Segmentation Models Diversity for Object Proposals

Marco Manfredi[a], Costantino Grana[a], Rita Cucchiara[a], Arnold W.M. Smeulders[b]

[a] *University of Modena and Reggio Emilia, Italy*
[b] *ISLA, Informatics Institute, University of Amsterdam, The Netherlands*

Abstract

In this paper we present a segmentation proposal method which employs a box-hypotheses generation step followed by a lightweight segmentation strategy. We introduce diversity in segmentation strategies enhancing a generic model performance exploiting class-independent regional appearance features. Foreground probability scores are learned from groups of objects with peculiar characteristics to specialize segmentation models. We demonstrate results comparable to the state-of-the-art on PASCAL VOC 2012 and a further improvement by merging our proposals with those of a recent solution. The ability to generalize to unseen object categories is demonstrated on Microsoft COCO 2014.

Keywords: Segmentation, Supervised Learning, Object Proposals

1. Introduction

Automatic object segmentation is among the oldest topics in computer vision, and apparently one of the hardest, in view of the results obtained thus far. Other topics, such as image recognition and image search, have increased from a poor to a solid performance in just a decade. While first ignoring location information altogether [1, 2, 3], recognition and search have recently reintroduced locality where it now plays an important role [4, 5]. We can obtain object localization in the form of a set of box-hypotheses [6, 7] or precise segmentation masks [8, 9, 10].

Inspired by interactive segmentation, where every object is perfectly inscribed in a user-placed bounding-box and then segmented, our goal is to start from a set of automatically obtained bounding-boxes and for each of them extract a precise segmentation [11]. A clear problem with respect to the interactive segmentation setting is that

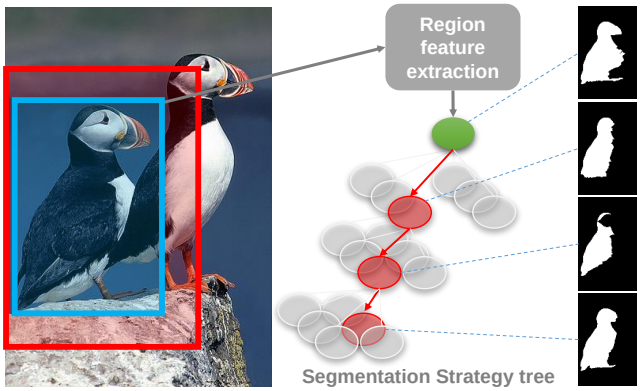

Figure 1: Segmentation strategy diversification is employed to produce diverse proposals.

the number of object candidates to analyze is in the order of 1000 per image and not only 1 per object, leading to large running times ([11] reports 6 to 10 minutes per image). We aim to develop a method to refine box-hypotheses scalable to thousands of proposals.

As objects may be discriminated from the background on the basis of their edge information, their texture, or other appearance cues, it is unlikely that there exists one single model for generic object segmentation [12, 13]. Differentiation and combination of several segmentation strategies is necessary to control object diversity [5]. One extreme approach for diversity is to build a new segmentation model for each new class of objects [11, 14]. A recognition step is thus required to select the appropriate model. Class-specific segmentation models are hard to apply in large-scale applications [15], and they are by definition not applicable to an unknown class of objects. We use the progress in the field of segmentation to strive for a class-independent approach [10, 9], while introducing diversity in the segmentation strategy to enhance its generic performance where needed.

Our approach starts with box-hypotheses built from edge statistics [6]. On the basis of lightweight superpixel features, we assess the probability of belonging to the foreground. The use of spatially-smooth visual features (e.g. geodesic distance) allows for accurate segmentations while avoiding any time-consuming regularization [16]. Rather, we rely on a simple threshold of the foreground probabilities to generate the

binary segmentations. We also avoid any proposal re-ranking [4, 9] delegating the ranking to the stage of the box-hypotheses. These choices allow for a fast segmentation proposal generation.

During training, diversity is included by unsupervised clustering, sorting objects into different types on the basis of regional appearance features. Ideally, each cluster contains a specific group of objects suited for a specific segmentation approach. For each group of objects, specialized segmentation models are learned. The same features are used to assign an unknown object to one of these clusters when applying the algorithm.

Our contributions are:

1. We propose a fast and class-independent segmentation technique, starting from recent methods for generating box-hypotheses;
2. By grouping objects into clusters, each suited for a specific segmentation strategy, we effectively achieve object-group diversity, reaching state-of-the-art results on PASCAL VOC 2012. We demonstrate how the learned segmentation strategies generalize to unseen categories on the Microsoft COCO 2014 dataset.
3. We further demonstrate a considerable improvement in segmentation accuracy over the state-of-the-art by enhancing the diversity after merging with a recent segmentation strategy [10].

The objects clusters obtained while diversifying the segmentation models are also used to highlight when our method or [10] are providing the best candidates. The highlight illustrates the importance of segmentation model diversity in the success of the integrated solution.

2. Related Work

Object localization with candidate segmentations has attracted a lot of attention in the last years [17, 10, 18, 11, 19, 9, 4, 12], mainly due to the improvement that precise localization offers in object recognition settings [8, 20].

The CPMC approach [4] uses multiple graph-cut computations at pixel-level to compute segmentation candidates from seeds placed on a grid over the image. The

region level affinities proposed in [9] have inspired our foreground probability score. Differently from our work, however, in the reference they are computed on bigger regions and transferred to a superpixel graph regularized in a CRF.

The approach in [19] is based on the idea that objects of different categories have similar local shapes. As a consequence, masks can be transferred from other objects and slightly adapted to the object of interest. The Geodesic Object Proposals technique [10] is based on geodesic distances to automatically placed foreground and background seeds. The use of a spatially-smooth feature as the geodesic distance makes the costly use of regularizing superfluous. We adopt the same tactic in our method.

