trains an ensemble of binary segmentation models. Models of different type produce a different number of region proposals. At test time, the algorithm simply applies all models in the ensemble to a given image and collects the resulting proposals. The training procedure jointly optimizes the size and composition of the ensemble and the parameters of all models.

the pairwise parameters in θ_k are constrained to be non-negative during training to guarantee submodularity. The features are described in Section 5. This model produces a single proposal for a given image: $\mathbf{X}_k^i = \{\mathbf{x}_k^i\}$, where

$$\mathbf{x}_k^i = \arg\min_{\mathbf{x}} E(\mathbf{x}|I^i;\theta_k).$$

The proposal $\mathbf{x}_k^i$ can vary dramatically as a function of the parameter vector θ_k. We train an ensemble that incorporates multiple global models. The models share the same feature vector, thus amortizing feature computation. Different parameter vectors are jointly optimized during training to maximize the performance of the ensemble.

Global CRFs effectively identify distinct objects with characteristic global features, at the cost of only a single proposal per model. During training, different global models can specialize in different commonly occurring object appearances. However, global features are generally not sufficiently expressive to precisely delineate smaller and less salient objects, and do not effectively distinguish multiple instances with similar appearance. For these reasons, we also use localized models.

Localized CRF. Localized models have the same form as the global CRFs, but their unary feature vector $f_u(s)$ incorporates features that are defined in terms of a seed superpixel $s \in \mathcal{V}$. The seed serves as a locus of attention. The vector $f_u(s)$ includes all features used by the global models plus simple features that summarize the distance between u and s in the superpixel graph. The distance features and the distribution of seeds in an image are described in Section 5. Note that the localized models are not constrained by any hardcoded dependence on the seeds, in contrast to prior work that interpreted seed locations as hard constraints and generated regions that enclosed the seeds [7, 12, 21]. Our training procedure makes the seed distance features available to the localized models alongside the global features. The distance features can be utilized by different models in different ways. For example, we have observed localized models that specialize in delineating objects that lie away from the given seed.

A localized model produces $|\mathbf{X}_k^i| = |S^i|$ proposals for a given image I^i, where S^i is the set of seeds:

$$\begin{aligned}\mathbf{X}_k^i &= \left\{\mathbf{x}_k^i(s) : s \in S^i\right\}, \\ \mathbf{x}_k^i(s) &= \arg\min_{\mathbf{x}} E(\mathbf{x}|I^i, s;\theta_k).\end{aligned}$$

One of the benefits of localized CRFs is shift-invariance: a model can specialize in a type of object appearance and rely on the seeds to point out individual instances of this appearance.

4. Training

Let $\mathbf{O} = \{O^1, \ldots, O^N\}$ be a set of ground-truth objects. Object O^i is represented as a binary mask over image I^i. For convenience of exposition, assume that each image contains a single ground-truth object. If a dataset image contains multiple objects, it is replicated accordingly.

Consider a candidate model M_k. The loss of this model on an image I^i is defined in terms of the minimal Jaccard distance between object O^i and the set of proposals $\mathbf{X}_k^i$:

$$\Delta(O^i, \mathbf{X}_k^i) = \min_{\mathbf{x} \in \mathbf{X}_k^i} \left(1 - \frac{|O^i \cap \mathbf{x}|}{|O^i \cup \mathbf{x}|}\right).$$

Given an ensemble $\mathcal{M} = \{M_1, \ldots, M_K\}$, the loss of $\mathcal{M}$ on object O^i is defined as

$$\min_{k \in \{1,\ldots,K\}} \Delta(O^i, \mathbf{X}_k^i).$$

Thus the loss of an ensemble is the loss of the most accurate proposal. Our training objective minimizes this loss over all training examples, balanced by the number of proposals:

$$\underset{\mathcal{M}}{\text{minimize}} \sum_i \min_{M_k \in \mathcal{M}} \Delta(O^i, \mathbf{X}_k^i) + \lambda \sum_{M_k \in \mathcal{M}} |\mathbf{X}_k|, \quad (1)$$

where $|\mathbf{X}_k|$ is the total number of distinct proposals generated by model M_k for images in the training set. The first term in the objective minimizes the Jaccard distance between the proposal set and the ground truth objects. The second term penalizes the size of the proposal set. The parameter λ balances the two objectives. A small value of λ yields ensembles that generate large proposal sets, while setting $\lambda \to \infty$ yields a model that produces a single proposal.

