

Boosting Object Proposals: From Pascal to COCO

Jordi Pont-Tuset and Luc Van Gool
Computer Vision Lab, ETH Zürich, Switzerland
{jponttuset, vangool}@vision.ee.ethz.ch

Abstract

Computer vision in general, and object proposals in particular, are nowadays strongly influenced by the databases on which researchers evaluate the performance of their algorithms. This paper studies the transition from the Pascal Visual Object Challenge dataset, which has been the benchmark of reference for the last years, to the updated, bigger, and more challenging Microsoft Common Objects in Context. We first review and deeply analyze the new challenges, and opportunities, that this database presents. We then survey the current state of the art in object proposals and evaluate it focusing on how it generalizes to the new dataset. In sight of these results, we propose various lines of research to take advantage of the new benchmark and improve the techniques. We explore one of these lines, which leads to an improvement over the state of the art of +5.2%.

1. Introduction

The sliding windows approach [27] dominated the object detection literature for several years. It consists in sweeping a large amount of locations and sizes in the image, represented by rectangular windows, and then applying an object/non-object classifier on each of them. The number of swept windows is usually in the order of millions, so to get a reasonable speed one needs to rely on very simple classifiers carefully implemented using tools like integral images.

In order to reduce the number of windows to consider, and thus being able to apply more advanced classification machinery, the concept of *objectness* [1] arose; which provides a reduced number of windows, in the order of hundreds or thousands, aiming at keeping the *true* objects as much as possible. These techniques have been improving to get more quality and efficiency until the present days [28, 7, 20, 25], widely known as *bounding-box object proposals* methods: they *propose* a reduced set of windows aiming at representing the *bounding box* of the true *objects*.

Object windows give us the location and scale of the objects, but they miss a crucial queue for object detection: its shape. While a window representation might be good



Figure 1. Example segmentation annotations from the three studied databases: Pascal SegVOC12 [11], SBD [12], and COCO [19]

enough for certain types of roughly-rectangular objects such as faces or cars, it is certainly not precise enough to represent more *deformable* objects such as animals or plants; and one might need the *mask* of the object for applications like image editing or inpainting.

In this context, the *segmented object proposal* techniques arose [5, 9], whose objective is to provide a reduced set of object *segmentations*, that is, a small set of binary image masks indicating sets of pixels belonging to an object in the image. It currently has become an active topic of research in computer vision [18, 4, 22, 15, 25, 17, 6, 10].

Annotated databases play a crucial role in the improvement of these techniques, both for training the algorithms and for benchmarking and comparing different approaches. The Pascal Visual Object Challenge [11] database has been the driving force behind many of the improvements in latest years, providing high-quality bounding-box and segmentation annotations for thousands of images.

With the increasing computation power given by modern computers and the annotation capacity provided by Amazon Mechanical Turk¹, however, much bigger annotated databases are taking its place and further pushing the literature. ImageNet [23] provides annotated bounding boxes on half million images. Semantic Boundary Dataset (SBD) [12] extends the Pascal segmented annotations to more than ten thousand images. Most notably, the recent Microsoft Common Objects in Context (COCO) [19]

¹<http://www.mturk.com>

pushes the number of annotated segmented images one order of magnitude, to more than one hundred thousand images. Figure 1 shows an example image and segmentation annotations from Pascal, SBD, and COCO.

The first contribution of this work (Section 3) is the analysis of the properties of COCO compared to SBD and Pascal. We focus on the size of the databases, the balance between the number of objects annotated on different categories, and the localization and size of the annotations. By showing the properties of the objects that are annotated we make researchers more aware of the new challenges.

The second contribution of this paper (Section 4) is the in-depth analysis of the state of the art in segmented object proposals in COCO. We will evaluate how well all techniques, that were designed and evaluated on Pascal, scale to COCO, focusing on the challenges detected in the previous section. The weaknesses of current methods are also opportunities to improve, which is what we encourage with these experiments.

Finally, Section 5 proposes a generic improvement to object proposal techniques by taking into account the results in previous sections, which leads to a 5.2% boost over state of the art. This improvement is only an example of the applications that can be given to the analysis we perform. In the conclusions (Section 6) we sketch other future lines of research we envision.

We present a website (<http://vision.ee.ethz.ch/~biwiproposals/boosting-coco>) where all the results from this paper are available interactively and kept up to date with the latest state of the art. Notably, the best proposal from each technique on all COCO images is available to visualize, directly from the browser.