In [12] the importance of segmentation in object recognition is stressed, along with a numerical demonstration of the importance of differentiating among segmentation techniques. The technique presented in [17, 21] combines edge detection, hierarchical segmentation and object proposals based on region grouping. Selective Search [5] uses segmentation strategy diversification by changing the criterion on which adjacent regions are being merged. The diversification enlarges the search space for possible objects. Both [11] and [18] use size as a cue to differentiate segmentation models, based on the idea that the relevance of visual features is related to object size. While [11] uses class-specific shape priors, [18] only relies on class-independent probabilistic models. In order to diversify segmentation strategies without including class information, we leverage regional level features, including size, in a hierarchically structured decision model.

In the interactive segmentation approach presented in [13], segmentation models are adapted to each object using two manually traced polygons to learn the optimal parameters of the segmentation model (e.g. feature importance). Our solution strives to a similar specialization in an automatic setting.

3. From Bounding Boxes to Segmentation Masks

Starting from a bounding-box R we want to outline the contained object. Locality in segmentation is of fundamental importance, and thus only a close neighborhood of the object is considered in the segmentation process.

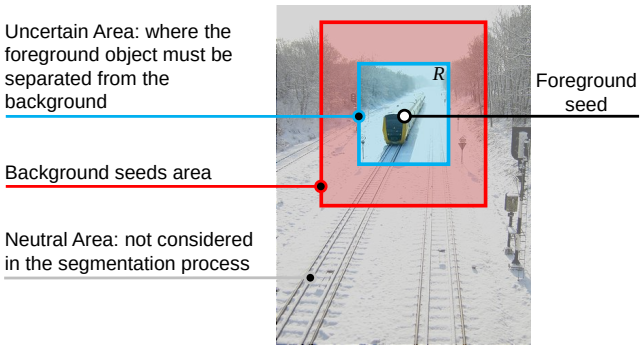

Figure 2: When segmenting the train, only its close neighborhood is used in the segmentation process. The foreground seed is placed at the center of the bounding-box.

We assume that the object is fully contained in R, by labeling the outside region as background. The area surrounding R, obtained by enlarging it by a 50% factor, defines the background area (used to model background information). We further assume that the center of R belongs to the object, using it as the foreground seed (Figure 2).

A superpixel over-segmentation of the image is computed, and each superpixel is labeled according to the area of maximum overlap. We obtain two sets of superpixels: the background seeds $\mathcal{B}$ and the foreground seed $\mathcal{F}$ (i.e. the superpixel containing the center of R).

A set of features (9 in total), presented below, is extracted from each superpixel and used in a supervised setting to compute a foreground probability score.

From $\mathcal{F}$ and $\mathcal{B}$ two color histograms are extracted representing the RGB color distributions of foreground and background (C_f and C_b respectively). For each superpixel S_i, we compute the similarity of its color histogram C_{S_i} with respect to C_f and C_b, and the difference between the two.

The geodesic distance to foreground and background seeds is another important feature of our framework. Following [10], a graph over the superpixel over-segmentation is created where the edges between adjacent nodes are weighted using an edge probability score [22].

The geodesic distance between superpixel S_i and S_j, $G(S_i, S_j)$, is the sum of the edge costs on the shortest path between the two, that can be computed with Dijkstra's

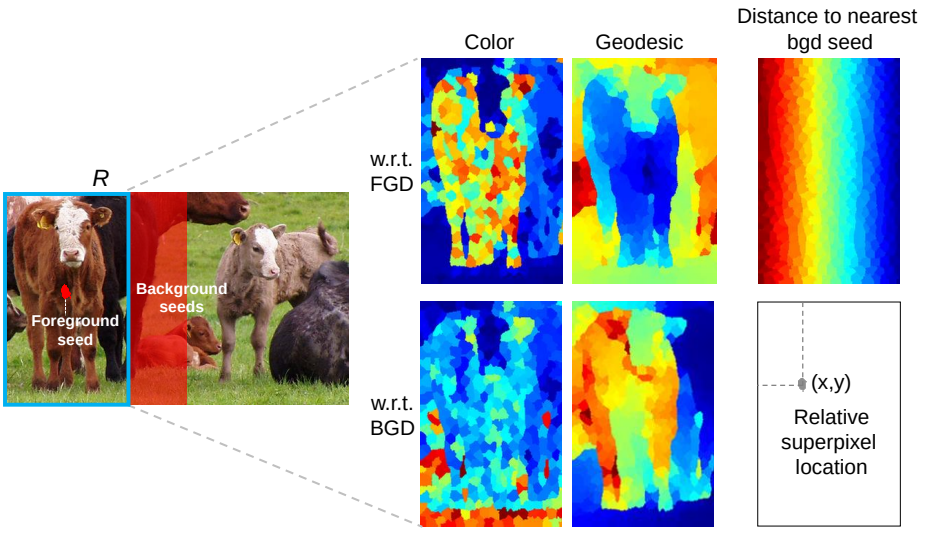

Figure 3: Visualization of the superpixel features extracted from an object, detected by the bounding-box R (best viewed in color).

algorithm. For each superpixel S_i inside R we compute the geodesic distance to the foreground seed $\mathcal{F}$, $G(S_i, \mathcal{F})$, and to the background seeds $\mathcal{B}$, $G(S_i, \mathcal{B})$, computed as:

$$G(S_i, \mathcal{B}) = \arg\min_{S_j \in \mathcal{B}} G(S_i, S_j). \tag{1}$$

$G(S_i, \mathcal{F})$, $G(S_i, \mathcal{B})$ and $G(S_i, \mathcal{F}) - G(S_i, \mathcal{B})$ are added to the feature set of superpixel S_i.