Objective 1 optimizes over the set $\mathcal{M}$. The size and composition of this set is optimized along with the parameter vectors. This objective is not easily amenable to latent-variable training methods such as expectation maximization, which optimize parameters but not the structure of the model [34]. To optimize the complete objective globally, we introduce a different approach.

Facility location. Our approach reduces the training to a sequence of combinatorial optimization problems. Specifically, assume that we have generated a superset $\mathcal{U}$ of potential models and that the ensemble $\mathcal{M}$ is drawn from this candidate set:

$$\underset{\mathcal{M}\subseteq\mathcal{U}}{\text{minimize}} \quad \sum_i \min_{M_k\in\mathcal{M}} \Delta(O^i, \mathbf{X}_k^i) + \lambda \sum_{M_k\in\mathcal{M}} |\mathbf{X}_k|. \qquad (2)$$

This is an instance of the uncapacitated facility location problem (UFLP) [30]. The UFLP formulation concerns a set of facilities F and a set of customers C. For each facility $k \in F$, the cost of opening this facility is $f_k \in \mathbf{R}^+$. For each facility $k \in F$ and customer $i \in C$, the cost of serving customer i by facility k is c_{ki}. The problem is to open a subset of the facilities and assign each customer to an open facility such that the total cost is minimized:

$$\underset{Y\subseteq F}{\text{minimize}} \quad \sum_i \min_{k\in Y} c_{ki} + \sum_{k\in Y} f_k. \qquad (3)$$

Objective 2 is a UFLP with service costs $c_{ki} = \Delta(O^i, \mathbf{X}_k^i)$ and facility costs $f_k = \lambda|\mathbf{X}_k|$.

While uncapacitated facility location is NP-hard, it is of great practical interest and has been extensively studied. A number of algorithms are known to perform extremely well, approaching the exact solution on benchmark problems within a fraction of a percent [28]. Note that the metric variant of UFLP has approximation algorithms with very strong approximation guarantees [20, 24]. While the Jaccard distance is a metric [10], objective 2 is not a metric UFLP. Nevertheless, the approximation algorithms themselves [20, 24] are known to have very good experimental performance even in the general case [28]. Our implementation optimizes objective 2 using three algorithms [20,24,28] and selects the lowest-cost solution.

Candidate generation. Objective 1 is optimized by solving a sequence of facility location problems on adaptively generated candidate sets $\{\mathcal{U}_1, \ldots, \mathcal{U}_T\}$. To generate an initial candidate set $\mathcal{U}_1$, we sample a subset $\mathbf{S}_1 \subseteq \mathbf{O}$ of the training data uniformly at random. A distinct CRF is optimized for each sampled training example using large-margin learning [31]. The CRF type, global or localized, is selected uniformly at random for each training example. Let Γ_1 denote this set of models. In this first iteration, we set $\mathcal{U}_1 = \Gamma_1$. We then optimize objective 2 with the candidate set $\mathcal{U}_1$ to obtain an ensemble $\mathcal{M}_1$.

The training proceeds iteratively. In iteration t, we sample a subset $\mathbf{S}_t \subseteq \mathbf{O}$ uniformly at random. A new set of models Γ_t is optimized by fitting a distinct CRF to each sampled object. The model type is again selected at random for each new CRF. We also retrain each model M_k from the ensemble $\mathcal{M}_{t-1}$ on the training examples $\hat{\mathbf{O}}_k \subset \mathbf{O}$ that are best fit by this model:

$$\hat{\mathbf{O}}_k = \left\{O^i : \forall l \neq k.\ \Delta(O^i, \mathbf{X}_k^i) \leq \Delta(O^i, \mathbf{X}_l^i)\right\}.$$

This is an EM-style step akin to structured latent variable training [34]. However, rather than replace $\mathcal{M}_{t-1}$ with the retrained models, we add the set $\mathcal{M}'_{t-1}$ of retrained models to the candidate set $\mathcal{U}_t$, along with the new models Γ_t and the previous ensemble: $\mathcal{U}_t = \mathcal{M}_{t-1} \cup \mathcal{M}'_{t-1} \cup \Gamma_t$. Objective 2 is then optimized with the candidate set $\mathcal{U}_t$. This yields an ensemble $\mathcal{M}_t$. The procedure is iterated until the final ensemble $\mathcal{M}_T$ is produced. We use $T = 10$ in all experiments. The algorithm is summarized in Algorithm 1.