Some of the closer works to ours are:

- Hoiem et al. [13]: They analyze the small subtleties on Pascal (boxes) after years of research. We evaluate (segmented) object proposals on a very new dataset and so we focus on the new generalities and caveats.
- Torralba and Efros [24]: They focus on how well object recognition generalizes across datasets. We are analyzing what should researchers, biased towards Pascal, be aware of when generalizing their techniques to COCO.
- Hosang et al. [14]: They focus on detection (boxes) performance on Pascal and ImageNet, while we focus on the segmentation task on COCO, especially targeting the novel properties of it with respect to Pascal.

2. State of the Art in Object Proposals

We focus the review of the state of the art in *segmented object proposal* techniques, that is, those that provide pixel-accurate object proposals.

One of the seminal works in this area was presented by Carreira and Sminchisescu [5, 6], who sample a set of foreground and background seeds and produce the object pro-

posals by *parametric min cuts*. The same year, Endres and Hoiem [9, 10] presented an object proposal technique using a hierarchy of regions taking contour and occlusion cues into account. Arbeláez *et al.* [2] presented their technique based on the combination of regions and parts toward semantic segmentation. Kim and Grauman [17] propose a partial shape matching algorithm of annotated objects to detected contours, shared between classes. Uijlings *et al.* [25] produce a set of partitions of the image, for a variety of color spaces, and perform a *selective search* algorithm to produce a set of object proposals.

Most recently, the topic has been reactivated by much more efficient algorithms, especially thanks to new, very efficient, and precise contour detectors [16, 8]. Krähenbühl and Koltun [18] compute geodesic distance transforms, and the object proposals are their critical level sets. Arbeláez *et al.* [4, 21] compute segmentation hierarchies at different image resolutions and combine their regions via an efficient combinatorial grouping strategy. Rantalankila *et al.* [22] perform a global and local search in the space of sets of superpixels. Humayun *et al.* [15] speed up parametric min-cuts by reusing the involved graphs.

3. Analysis of the Databases

We have selected three of the most representative databases for object detection and segmentation:

- **Segmentation challenge of Pascal 2012 Visual Object Classes (SegVOC12) [11]:** The Pascal challenges were a series of computer vision competitions on object detection, object segmentation, action detection, etc. Apart from non-overlapping object segmentations, the annotations provide *void* pixels that were not taken into account in the evaluation. We will use the training and validation sets of year 2012.
- **Berkeley Semantic Boundaries Dataset (SBD) [12]:** This database adds segmentation annotations to the *main* set of images in the detection challenge of Pascal (which was only annotated with bounding boxes). The annotations have the same format than SegVOC12 except they don't have *void* pixels. Those images that were annotated in the segmentation challenge are re-annotated. We use the training and validation sets.
- **Microsoft Common Objects in Context (COCO) [19]:** The biggest and newest of the three databases. In contrast with SegVOC12 and SBD, the object segmentations can overlap (a single pixel can be part of two objects). We use the training and validation sets of the 2014 release (COCO14). We ignore the instances marked as *crowd*.

The three databases consist of annotated objects of different categories. Table 1 summarizes the number of categories, images, and object instances in each database.



Figure 2. **Qualitative examples:** Images and segmentation annotations from Pascal SegVOC12, SBD, and COCO14.

	Number of Categories	Number of Images	Number of Instances
SegVOC12	20	2 913	6 934
<i>Train+Val</i>		<i>1 464+1 449</i>	<i>3 507+3 427</i>
SBD	20	11 355	26 843
<i>Train+Val</i>		<i>8 498+2 857</i>	<i>20 172+6 671</i>
COCO14	80	123 287	886 284
<i>Train+Val</i>		<i>82 783+40 504</i>	<i>597 869+288 415</i>

Table 1. Sizes of the databases

As we can observe, each new database is roughly one order of magnitude larger than its predecessor, giving more significance to the experiments at each step. COCO14 has four times the number of categories of previous databases, giving more representativity to the results. These 80 categories are grouped into 12 super-categories.

The annotations in the three datasets are both in the form of bounding boxes and segmentations. Figure 2 shows some images of the three datasets, along with their object segmentations. The SegVOC12 annotations include *void* regions (in cream color) that mark uncertainty or difficult pixels. SBD annotations are shown in the same format than SegVOC12 but they do not have *void* pixels.