Location information for superpixel S_i are included computing the x and y coordinates relative to the center of R. Figure 3 visualizes the features used in our model. As it can be observed, when R touches the border of the image but the object does not, the geodesic distance to background seeds has a different interpretation depending on the observed superpixel S_i. The more S_i is near the image border, the more unreliable $G(S_i, \mathcal{B})$ become, because of the increasing distance to the nearest background seed. This observation led us to add the Euclidean distance to the nearest background seed as the last feature of the method.

The Edge Boxes technique [6] is used, with default settings, to produce a set of box-hypotheses. For each candidate region we compute one segmentation mask. To

set the number of segmentation proposals per image, we adjust the average number of box-hypotheses to consider. The purpose of Edge Boxes is to cover all objects with as few candidates as possible. The more tight a bounding-box is to an object, the better it is. The overlap metric used to measure object hypotheses accuracy does not evaluate if a bounding-box fully contains an object or not. Instead, this is a crucial property for our method since the area outside the box is used to initialize background seeds. Leaving part of the object outside the box would potentially lead the background region to leak in the foreground object. For this reason we enlarge each proposed bounding-box by 20%, that was found to work well in practice. The segmentation masks proposed by our method are ranked relying on the candidates ordering provided by Edge Boxes.

Learning. For each object belonging to the training set, all the superpixels inside its bounding-box are extracted and described with the aforementioned visual features. Each superpixel is labeled as foreground if the overlap with the ground-truth segmentation mask is greater than 50%, background otherwise.

We formulate the problem of computing foreground probabilities as a supervised binary classification, where foreground (background) superpixels are the positive (negative) samples. A Random Forest classifier is set to output a foreground probability score in the range $[0, 1]$ so that simply thresholding the scores at 0.5 provides a binary segmentation.

In the following, we will refer to the segmentation model learned on the entire training set as *Generic Segmentation Strategy* (GSS).

Since in our approach bounding-box candidates come from an automatic method, the tightness of each box to the detected object can not be estimated. Using ground-truth bounding-boxes in training would potentially lead our model to fail in test (where tightness varies greatly). Thus, we decided to compute box-hypotheses also in training, and select the tightest one that fully contains the object.

4. Diversifying Segmentation Strategies

Segmentation algorithms' performance are heavily influenced by several object characteristics like object saliency, scene cluttering and occlusion. Depending on the

specific segmentation method, the relative impact of these characteristics changes.

Knowing in advance how an object looks like could potentially be helpful to configure the selected segmentation algorithm to perform best on the specific object. When segmenting an unknown object, such specific information are not available, but we can rely on region properties extracted from the object area (i.e. the bounding-box R). For example, the size of the object can be estimated from the size of R, along with its position in the image. A positive property of region features is that they do not exploit ground-truth information and thus they can be computed identically for training and testing objects.

The diversification of segmentation strategies works as follows: (i) we form groups of objects that share similar region characteristics; (ii) for each group a separate segmentation model is learned. The idea is that a segmentation model learned on a group of objects is adapted to their characteristics and thus can segment them better than the generic model (learned on all objects). At testing time, the same region properties are used to infer the group to which each unknown object belongs, along with its segmentation model. The extension of our solution exploiting diversified strategies will be referred to as *Segmentation Strategy Diversification Tree* (SSDT).

4.1. Region Features

We design the object region features to encode relevant characteristics for segmentation. Although some object properties potentially impact any kind of segmentation algorithm (e.g. weak object edges), we design the following features to influence the behavior of our specific segmentation method. Given the bounding-box R capturing an object extent, we measure:

- **Size**: the size of R captures the approximate object size, and it potentially impacts the informativeness of superpixel features.
- **Appearance w.r.t. surroundings**: The color difference between the inner part of R and its surroundings are a rough measure of the color saliency of the object.
- **Internal complexity**: The internal structure complexity of R encode a description of the object edges.

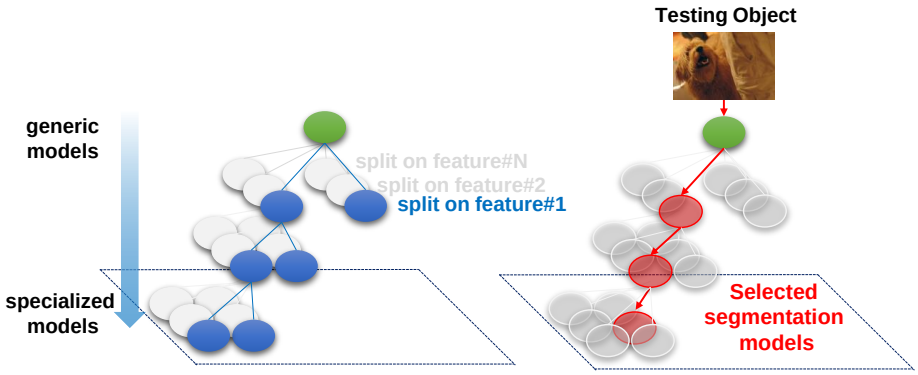

Figure 4: The root node, containing all objects, is hierarchically split in clusters using N different region features. Each node of the tree has a separate segmentation model, that is more specialized the more deep in the tree the node is. At testing time, the path that maximizes diversity is chosen and the segmentation models on the path are used to compute alternative segmentation masks.

- **External complexity**: The structure complexity of the surroundings of R are a reasonable measure for scene cluttering.
- **Location**: As shown in Figure 3, knowing the location of the object with respect to the image borders gives insights on geodesic feature reliability.