Algorithm 1: Ensemble training

$\mathcal{M}_0 := \emptyset$;
for $t = 1 \ldots T$ **do**
  $\mathbf{S}_t :=$ new training examples;
  $\Gamma_t :=$ new models optimized for $\mathbf{S}_t$;
  $\mathcal{M}'_{t-1} :=$ reoptimized models from $\mathcal{M}_{t-1}$;
  $\mathcal{U}_t := \mathcal{M}_{t-1} \cup \mathcal{M}'_{t-1} \cup \Gamma_t$;
  $\mathcal{M}_t := \text{UFLP}(\mathcal{U}_t)$;
end
Return $\mathcal{M}_T$;

5. Implementation

We use the superpixel segmentation of Krähenbühl and Koltun [21], which is based on the boundary detection algorithm of Dollár and Zitnick [11]. The boundary detection and superpixel segmentation provides a weighted superpixel graph, where the weight $w_{u,v}$ indicates the boundary strength between adjacent superpixels u and v. The global unary feature vector f_u has 18 elements. We use nine RGB color features: average color of superpixel u, average color

of the entire image, and the element-wise squared difference between the two. We also use five position features: The center of mass (x, y) of superpixel u normalized to $[-1, 1]$, as well as x^2, y^2, and xy. Finally, we add four boundary distance features, using the geodesic distance of u from the image boundary on the superpixel graph $(\mathcal{V}, \mathcal{E})$, with each edge $(i, j) \in \mathcal{E}$ reweighted by $w^{\alpha}_{i,j}$ for $\alpha = 0, 1, 2, 3$. Note that our features are elementary: we rely on the learning algorithm to find good parameter sets that utilize these simple features as needed. The upshot is fast proposal generation unencumbered by expensive feature evaluation.

For the localized models we add four additional elements, which summarize the distance between u and a seed superpixel s. We use the geodesic distance between u and s with the same four sets of edge weights.

The pairwise feature vector $f_{u,v}$ has five elements: $\exp(-\beta w_{u,v})$ for $\beta = 0, 1, 2, 3, 4$. The exponent ensures that the pairwise features are positive and the pairwise potentials are submodular.

We train three types of localized models on three seed distributions. Seeds are distributed using the seed placement model of Krähenbühl and Koltun [21]. To train the seed placement models, we partition the set of objects in the Pascal VOC 2012 training set by size into the largest third, the medium third, and the smallest third. Three seed placement models are trained separately on these sets. The placement models distribute on average 15, 70, and 200 seeds per image, respectively.

Empty proposals are filtered out trivially. Near-duplicate proposals are filtered out using the fast duplicate detection of Krähenbühl and Koltun [21].

5.1. Small objects

The model types described so far – the global CRF and the localized CRFs – operate on the same superpixel segmentation. This enables rapid feature computation and inference, but the quantization of the image domain has a cost. Any partition at the superpixel level will perform poorly for objects that are roughly the size of a single superpixel or smaller and do not align well with superpixel boundaries. This is particularly relevant for the Microsoft COCO dataset [23], where 33% of the annotated objects have an area of 25×25 pixels or less. On this part of the dataset, any proposals based on our superpixel segmentation cannot achieve an average best overlap (ABO) above 45%. This is in contrast to the remainder of the dataset, on which the superpixel segmentation limits the highest achievable ABO to 90%.

The presented ensemble training approach easily accommodates additional model types. We add a model type that specifically targets small objects. This model oversegments the image using the algorithm of Felzenszwalb and Huttenlocher [15] and proposes all segments smaller than 1000 pixels. The model has two parameters: the color space (Lab or HSV) and a minimum internal difference parameter used by the Felzenszwalb-Huttenlocher algorithm. During training, this model type is simply sampled alongside the others when a candidate model is generated. The parameters of this model type are sampled uniformly at random. Since the training procedure is completely general, it requires no modifications. Advantageous small-object models are chosen automatically if including them in the model set improves Objective 2.