COCO annotations have some particularities with respect to SBD and SegVOC12. First, some of the annotations mark objects that are really challenging to segment, as the wine glasses that are semi-transparent. Some segmentations are less accurate than those in the previous datasets, such as the chair in Figure 2: the contour is not very accurate and the wall seen in between the bars is not subtracted. Finally, annotations in COCO overlap between them, meaning that some pixels are annotated as two different objects. The dogs in Figure 2, for instance, are annotated as dogs and sofa, which should be handled carefully in the evaluation.

3.1. Category Balance

The analysis of the results on these databases is commonly separated into per-category performance, thus the distribution of objects among categories is an important fea-

ture of the database. An uncompensated distribution might entail bias towards a specific type of object in the evaluation. Let us therefore analyze the percentage of objects among every category in the three databases. Figure 3 plots the frequency of each category (percentage of the whole set of objects that fall into that category). Categories are sorted by frequency and plotted with respect to the accumulated frequency of all categories up to that category, to align the results with 20 and 80 categories (note that all end at 100%). The most common categories are labelled.

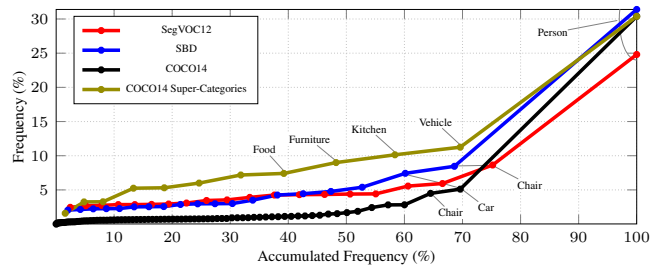


Figure 3. **Object category and super-category frequencies** with the most common categories annotated. X axis is accumulated frequency, to plot different number of categories in the same axis.

In the three databases, *person* is by far the most common category (around 25-30%). In COCO, this is especially exaggerated: the second category is at 5%, 59 categories have less than 1% of the objects, and 20 less than 0.5%. SegVOC12 and SBD categories, and especially super-categories in COCO, are more balanced. One should be aware of this unbalance when performing the evaluation, in order not to bias the results and interpret them correctly.

3.2. Annotated Instances Localization

Another potential source of bias is the localization of the annotated instances. Are objects situated in the center of the image more common than those on the borders?

We rescale all images and annotations to 600×600 pixels, and we count the number of annotated objects that overlap with every pixel position. Figure 4 shows the *heat map* of the localization of the annotated objects, being red the

most common localization (1), dark blue meaning that no object overlaps that pixel (0), and the range in between meaning that the value at that pixel is a certain ratio with respect to the value at the most frequent pixel.

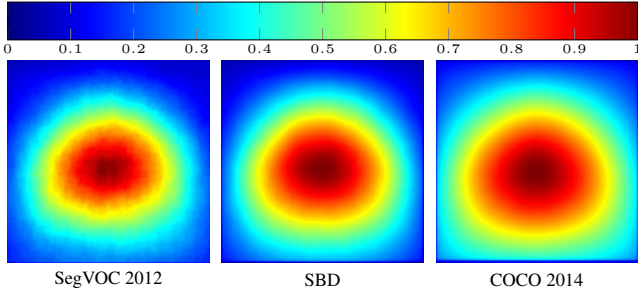


Figure 4. **Ground-truth object localization frequency:** Heat map of the localization of the annotated objects.

The three databases are biased towards objects in the center of the image. COCO is the one with more percentage of objects close to the border of the lower part of the image. The upper-half borders of the images are almost equally scarce in the three databases.

3.3. Annotated Instances Areas

The size of the annotated objects might also be a differentiating factor between databases. To explore it, we calculate the percentage of the image area that every annotated object occupies and compute a histogram. For all three databases, the majority of the objects represents a very small percentage of the image, so the three histograms of instance areas have a very pronounced peak close to 0 but are not visually very informative.

To grasp the differences between the databases graphically, Figure 5 plots the Cumulative Distribution Function (CDF) of the percentage of image area occupied by an-

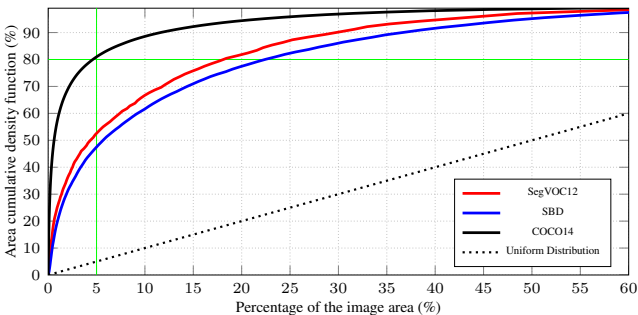


Figure 5. **Annotated objects area:** Cumulative distribution of the percentage of image area occupied by the annotated objects for SegVOC12, SBD, and COCO, i.e., percentage of instances whose area is below a certain threshold. As a baseline, we plot the function of a uniform distribution. The plot is cut at 60% for the sake of readability, but should go up to 100%.

notated objects, i.e., the percentage of annotated instances whose area is below a certain percentage of the image.