Size (`SIZE`) is computed as the area of R, the appearance w.r.t. surroundings (`COLOR_DIFF`) is the color histogram comparison between the foreground seed and the background seeds. Internal and external complexities (`FGD_COMPLEX` and `BGD_COMPLEX` respectively) are encoded in 8-bin edge magnitude histograms computed on an edge probability map [22]. The location feature (`LOC`) is computed as the percentage of the perimeter of R that touches the image border.

4.2. Hierarchical Object Clustering

Starting from the entire training set of objects $\mathbf{O}_{all}$, and focusing on a region feature $feat$, we split the set in two using a k-means clustering computed on $feat$. Two non-overlapping sets of objects are identified, $\mathbf{O}_1$ and $\mathbf{O}_2$. For example, selecting `SIZE` as the region feature, $\mathbf{O}_1$ and $\mathbf{O}_2$ separate small objects from big objects. We can repeat the procedure by spliting $\mathbf{O}_1$ in $\mathbf{O}_{11}$ and $\mathbf{O}_{12}$, and $\mathbf{O}_2$ in $\mathbf{O}_{21}$ and $\mathbf{O}_{22}$. Proceeding in this way, we would obtain a binary tree of object groups. The splitting continues until

the fourth level of the tree is reached or the number of object to be splitted is lower than K (200 in our experiments).

Since the available region features are 5, at each node we compute 5 binary splits (one per feature), thus obtaining a pyramid of object clusters (see Figure 4). The root of the tree contains all objects, the leaves contain only a small portion of objects, with peculiar region properties.

For each node we now compute a separate segmentation model, that is tailored to the characteristics of the objects belonging to that node. Moving from the root to the leaves, segmentation models become more and more specialized. Using k-fold cross-validation ($k = 5$ in our experiments) we measure segmentation accuracies for all objects of the node, in all the nodes of the tree. The segmentation accuracy of an object can be traced from the root to a leaf following a specific path.

The purpose of the tree is to diversify segmentation strategies, that is creating segmentation models complementary to those available at shallower levels. To measure the complementarity of the segmentation model of node i to its father, we select from its objects $\mathbf{O}_i$ the subset $\mathbf{O}_i^+$, containing the objects that have an accuracy gain moving from the father to the child. We then measure the average accuracy gain on these objects ($\overline{\Delta}_i$), i.e. the average difference between the segmentation accuracies of the objects of $\mathbf{O}_i^+$ using the two models. The complementary score of a node is computed as:

$$C_i = \frac{|\mathbf{O}_i^+|}{|\mathbf{O}_i|} \cdot \overline{\Delta}_i \tag{2}$$

For all the nodes (except for the root) a complementary score is computed.

At testing time, for each candidate object, multiple paths can be taken, since at each node the object belongs to one cluster per feature. The path with the highest complementary score is thus chosen at every tree level.

Each of the 5 region features employed in the tree splits the available objects in two groups, using k-means. It is useful, for presentation purposes, to label each of the obtained group with the peculiar characteristic it has after the splitting. Splitting using the `SIZE` feature separates *small* objects from *big* objects. Splitting on the `COLOR_DIFF` feature, separates objects with *high fgd-bgd contrast* from objects with *low fgd-bgd*

Table 1: Results using the Generic Segmentation Strategy (GSS) on the validation set of PASCAL VOC 2012. Here no diversification strategy is applied.

Method	#	ABO	Cov.	R50	R70	Time
CPMC[4]	646	.703	.850	.784	.609	250s
GOP [10]	652	.720	.815	.844	.632	1s
GSS	655	.701	.781	.837	.601	3.0s
GOP [10]	1018	.733	.834	.853	.665	1.1s
GSS	991	.717	.790	.860	.628	3.4s

contrast. In the same way, the `LOC` feature separates object *far from/near to image border*. For the `FGD_COMPLEX` and `BGD_COMPLEX` features, the clustering is performed on 8-bin histograms, representing the strength of the edges inside/outside the bounding-box. Empirically, we found that a splitting on these features separates object with *weak internal/external edges* from the ones with *strong internal/external edges*.

5. Experiments

We evaluate segmentation proposals accuracies on the PASCAL VOC 2012 segmentation dataset [23]. The segmentation quality of a proposed segmentation w.r.t. a ground truth mask is measured with the intersection over union metric [23] (also called overlap), defined as the ratio between the intersection of the two masks divided by their union. To evaluate a set of proposals, three measures are used: the average best overlap (ABO), the covering and the recall [4]. The ABO measures the best segmentation accuracy achieved by all proposals for any given object, averaged over the entire dataset. The recall is the percentage of objects that have a best overlap greater than a specific threshold. The covering measure is defined similarly to the ABO but it is weighted by the object size. This measure highlights the segmentation performance on bigger objects. In every experiment the average number of proposals per image is reported for each method for a clear comparison.

The superpixel over-segmentation is computed using geodesic k-means [24], providing about 1000 superpixels per image. Color distributions are modeled with 128-bin

Table 2: Segmentation accuracies on PASCAL VOC 2012 of the Segmentation Strategy Diversification Tree (SSDT) w.r.t. the Geodesic Object Proposals solution [10].

Method	#	ABO	Cov.	R50	R70	Time
GOP [10]	2008	.750	.850	.868	.697	1.4s
GSS	2015	.734	.801	.864	.654	4.5s
SSDT	2025	.741	.810	.878	.685	5s
GOP [10]	3983	.762	.857	.882	.714	1.7s
SSDT	4030	.760	.816	.903	.718	9.5s

Bag of Words histograms compared with the Histogram Intersection metric.