6. Results

We evaluate the presented approach on the PASCAL VOC2012 dataset [13] and the Microsoft COCO dataset [23]. For the PASCAL VOC2012 dataset, we train on all segment annotations in the training set (1464 images, 3507 segmented objects) and evaluate on the validation set (1449 images, 3422 segmented objects). Bounding box proposal accuracy is evaluated on the larger detection dataset (5823 images, 13841 bounding boxes). The Microsoft COCO dataset is much larger, with 82783 training images and 40504 validation images. We train on a subset of 8000 training images with 62135 segmented objects and evaluate on the complete validation set with 296492 segmented objects. All experiments were performed on an Intel Core i7-3770K processor clocked at 3.5 GHz. Runtimes for all methods are reported for single-threaded execution and cover all operations, including boundary detection and oversegmentation.

To evaluate the quality of our object proposals we use the Average Best Overlap (ABO) and α-recall measures [7, 21]. The ABO between a ground truth object set $\mathbf{O} = \{O^1, \ldots, O^N\}$ and a set of proposals $\mathbf{X}$ is computed using the overlap between each ground truth region $O^i \in \mathbf{O}$ and the closest object proposal $\mathbf{x} \in \mathbf{X}$:

$$\text{ABO} = \frac{1}{|\mathbf{O}|} \sum_{O^i \in \mathbf{O}} \max_{\mathbf{x} \in \mathbf{X}} \mathcal{J}(O^i, \mathbf{x}).$$

Here $\mathcal{J}$ is the Jaccard coefficient: $\mathcal{J}(O^i, \mathbf{x}) = \frac{|O^i \cap \mathbf{x}|}{|O^i \cup \mathbf{x}|}$. The α-recall of $\mathbf{X}$ is the fraction of segments O^i in $\mathbf{O}$ for which $\max_{\mathbf{x} \in \mathbf{X}} \mathcal{J}(O^i, \mathbf{x}) \geq \alpha$.

We first evaluate the different components of our training procedure and then present a set of comparisons to prior work.

Training procedure. To evaluate different components of our training procedure we use different variants of the procedure to train an ensemble with roughly 2200 proposals on the VOC 2012 dataset. First, we only use the initial stage of the procedure, in which random training examples are sampled and models are optimized for individual examples. We optimize for the composition of the ensemble using a single

Training procedure	ABO	p-value
Initial stage	0.749	–
EM	0.777	<0.01
EM + UFLP	0.781	<0.01
EM + New models + UFLP	0.785	<0.01

Table 1: Evaluation of different components of the training procedure, using our full model with $\lambda = 0.03$ on the VOC 2012 test set (roughly 2200 proposals). The p-value in each row measures the statistical significance of improvement over the prior row.

round of UFLP. Second, we add T iterations of retraining, in which each model is retrained on the objects best fit by this model. This is an EM-style structured latent variable training procedure [34], initialized by UFLP. The third variant adds the combinatorial (UFLP) optimization to each iteration. The fourth variant is the complete procedure, which injects new models trained on randomly sampled examples in each iteration. The results are reported in Table 1. Each component of the procedure improves the performance of the trained ensemble with strong statistical significance.

6.1. VOC 2012 region accuracy

We now evaluate the accuracy of the region proposals produced by our approach on the VOC 2012 dataset. The results are reported in Table 3. We compare the presented approach to six state-of-the-art object proposal algorithms. The parameter λ is set to several different values to match the different numbers of proposals produced by each prior approach, as shown in Table 3. For each method we measure the ABO, 50%-recall, 70%-recall, and the p-value computed using Student's t-test. The t-test measures the statistical significance with which our approach outperforms each competing method. Each ground truth object is treated as an independent observation. For each object and each competing method, the test evaluates whether the set of proposals produced by our approach has lower or equal overlap with this object than the set of proposals produced by the competing method.

Our approach outperforms all prior methods with strong statistical significance ($p < 0.01$), except MCG [3] for which the results are not statistically significant. Our approach also has the lowest running time for all proposal set sizes. See Table 3 for details.

For the first tier of proposal set sizes (roughly 650 proposals), our approach has the highest ABO. For the second tier (1000-1600 proposals), our approach has the highest ABO and outperforms the closest prior method (GOP) by 2 percentage points. For higher tiers (above 2000 proposals), our approach has an ABO of 3 to 8 percentage points higher than all prior approaches except MCG. Note that the recall measures for our approach are consistently higher than for MCG and that our algorithm is more than an order of magnitude faster.