The plot shows that indeed the majority of the objects in the three databases are small (above the uniform distribution $\cdots\cdots$), and that COCO has the most pronounced bias. To put a particular example, the green line (---) highlights that while 50% of SegVOC12 objects are below 5% of area, in COCO 80% of the annotated objects are below this threshold. In the other axis, the percentile 80% is at 20% of the area for SegVOC12 and at 5% for COCO.

4. Analysis of the State of the Art

4.1. Evaluation Overview

As a first approach to the analysis of the results, we evaluate the recall of all State-of-the-Art (SoA) object proposal techniques, i.e., we count the percentage of annotated objects for which there is a segmented object proposal (not box) in the set whose Jaccard J (Intersection over Union) is higher than certain thresholds.

As a rule of thumb, $J = 0.5$ represents a very rough approximation, $J = 0.7$ an approximate representation, and $J = 0.9$ an accurate segmentation [18]. To summarize the different behaviors in a single value, we compute the Average Recall (AR) over different values for J , as proposed in [14], in the range of $J \in [0.5, 1]$.

Figure 6 shows the AR values with respect to the number of proposals for the validation sets of Pascal SegVOC12 and SBD; and for the training set of COCO14. We use the training set of COCO14 in order to fairly evaluate our improvements in the validation set.

There is a noticeable and consistent decrease in the quality of the proposals from SegVOC12 to COCO, suggesting that COCO is indeed a more challenging dataset. It could also happen that the techniques are trained for SegVOC12 and they are not generalizing well and should be trained on COCO. This work is focused on the former, and so finding the particularities that make COCO different from previous datasets; and leave the latter as a future work, with the help of the conclusions of this work.

In terms of comparing the object proposal techniques, MCG obtains the best results in all range of number of proposals and in all datasets. Table 2 shows the timing results of all techniques, in which we observe that GOP is considerably more efficient than the rest of techniques.

GOP	SCG	GLS	SeSe	RIGOR	MCG	CPMC	CI	ShSh
1.0	6.1	7.7	16.2	31.1	41.5	≥ 150	≥ 150	≥ 150

Table 2. Timing (in seconds) for all considered techniques

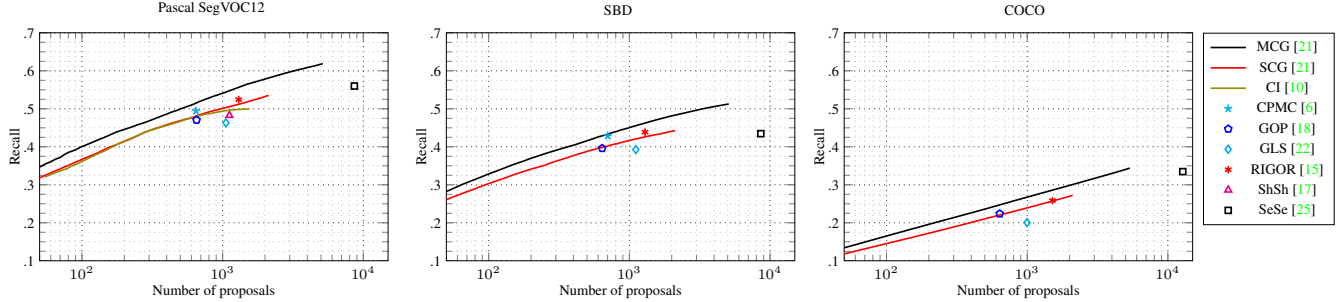


Figure 6. **Average Recall for segmented proposals.** Percentage of annotated objects for which there is a proposal whose overlap with the segmented ground-truth shapes (not boxes) is above different J thresholds, averaged in the range of $J \in [0.5, 1]$, for different number of proposals per image. Results on SegVOC12 validation, SBD validation, and COCO training sets.