All experiments are computed on a Intel Core i7 machine with 16GB of RAM. A public implementation of our method along with the trained models and object clusters is available online [1].

5.1. Generic Model Performance

In the first experiment we compare the object proposals segmentation accuracy of GSS with CPMC [4] and Geodesic Object Proposals (GOP) [10]. In this setting only the generic segmentation model, learned on all objects of the training set, is used. Near duplicates removal is applied to avoid multiple identical segmentations. The experiments are presented for 650 and 1000 proposals, to be directly comparable with the other techniques.

In Table 1, a numerical analysis is presented. The ABO of GSS is comparable with the one of CPMC while the other metrics highlight a big difference in how the two methods behave. We achieve a lower covering w.r.t. CPMC but a higher recall at 50, meaning better accuracies on small objects but worst on big objects. CPMC, using costly pixel-level segmentations, runs almost 100 times slower than our method. Limiting our segmentations to the generic model only does not allow our method to compete favorably w.r.t. the GOP approach, that is the most accurate.

[1] http://imagelab.ing.unimore.it/segmprops/

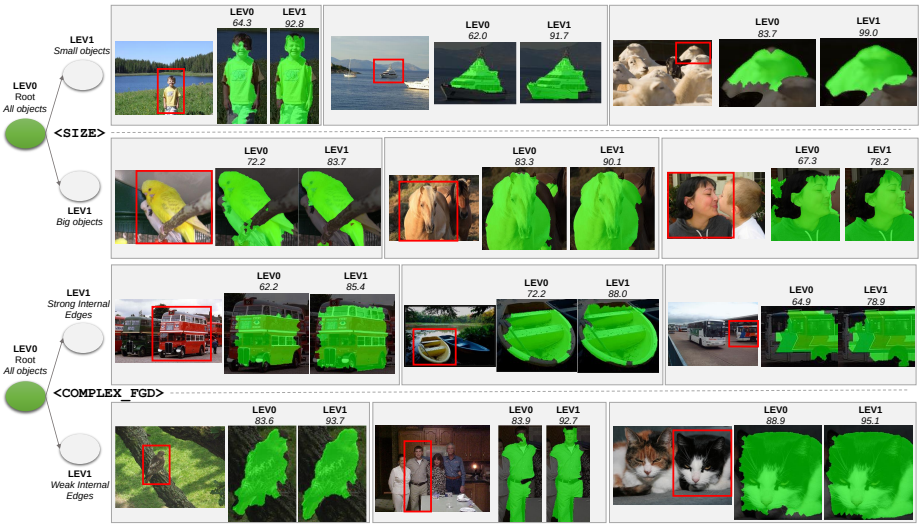

Figure 5: Segmentation results obtained through segmentation strategy diversification. Two cases are analyzed: splitting on bounding-box size and splitting on the box internal complexity. On the left, the object tree split is presented. On the right, several examples per node are presented comparing the results obtained with the generic model (LEV0) and the specialized model (LEV1). Learning specialized segmentation strategies helps in segmentation: for example, the model learned for objects with strong internal edges (row 3) avoids the excessive shrinking of the generic model capturing the true objects extents.

5.2. Adding Diverse Proposals

In this section, SSDT presented in Section 4 is employed to measure the effectiveness of segmentation strategy diversification.

The more bounding-boxes we use, the more objects we find, but as stated in [6], 1000 box-candidates are sufficient to cover almost every object with 50% of overlap. Segmentation strategy diversification, on the other hand, does not allow to find new objects, but enhances segmentation performance on the ones already detected. To rely on a sufficiently high number of detected objects we decided to start diversifying segmentation strategies only for proposing more than 1000 object candidates. This means that, while for 650 and 1000 proposals we only use the generic model, for 2000 and 4000 we use more than one model per object exploiting the object tree. Specifically, to obtain 2000 proposals per image on average, we use two segmentation models per bounding-box: the generic one, and one from the first level of the tree; to obtain 4000

proposals we leverage the entire depth of the tree.

In Figure 5, a visual comparison of segmentation results using different segmentation models is presented. The purpose of the image is to show how choosing a specialized segmentation strategy instead of the generic one affects the final segmentation. As can be observed, the specialized segmentation models (computed at level 1 of the tree) are capable of producing a different set of masks w.r.t. the generic one.

In Table 2, the effectiveness of diversifying segmentation models for our method is tested comparing with the Geodesic Object Proposal solution. SSDT achieves very high recall values, but it suffers, as previously noted for GSS, on big objects, achieving lower covering results. We will further investigate this behavior in the next section. The gap between SSDT and GOP almost disappears at 4000 proposals, showing how the proposal of alternative segmentations for each detected object is able to enhance segmentation accuracy.

The number of box-candidates given by Edge Boxes would allow to generate 2000 segmentation proposals per image on average without segmentation strategy diversification. We did this experiment, doubling the number of considered bounding-boxes per image; results are reported in Table 2 under the GSS label. To reach 4000 proposals, multiple segmentations per bounding-box are necessary, since many of the bounding-boxes proposed by Edge Boxes overlap and are filtered out. This experiment shows that, once objects are correctly detected, it is more effective to stop proposing new boxes, focusing instead on diversifying segmentation models for each box.

5.3. Merging Object Candidates

In this experiment, we investigate segmentation diversity by mixing the proposals from our solutions (both GSS and SSDT) with the ones of Geodesic Object Proposals [10]. We chose GOP because on one hand it is a state-of-the-art technique that is both class-independent and fast, as our algorithm, and on the other hand it starts from different initializations (seeds instead of boxes) producing free-form segmentations (instead of ours box-constrained masks). Mixing the segmentations from different methods has been done before [17]. Our contribution in this section is to highlight the

Table 3: Comparison of segmentation proposals accuracies on the validation set of PASCAL VOC 2012.