Models	# prop.	% best	$\sqrt{\text{med. area}}$	time
Global	214	8.0	201	0.05s
Localized (l)	1514	27.8	171	0.34s
Localized (m)	1357	23.5	116	0.31s
Localized (s)	2846	48.8	89	0.56s
Small objects	1378	8.6	14	0.16s
All	7309	100.0	115	1.43s

Table 2: Composition of an ensemble trained on the VOC 2012 dataset with $\lambda = 0.01$. The ensemble comprises global CRFs, localized CRFs (small, medium, and large), and small object proposals. For each model type we report the number of proposals produced by models of the given type (before near-duplicate and empty proposal removal), percentage of objects best fit by models of the given type, median area of proposals, and running time. The percentages sum up to more than 100% because some of the objects are fit equally well by multiple models.

The running time of our approach includes 0.5s for boundary detection and superpixel segmentation. The remainder of the running time is divided almost equally into feature computation, multiplication of feature and parameter vectors, energy minimization via graph cuts, and near-duplicate removal.

Table 2 shows the composition of a complete ensemble, trained on the VOC 2012 dataset with $\lambda = 0.01$. Proposal numbers are reported before near-duplicate and empty proposal removal. The global CRF produces predominantly large proposals, which best fit roughly 8% of the objects. Most of the proposals are generated by the localized CRFs, which outperform the other model types for a large majority of the objects. The availability of small-object models during training has no effect on ensemble accuracy up to about 2000 proposals. For higher proposal budgets, small-object models improve the 70%-recall by up to 1%. The ABO and 50%-recall for ensembles trained without small-object models differ by less than 0.005. The results of the presented approach in our experiments are almost entirely due to the (global and localized) CRF models.

For high proposal budgets (above 5000), our approach has a 50%-recall of 94%: only 6% of the objects in the VOC 2012 dataset are missed. Some of these objects are shown in Figure 3, along with randomly sampled images from the dataset. The missed objects are in part tiny segments, such as very distant animals, and in part ground-truth annotations that have poor bottom-up evidence, such as people behind reflective car windows. As expected, for images that were randomly sampled for Figure 3, the 50%-recall of our approach is 100%.

6.2. VOC 2012 bounding box accuracy

We evaluate the accuracy of bounding box proposals that can be obtained with our approach by taking the bounding boxes of our region proposals. We follow the evalua-

Method	# prop.	ABO	50%-recall	70%-recall	time	p-value
Our approach, $\lambda = 0.2$	635	**0.732**	**0.861**	**0.634**	**0.8s**	–
CPMC [7]	646	0.704	0.785	0.609	252s	<**0.01**
GOP [21]	652	0.720	0.844	0.632	1.0s	<**0.01**
Global/Local [27]	1056	0.689	0.780	0.579	8s	<**0.01**
GOP [21]	1199	0.741	0.865	0.673	1.1s	<**0.01**
Our approach, $\lambda = 0.1$	1236	**0.759**	**0.890**	**0.685**	**0.9s**	–
RIGOR [19]	1299	0.735	0.832	0.657	5s	<**0.01**
Cat-Ind OP [12]	1536	0.718	0.821	0.624	119s	<**0.01**
SCG [3]	2125	0.755	0.871	0.664	5s	<**0.01**
Our approach, $\lambda = 0.03$	2133	**0.785**	**0.924**	**0.733**	1.1s	–
MCG ranked [3]	2199	**0.785**	0.897	0.721	30s	0.58
GOP [21]	2286	0.756	0.877	0.699	1.4s	<**0.01**
Our approach, $\lambda = 0.02$	2707	**0.793**	**0.930**	**0.762**	**1.4s**	-
GOP [21]	4186	0.766	0.889	0.715	1.7s	<**0.01**
Selective Search [32]	4374	0.735	0.891	0.597	2.6s	<**0.01**
Our approach, $\lambda = 0.01$	5144	**0.810**	**0.943**	**0.785**	1.9s	-
MCG [3]	5158	0.808	0.922	0.772	30s	0.14

Table 3: Quantitative results on the PASCAL VOC2012 dataset. Six state-of-the-art object proposal methods are compared to the presented approach. The methods are ordered by number of proposals. The table is divided into four tiers with similar proposal set sizes within each tier. Our approach outperforms most methods by a wide margin, with strong statistical significance, at the lowest running time.