4.2. Per-Category Evaluation

This section analyzes the quality obtained by the state-of-the-art methods on each of the 80 object categories in COCO. Figure 7 shows the per-category Average Best Overlap (ABO) (mean best J) for all studied techniques at 650 proposals per image. For those techniques that provide more than 650 proposals and do not rank them, a random selection was performed on each image. Although it might be argued that this is not fair for those techniques, it is out of the scope of this paper to re-parameterize all techniques to get 650 proposals per image (not even possible for some of them); since we are not focusing on the technique comparison but on the general behavior on COCO.

Categories are sorted by *category difficulty*, as the mean quality obtained by all techniques. We also plot the mean area of the annotated objects in each category (.....), as the percentage of the image area they occupy.

We observe a clear general trend in all techniques, that is, difficult/easy categories are approximately the same for all methods. Selective Search (SeSe —) is the only one that slightly separates from this trend: while it performs below the mean in all easy categories, it is slightly above the mean at the difficult ones. The plots also show a certain correlation between category difficulty and the mean area of their annotated objects: the easy categories are the big ones, while the small ones are the most challenging.

To get an overall idea of how the techniques compare at this number of proposals per image, we rank all of them for each category. Table 3 shows the mean position in the ranking and the number of times each technique is the best ranked. (Perfect result would get a mean ranking of 1 and would be the best in all 80 categories.)

	MCG	GOP	SCG	RIGOR	SeSe	GLS
Mean rank	1.36	3.01	3.03	3.49	4.44	5.68
Times best	59	13	0	2	6	0

Table 3. SoA ranking in per-category results

Again, please keep in mind that some methods could be optimized to get this number of proposals and potentially improve their results.

4.3. Area and Localization of the Proposals

Section 3.3 analyzes the distribution of the annotated objects size. This section performs the same analysis but on the state-of-the-art object proposal techniques. Figure 8 shows the CDF of the generated proposals (as in Figure 5) on the training set of COCO. We also plot the CDF for the annotated objects and uniform distribution for reference.

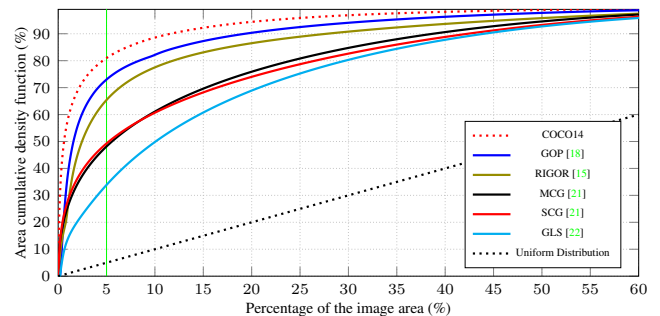


Figure 8. **State-of-the-art proposals area:** Cumulative distribution of the percentage of image area occupied, as in Figure 5.

We can observe significant differences between techniques illustrated, for instance, by the percentage of proposals whose area is below 5% of the image area (green line —): 73% in case of GOP (—) and 34% for GLS (—). In all cases, the percentage of small objects is even more significant in COCO, although GOP gets close.

To quantify the bias towards small objects, we compute the area under these curves. A uniform distribution (.....) would give 0.5, and 1 would correspond to the extreme case in which all proposals have a single pixel. Table 4 shows these values for the SoA techniques and for the annotated objects in COCO (.....).

Next, as done in Section 3.2 for the annotated objects, Figure 9 shows the heat map of the proposal localization for