Method	#Props	ABO	Covering	Recall@50	Recall@70	Time
CPMC [8]	646	0.703	0.850	0.784	0.609	250s
GSS	655	0.701	0.781	0.837	0.601	3.0s
GOP [10]	652	0.720	0.815	0.844	0.632	1.0s
GSS+GOP	660	0.718	0.815	0.848	0.650	2.4s
GSS	991	0.717	0.790	0.860	0.628	3.4s
GOP [10]	1018	0.733	0.834	0.853	0.665	1.1s
GSS+GOP	1023	0.740	0.834	0.865	0.684	3.0s
C.I.O.P. [9]	1536	0.718	0.840	0.820	0.624	119s
SCG [17]	2000	0.751	0.835	0.870	0.661	5s
SSDT	2025	0.741	0.810	0.878	0.685	5s
GOP [10]	2008	0.750	0.850	0.868	0.697	1.4s
SSDT+GOP	2028	0.769	0.852	0.896	0.726	4.1s
Sel. Search [5]	4374	0.735	0.786	0.891	0.597	2.6s
MCG [17]	4000	0.801	0.862	0.914	0.761	30s
SSDT	4030	0.760	0.816	0.903	0.718	9.5s
GOP [10]	3983	0.762	0.857	0.882	0.714	1.7s
SSDT+GOP	3991	0.785	0.860	0.911	0.763	7.1s

complementarity of the two methods through visual examples and quantitative analysis, with the aim of a deeper understanding of the algorithms' behavior.

To propose N segmentation candidates per-image on average, we generate $N/2$ with our method and $N/2$ with GOP. Both methods have their specific way of assessing candidates quality, used internally to sort the proposals. When merging the set of proposals coming from our method with the one proposed by GOP, one candidate at a time from each list is taken, and results are filtered to avoid near duplicates. The results of the two separate approaches along with the ones of the merged solution, compared with several state-of-the-art algorithm, are presented in Table 3. As it can be

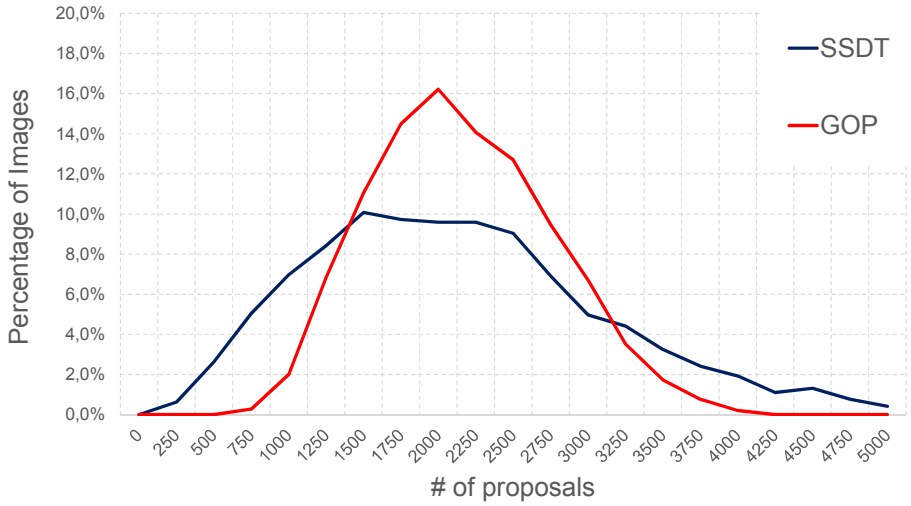

Figure 6: Distribution of the actual number of segmentation candidates provided by SSDT and by Geodesic Object Proposals when 2000 proposals are requested on average.

observed, the performance of our solutions alone become more and more competitive with respect to the state-of-the-art the more proposals are added through diversification. The contribution of our method to the merged solution follows the same trend: while at 650 proposals the merged solution is comparable with the performance of GOP alone, at 4000 the gap on ABO is more than 2 points. Notably, the merged solution at 2000 proposals achieves better results than both GOP and SSDT alone at 4000.

Figure 7: Number of proposals output by our method (SSDT) and GOP in very different images taken from the VOC 2012 dataset.

Analysis. We conducted an extensive analysis on the merged solution SSDT+GOP 4000 (2000 proposals per method). The first interesting observation is that when we ask the two methods to provide 2000 proposals per-image on average, the actual number of candidates per-image varies greatly. This is a desirable property, since the algorithms should adapt to the *segmentation complexity* of a scene [25]. For simple images (e.g. low clutter, high fgd/bgd separablity) we expect few proposals, while for highly complex scenes (e.g. low contrast, strong textures) a wide coverage of all possible object locations/sizes is requested. In Figure 6 the two methods are compared measuring the distribution of object proposals on the entire validation set of PASCAL VOC 2012. While the average number of proposals is about 2000 for both, their distributions are different. This behavior is due to the technique used to asses the number of proposals for each specific image: where Edge Boxes searches for non-overlapping box-candidates with sufficient quality (i.e. edge support), GOP relies on the number of non-overlapping segmentations computed from a fixed number of seeds. A visual example is presented in Figure 7, where the number of proposals output by each method is reported for three sample images.