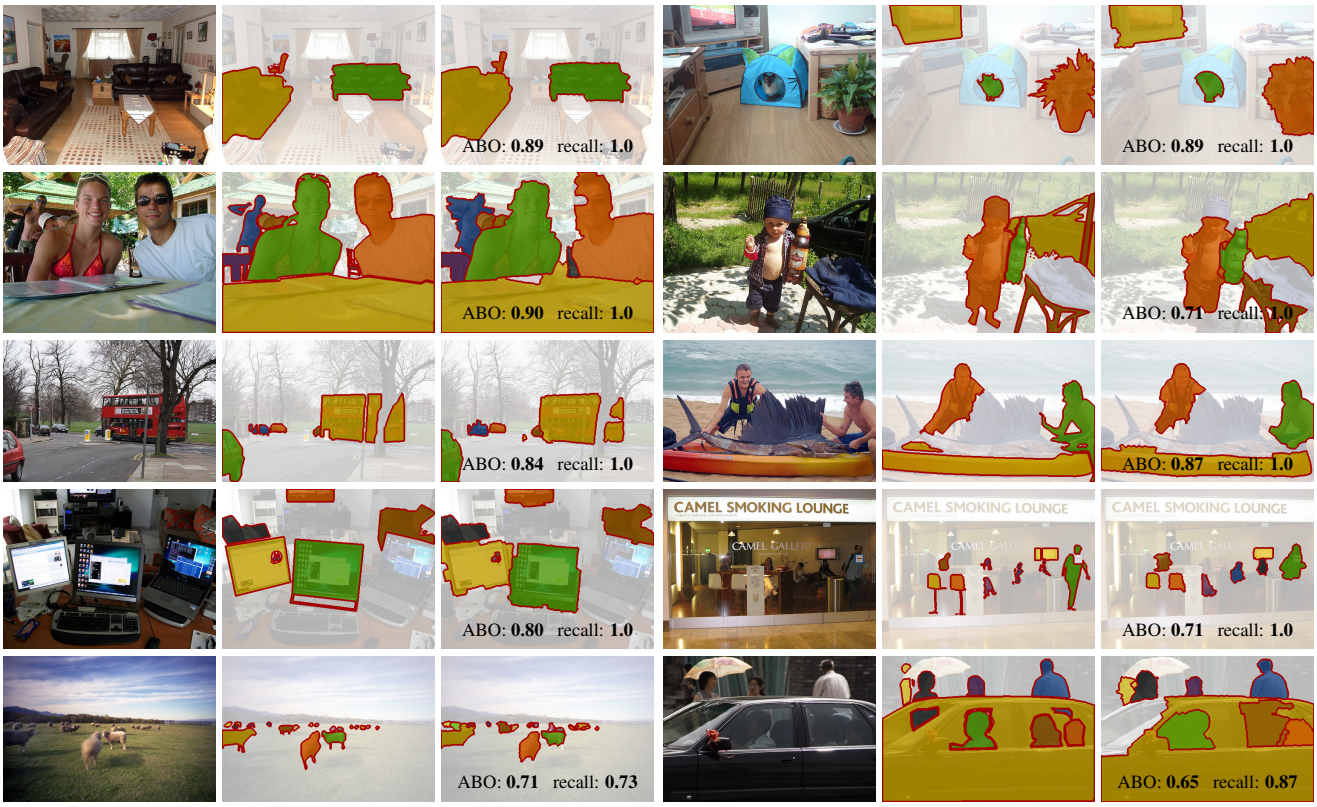
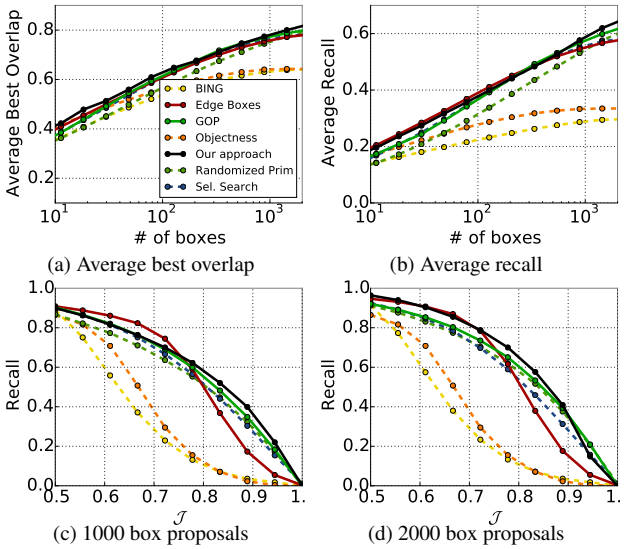


Figure 3: Qualitative results on the VOC 2012 dataset. The top four rows show **random** images that contain three or more objects. The bottom row shows images that have a ground-truth object that is not predicted well by our algorithm. For each image, the figure reports the ABO and the 50%-recall of our algorithm. See text for details.