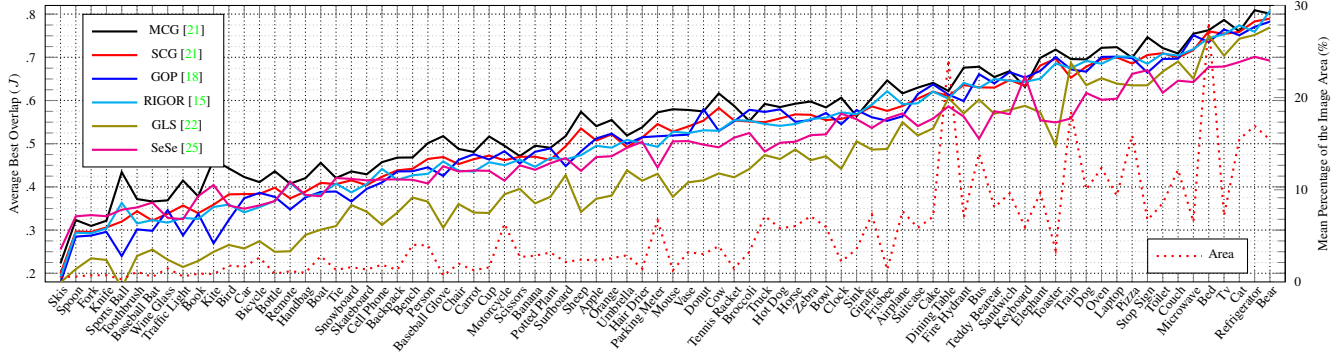


Figure 7. **State-of-the-art per-category evaluation:** Per-category ABO on COCO14 train. All techniques are evaluated at around 650 proposals per image. The categories are sorted by *category difficulty*.

COCO	GOP	RIGOR	MCG	SCG	GLS	Uniform
0.95	0.93	0.91	0.87	0.86	0.83	0.50

Table 4. Bias towards small objects

the different proposal techniques. Comparing these plots to those of the annotated objects (top-left), all techniques are considerably less biased towards the center of the image.

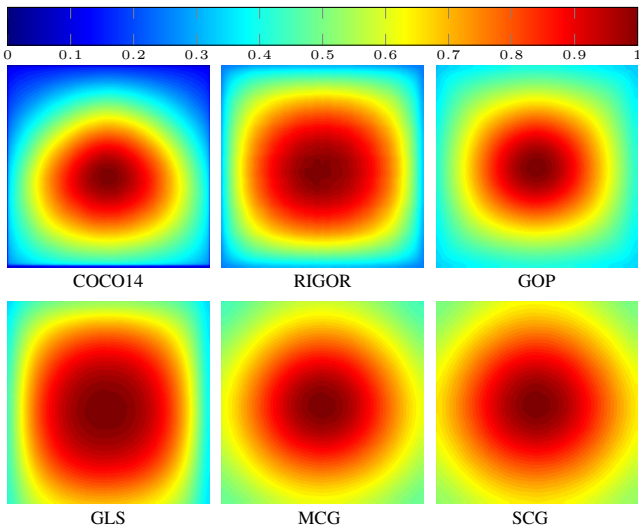


Figure 9. **Object proposals localization frequency:** Heat map of the localization of the proposals by SoA techniques and COCO.

Symmetries are also characteristic: while COCO has more objects on the lower half of the image, the current techniques are almost perfectly symmetrical in all directions. In other words, SoA techniques do not take into account whether the image is upside down or not. There are also differences in terms of shape: while most techniques tend to have a rectangular distribution, MCG and SCG have almost perfectly circular distributions.

To quantify the bias towards the center of the image, we compute the volume under the heat map (recall that heat

maps are normalized to 1). A perfectly evenly-distributed result would give 1, the extreme case of bias in which all proposals are a single pixel in the center of the image would approach 0. Table 5 shows these values for the object proposal techniques and for the annotated objects in COCO.

COCO	GOP	RIGOR	MCG	GLS	SCG	Uniform
0.51	0.59	0.63	0.70	0.72	0.73	1.00

Table 5. Bias towards the center of the image

Again, we observe significant differences between techniques, being some closer to the center-biased COCO, and others closer to the uniform distribution.

4.4. Quality with respect to the Object Areas

As we have shown in Section 3.3, COCO is characterized by having much more percentage of small annotated objects than its predecessors and, as seen in Section 4.2, the categories with smaller objects tend to be more difficult for current techniques. For these reasons, this section delves into the analysis of the results with respect to the size of the annotated objects.

A first attempt to visualize this data would be to plot the best overlap (J) on all annotated objects with respect to the percentage of the image area they occupy. Having so many annotated objects, however, makes this plot unreadable. We instead compute the bi-dimensional histogram of these points, by counting the number of occurrences at each square bin at a particular area and quality. Figure 10 shows the heat map representing this histogram for MCG [4] and GOP [18] on COCO14 train.