The second result is that the objects clusters computed with SSDT effectively highlight strengths and weaknesses of both our method and GOP. For each object of the validation set, given the merged set of proposals SSDT+GOP 4000, we can check which of the two methods has provided the best candidate. Aggregating this information for all objects in each cluster of the tree provides an average quality measure for the two methods. We call α_{ours} the percentage of objects that our method is able to cover with better accuracy. When all objects are considered $\alpha_{ours} = 43\%$, but the situation changes greatly depending on the group of object that we analyze. For objects far from image border $\alpha_{ours} = 49\%$ while for big objects $\alpha_{ours} = 31\%$. In Figure 8, four clusters of objects are analyzed: the first two depict situations in which our method excels, the last two presents two clusters where GOP is generally the most accurate.

5.4. Generalization on COCO 2014

In this section we test the generalization capabilities of our method on the recently proposed Microsoft COCO 2014 dataset [15]. The COCO dataset is composed of

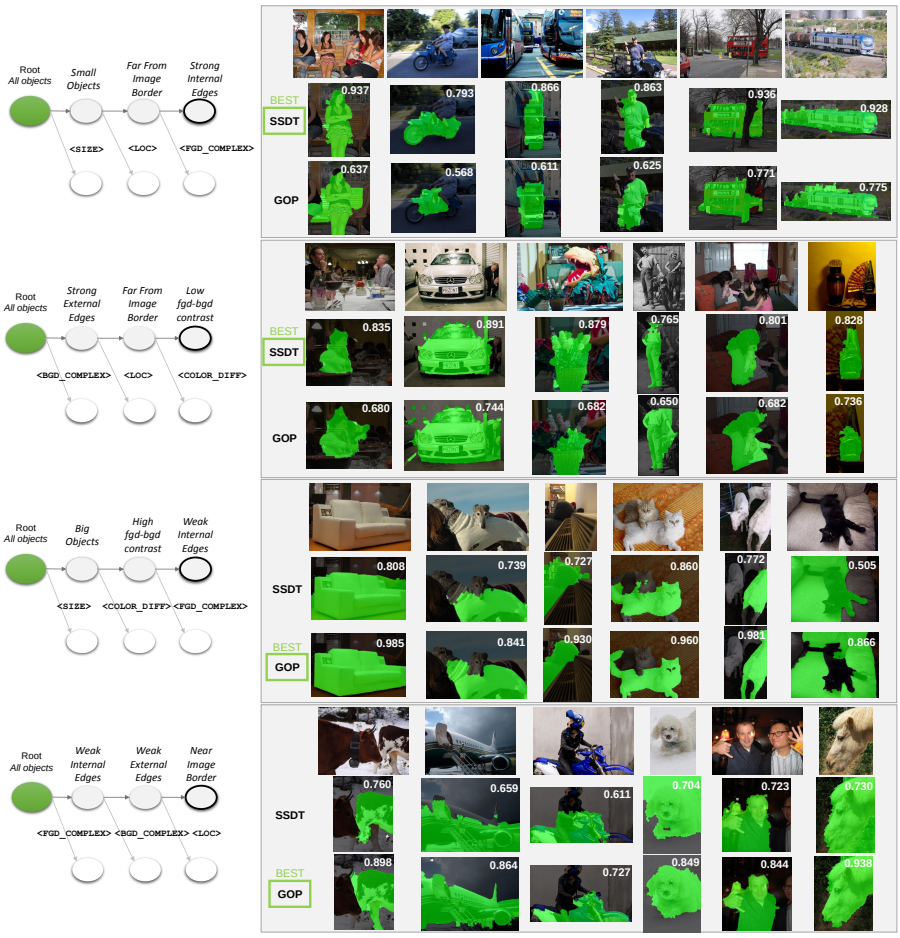

Figure 8: Visualization of segmentation results on PASCAL VOC 2012 for SSDT and GOP [10]. On the left the analyzed group of objects is presented. For each object the best object candidate obtained by SSDT and by GOP are reported. The first two rows present objects groups for which SSDT generally obtains better segmentations than GOP, the opposite applies in the last two rows. SSDT and GOP are indeed complementary and the use of both allows to greatly enhance the segmentation diversity of a set of proposals.

82783 training images and 40504 validation images used for testing. In this experiment we use the segmentation models learned on VOC 2012 and we test on the validation images of COCO. Results are presented in Table 4. The first observation is that COCO is much more challenging than VOC 2012, since all methods have a drop in performance of about 30%. A possible explanation for the performance loss can be found

Table 4: Segmentation accuracies on Microsoft COCO 2014 validation set. All methods marked with * are trained on Pascal VOC2012, the others are trained on the COCO training set.

Method	#Props	ABO	R50	R70
RIGOR [26]	1650	.530	.576	.348
SCG [21]	2000	.530	.575	.324
MCG [21]	2000	.565	.615	.352
SSDT*	2035	.518	.569	.344
GOP [10]*	2066	.524	.572	.358
SSDT+GOP*	2051	.545	.605	.392
MCG [21]	4000	.585	.665	.395
SSDT*	4018	.558	.629	.385
GOP [10]*	4037	.546	.588	.381
SSDT+GOP*	4032	.567	.635	.424

in the higher percentage of small objects in the COCO corpus w.r.t. VOC 2012 [15]; small objects are generally more difficult to detect and to outline by superpixel-based approaches like ours and MCG [21]. The SSDT learned on VOC 2012 is capable of obtaining comparable results to SCG [21] and RIGOR [26], showing that the regional appearance features used to specialize our segmentation strategies are generic enough to be effective on previously unseen object categories. Moreover, when SSDT proposals are merged with GOP proposals we observe the same performance gain measured in VOC 2012. Differentiating segmentation strategies is again a key factor to boost object proposals quality. The merged solution (learned on VOC 2012) is able to achieve comparable results to MCG, a state-of-the-art object proposal technique learned on COCO.