(a) Average best overlap (b) Average recall

(c) 1000 box proposals (d) 2000 box proposals

Figure 4: Recall for bounding box proposals. (a,b) Average best overlap and average recall for a varying proposal budget. (c,d) Recall at different accuracy thresholds with 1000 proposals and 2000 proposals.

tion methodology of Krähenbühl and Koltun [21]. The results are shown in Figure 4. Objectness [1] and BING [8] perform well at 50%-recall but their performance degrades rapidly for higher recall thresholds. Edge boxes [35] performs best at 70%-recall but their performance also drops at more stringent accuracy levels. Our approach outperforms all alternatives at high accuracy levels ($\mathcal{J} > 0.8$).

We further compute the volume under surface (VUS) [21], average best overlap (ABO), and average recall (AR) [18] for 2000 bounding box proposals. Note that AR is known to be a particularly good predictor of detection performance [18]. The results are provided in Table 4. The presented approach outperforms all prior work in all accuracy measures.

Method	VUS	ABO	AR	Time
BING [8]	0.278	0.640	0.296	0.003s
Objectness [1]	0.324	0.643	0.335	2.2s
Edge boxes [35]	0.527	0.800	0.577	0.3s
Sel. search [32]	0.528	0.781	0.580	2.2s
GOP [21]	0.546	0.797	0.615	0.9s
Our approach	**0.558**	**0.819**	**0.644**	1.1s

Table 4: Bounding box proposal accuracy for 2000 proposals.

6.3. Microsoft COCO

We have evaluated our algorithm on the recent Microsoft COCO dataset [23]. The ground truth segmentation annotations in this dataset are quite rough. To deal with imprecise annotations, we disregard a 3-pixel band around the annotated boundaries in the evaluation. Table 5 reports the accuracy of our approach and of state-of-the-art proposal algorithms that could feasibly be run on this large dataset. Our approach achieves the highest 70%-recall. In ABO our approach outperforms prior work by 2 to 4 percentage points, in 50%-recall by 4 to 6 percentage points.

Dataset generalization. We have also trained models on the entire VOC 2012 segmentation dataset and then evaluated them on COCO. The results are reported in Table 5. Models trained on COCO and models trained on VOC perform similarly. This strongly suggests that our approach is capable of learning a general model of bottom-up object segmentation, biased neither to a specific dataset nor to specific object classes.

Method	# prop.	ABO	50%-rec.	70%-rec.
GOP [21]	5501	0.649	0.749	0.527
Sel. search [32]	6504	0.654	0.770	0.471
MCG [3]	5377	0.669	0.759	0.563
Ours, $\lambda = 0.3$	1920	0.626	0.717	0.437
Ours, $\lambda = 0.2$	4078	0.674	0.791	0.526
Ours, $\lambda = 0.1$	5175	0.689	0.809	0.565
Ours (VOC)	2027	0.628	0.707	0.462
Ours (VOC)	4331	0.676	0.781	0.558
Ours (VOC)	5480	0.690	0.802	0.573

Table 5: Generalization across datasets. We trained three ensembles on the Microsoft COCO dataset and three on the Pascal VOC2012 dataset, then tested all six on COCO. Ensembles trained on VOC generalize well to COCO.

7. Conclusion

We presented a new approach to bottom-up object segmentation. Our approach trains an ensemble of figure-ground segmentation models. When applied to an image, each model independently identifies candidate objects. The ensemble is trained jointly, enabling different models to specialize. We show that ensemble training can be reduced to a sequence of combinatorial optimization problems. The training procedure is general and accommodates different model types. The size and composition of the ensemble are optimized along with the parameters of the incorporated models, all for the same objective. Experimental results demonstrate that the presented approach significantly outperforms prior object proposal algorithms in terms of detailed shape overlap as well as bounding box overlap. The results also indicate that the trained ensembles generalize across datasets, suggesting that the presented approach is capable of producing generally applicable models of bottom-up object segmentation.