The trend of both techniques is clear: the smaller the object, the worse the results. Comparing both techniques, the most noticeable difference is at the objects smaller than $2^{-9} \approx 0.2\%$, where GOP drops its performance drastically while MCG holds better; but in any case almost all objects in this range are clearly missed by the techniques (less than

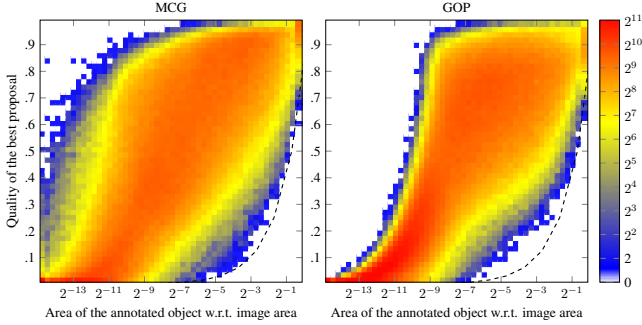


Figure 10. MCG vs GOP - ABO versus Area: Bi-dimensional histogram on COCO comparing MCG and GOP at the same number of proposals per image (≈ 650)

$J = 0.5$). The bigger objects are marginally better represented by MCG while the middle ones by GOP.

The dashed line corresponds to the quality of the proposal consisting of the whole image. Intuitively, the intersection with the ground-truth object would be the annotated object itself and the union would be the whole image, so the intersection over union of the full-image proposal would be the percentage of area occupied by the object, thus $y = x$.

In other words, we can avoid having any result below the dashed line by adding a single proposal to the set: the full image. This gives us an idea of why big objects are somehow *easier* to segment: it is more likely to overlap with them *by chance*.

4.5. Maximum quality to expect in COCO

Superpixel techniques are commonly used to reduce the computational load of the segmentation algorithms, while keeping the achievable quality almost unchanged. In this section we evaluate how true this statement is in COCO, especially focusing on the small object instances; which we have show that are very numerous in COCO (Section 3.3) and current object proposals techniques struggle to represent them correctly (Section 4.4).

We compute the SEEDS superpixels [26] (one of the best-performing techniques to date) on the COCO training dataset, and we evaluate how well we could perform if we had an oracle merging the superpixels optimally to represent the annotated objects. Table 6 shows the achievable Average Best Overlap, as well as the achievable Average Recall, both computed as in Section 4.1.

Number of Superpixels	416	609	758	1074	3714
Av. Best Overlap COCO	0.587	0.625	0.649	0.681	0.769
Average Recall COCO	0.383	0.425	0.452	0.494	0.623
Boundary Recall BSDS	0.973	0.986	0.992	0.995	0.996

Table 6. SEEDS superpixel evaluation: Best representation given by an oracle selecting the optimal superpixels

The results show that COCO objects are very challenging even if we could perform a perfect combination of state-of-the-art superpixels. With 1074 superpixels, for instance, we cannot even reach an AR of 0.5, and with four times the number of superpixels we only get to 0.62.

As a sanity check, we compute and evaluate the SEEDS superpixels also on the well-established BSDS500 [3]. The last row of Table 6 shows the percentage of annotated boundaries that we can recover from the superpixels (boundary recall). In this case, the performance of the superpixels is the expected: an almost perfect result already for less than 1000 superpixels.

So are current superpixel techniques as bad in COCO as the results show? Or some of this loss of performance could also be due to the lack of accuracy in COCO annotations (as we can observe in Figure 2)? In other words, it is clear that we cannot expect an achievable quality of $J = 1$ or full recall, but how far from this we can achieve in COCO is still to be proven. Having a subset of COCO annotated very precisely by different individuals, for instance, would be a way to find out, but the results of this section suggest that we would not be close $J = 1$.

5. Boosting Object Proposals

Up until this point, we have shown that COCO is indeed challenging for current state of the art, especially for having very small objects. We could ask, however, whether all the SoA techniques fail at the same objects systematically, or instead their performance in the annotated objects is not correlated. This could give us a sense of their *combinability*, that is, whether we would benefit from combining the proposals from different techniques.

To analyze it, we compute the quality that we would get if we had an oracle choosing the best proposal on each object from the full set of proposals from different techniques. Table 7 shows the average recall, as done in Section 4, as well as the same value for the techniques alone (in italics).