6. Conclusions

We have presented an effective segmentations proposals technique initialized by bounding-boxes, which is fast enough to be scalable to thousands of proposals per image. We demonstrated that diversifying segmentation strategies works both when

applied to our method and when used to integrate diverse algorithms. Our method lies in between generic segmentation models and class-specific solutions, providing diversity while maintaining class-independence for state-of-the-art results.

[1] F. Perronnin, J. Sánchez, T. Mensink, Improving the fisher kernel for large-scale image classification, in: Proc. Eur. Conf. Comput. Vision, 2010, pp. 143–156.

[2] H. Jégou, M. Douze, C. Schmid, P. Pérez, Aggregating local descriptors into a compact image representation, in: Proc. IEEE Int. Conf. Comput. Vision Pattern Recognit., 2010, pp. 3304–3311.

[3] J. Uijlings, A. Smeulders, R. Scha, Real-time visual concept classification, IEEE Trans. on Multimedia 12 (7) (2010) 665–681.

[4] J. Carreira, R. Caseiro, J. Batista, C. Sminchisescu, Semantic segmentation with second-order pooling, in: Proc. Eur. Conf. Comput. Vision, 2012, pp. 430–443.

[5] J. Uijlings, K. van de Sande, T. Gevers, A. Smeulders, Selective Search for Object Recognition, Int. J. Comput. Vision 104 (2) (2013) 154–171.

[6] C. L. Zitnick, P. Dollár, Edge boxes: Locating object proposals from edges, in: Proc. Eur. Conf. Comput. Vision, 2014.

[7] M.-M. Cheng, Z. Zhang, W.-Y. Lin, P. Torr, Bing: Binarized normed gradients for objectness estimation at 300fps, in: Proc. IEEE Int. Conf. Comput. Vision Pattern Recognit., 2014, pp. 3286–3293.

[8] J. Carreira, C. Sminchisescu, CPMC: Automatic Object Segmentation Using Constrained Parametric Min-Cuts, IEEE Trans. Pattern Anal. Mach. Intell. 34 (7) (2012) 1312–1328.

[9] I. Endres, D. Hoiem, Category-Independent Object Proposals with Diverse Ranking, IEEE Trans. Pattern Anal. Mach. Intell. 36 (2) (2014) 222–234.

[10] P. Krähenbühl, V. Koltun, Geodesic object proposals, in: Proc. Eur. Conf. Comput. Vision, Springer, 2014, pp. 725–739.

[11] D. Weiss, B. Taskar, Scalpel: Segmentation cascades with localized priors and efficient learning, in: Proc. IEEE Int. Conf. Comput. Vision Pattern Recognit., 2013, pp. 2035–2042.

[12] T. Malisiewicz, A. A. Efros, Improving spatial support for objects via multiple segmentations, in: British Machine Vision Conference, 2007.

[13] Z. Kuang, D. Schnieders, H. Zhou, K.-Y. Wong, Y. Yu, B. Peng, Learning image-specific parameters for interactive segmentation, in: Proc. IEEE Int. Conf. Comput. Vision Pattern Recognit., IEEE, 2012, pp. 590–597.

[14] Q. Dai, D. Hoiem, Learning to localize detected objects, in: Proc. IEEE Int. Conf. Comput. Vision Pattern Recognit., 2012, pp. 3322–3329.

[15] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft coco: Common objects in context, in: Proc. Eur. Conf. Comput. Vision, Springer, 2014, pp. 740–755.

[16] Y. Boykov, M.-P. Jolly, Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images, in: Proc. IEEE Int. Conf. Comput. Vision, Vol. 1, 2001, pp. 105–112.

[17] P. Arbelaez, J. Pont-Tuset, J. Barron, F. Marques, J. Malik, Multiscale combinatorial grouping, in: Proc. IEEE Int. Conf. Comput. Vision Pattern Recognit., IEEE, 2014, pp. 328–335.

[18] Z. Ren, G. Shakhnarovich, Image segmentation by cascaded region agglomeration, in: Proc. IEEE Int. Conf. Comput. Vision Pattern Recognit., IEEE, 2013, pp. 2011–2018.

[19] J. Kim, K. Grauman, Shape sharing for object segmentation, in: Proc. Eur. Conf. Comput. Vision, Springer, 2012, pp. 444–458.

[20] J. Carreira, F. Li, C. Sminchisescu, Object recognition by sequential figure-ground ranking, Int. J. Comput. Vision 98 (3) (2012) 243–262.

[21] J. Pont-Tuset, P. Arbelaez, J. T. Barron, F. Marques, J. Malik, Multiscale combinatorial grouping for image segmentation and object proposal generation, arXiv preprint arXiv:1503.00848.

[22] P. Dollar, C. Zitnick, Structured forests for fast edge detection, in: Proc. IEEE Int. Conf. Comput. Vision, 2013, pp. 1841–1848.

[23] M. Everingham, L. Van Gool, C. Williams, J. Winn, A. Zisserman, The pascal visual object classes (voc) challenge, Int. J. Comput. Vision 88 (2) (2010) 303–338.

[24] F. Perazzi, P. Krahenbuhl, Y. Pritch, A. Hornung, Saliency filters: Contrast based filtering for salient region detection, in: Proc. IEEE Int. Conf. Comput. Vision Pattern Recognit., IEEE, 2012, pp. 733–740.

[25] S. D. Jain, K. Grauman, Predicting sufficient annotation strength for interactive foreground segmentation, in: Proc. IEEE Int. Conf. Comput. Vision, IEEE, 2013, pp. 1313–1320.

[26] A. Humayun, F. Li, J. M. Rehg, Rigor: Reusing inference in graph cuts for generating object regions, in: Proc. IEEE Int. Conf. Comput. Vision Pattern Recognit., 2014, pp. 336–343.