References

[1] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *PAMI*, 34(11), 2012. 2, 8

[2] P. Arbeláez, B. Hariharan, C. Gu, S. Gupta, L. D. Bourdev, and J. Malik. Semantic segmentation using regions and parts. In *CVPR*, 2012. 2

[3] P. Arbeláez, J. Pont-Tuset, J. T. Barron, F. Marqués, and J. Malik. Multiscale combinatorial grouping. In *CVPR*, 2014. 2, 6, 7, 8

[4] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 23(11), 2001. 2

[5] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu. Free-form region description with second-order pooling. *PAMI*, 2015. To appear. 1, 2

[6] J. Carreira, F. Li, and C. Sminchisescu. Object recognition by sequential figure-ground ranking. *IJCV*, 98(3), 2012. 1, 2

[7] J. Carreira and C. Sminchisescu. CPMC: automatic object segmentation using constrained parametric min-cuts. *PAMI*, 34(7), 2012. 1, 2, 3, 5, 7

[8] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. H. S. Torr. BING: Binarized normed gradients for objectness estimation at 300fps. In *CVPR*, 2014. 2, 8

[9] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 2

[10] M. Deza and M. Laurent. *Geometry of cuts and metrics*. Springer, 1997. 4

[11] P. Dollár and C. L. Zitnick. Structured forests for fast edge detection. In *ICCV*, 2013. 4

[12] I. Endres and D. Hoiem. Category-independent object proposals with diverse ranking. *PAMI*, 36(2), 2014. 1, 2, 3, 7

[13] M. Everingham, L. J. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman. The PASCAL visual object classes (VOC) challenge. *IJCV*, 88(2), 2010. 5

[14] P. F. Felzenszwalb, R. B. Girshick, D. A. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 32(9), 2010. 2

[15] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 59(2), 2004. 5

[16] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 1, 2

[17] B. Hariharan, P. Arbeláez, R. B. Girshick, and J. Malik. Simultaneous detection and segmentation. In *ECCV*, 2014. 1, 2

[18] J. Hosang, R. Benenson, P. Dollár, and B. Schiele. What makes for effective detection proposals? In *PAMI*, 2015. 8

[19] A. Humayun, F. Li, and J. M. Rehg. RIGOR: Reusing inference in graph cuts for generating object regions. In *CVPR*, 2014. 1, 2, 7

[20] K. Jain, M. Mahdian, E. Markakis, A. Saberi, and V. V. Vazirani. Greedy facility location algorithms analyzed using dual fitting with factor-revealing LP. *Journal of the ACM*, 50(6), 2003. 4

[21] P. Krähenbühl and V. Koltun. Geodesic object proposals. In *ECCV*, 2014. 2, 3, 4, 5, 7, 8

[22] C. H. Lampert, M. B. Blaschko, and T. Hofmann. Efficient subwindow search: A branch and bound framework for object localization. *PAMI*, 31(12), 2009. 2

[23] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 1, 2, 5, 8

[24] M. Mahdian, Y. Ye, and J. Zhang. Approximation algorithms for metric facility location problems. *SIAM Journal on Computing*, 36(2), 2006. 4

[25] T. Malisiewicz and A. A. Efros. Improving spatial support for objects via multiple segmentations. In *BMVC*, 2007. 2

[26] T. Malisiewicz and A. A. Efros. Recognition by association via learning per-exemplar distances. In *CVPR*, 2008. 2

[27] P. Rantalankila, J. Kannala, and E. Rahtu. Generating object segmentation proposals using global and local search. In *CVPR*, 2014. 2, 7

[28] M. Resende and R. Werneck. A hybrid multistart heuristic for the uncapacitated facility location problem. *European Journal of Operational Research*, 174(1), 2006. 4

[29] B. C. Russell, W. T. Freeman, A. A. Efros, J. Sivic, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *CVPR*, 2006. 2

[30] D. B. Shmoys. Approximation algorithms for facility location problems. In *Approximation Algorithms for Combinatorial Optimization*, 2000. 4

[31] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *JMLR*, 6, 2005. 2, 4

[32] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *IJCV*, 104(2), 2013. 1, 2, 7, 8

[33] P. A. Viola and M. J. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001. 2

[34] C. Yu and T. Joachims. Learning structural SVMs with latent variables. In *ICML*, 2009. 4, 6

[35] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, 2014. 2, 8