	MCG	SCG	GOP	GLS	SeSe	RIGOR
MCG	<i>34.37</i>	35.37	35.64	35.86	40.82	36.73
SCG		<i>27.19</i>	30.32	30.28	37.89	32.20
GOP			<i>22.36</i>	26.42	36.44	29.80
GLS				<i>20.04</i>	35.80	28.79
SeSe					<i>33.46</i>	37.24
RIGOR						<i>25.84</i>

Table 7. Average Recall ($\times 100$) of the combined techniques: If an oracle selected the best proposal from each technique on every object on the training set of COCO14

We observe significant improvements in the majority of the combinations, and especially between MCG and SeSe, which leads to an improvement of +6.45 with respect to the best of the two. The global combination of all the tech-

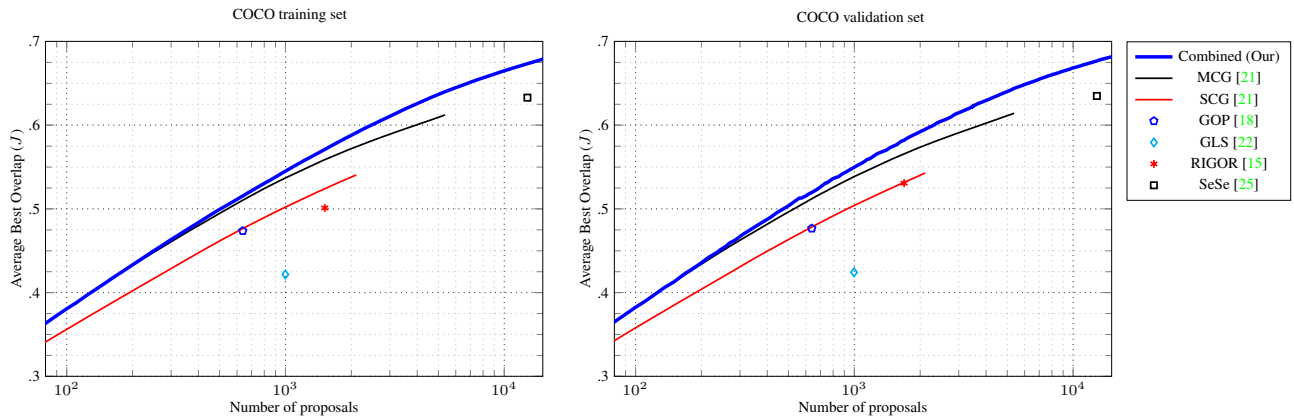


Figure 11. **Achievable quality of current object proposal techniques and the suggested combination:** Average Best Overlap (Jaccard index) with respect to the number of proposals per image, on COCO14 train (left) and val (right)

niques leads to $AR = 42.93$. In a realistic environment, in the absence of an oracle, however, this combination can only be obtained by merging the sets of proposals from different techniques, which increases the number of proposals significantly; so we would like to reduce the size of the whole set, keeping the quality as much as possible.

Intuitively, we would like to combine a subset of each set of proposals into the global pool. For the techniques that give ranked proposals, we will simply take the N best-ranked proposals, and for the techniques without ranking, we will consider a random subset of proposals. To learn the number of proposals N_i that we take from each technique i , we will use the Pareto front optimization presented in [4], which aims at finding the parameterizations that correspond to the Pareto front of number of proposals versus quality.

Figure 11 (left) shows the ABO on COCO train of the state of the art techniques, as well as the proposed combinations of proposals (—). As we can see, the full set of combined proposals effectively improves the quality of all techniques. On top of that, we are capable of reducing the number of proposals being better than all techniques down to 300, where the combined quality converges to that of MCG.

Figure 11 (right) shows the results on COCO validation, where we have taken the parameters learnt in the training set and used them directly on validation. As we can observe, this technique does not overfit and the improvement in quality is kept also at test time. Overall, we achieve an improvement of +3.2 points (5.2%) with respect to the full set of MCG and +4.2 (6.7%) with respect to SeSe at their same number of proposals per image.

6. Conclusions and Discussion

This paper analyzes the state of the art in segmented object proposals, especially focusing on the challenges encountered when going from Pascal and SBD to COCO:

- We first provide an in-depth analysis of the three databases: their number of images and categories, the balance of their types of objects, and where and how big the annotated objects are. We show that the three of them, especially COCO, are biased towards small objects, centered in the image.
 - We then provide an exhaustive and comprehensive review of the state of the art in segmented object proposals, and we thoroughly evaluate them on the three databases. We analyze how aligned the SoA techniques are with the bias observed in COCO towards small objects and the center of the image. We also show how challenging small objects are for current SoA, and we question the maximum quality that we can expect in COCO.
 - Finally, we propose an algorithm to combine current SoA techniques in COCO to boost their performance as an example of the room for improvement available in COCO.
- As future lines that we believe this work envisions:
- We have shown that the strong bias of the annotated databases towards small and centered objects are not present in the SoA proposals. Re-sampling the proposals to fit the distribution of annotated objects size and location could entail a significant improvement.
 - The smallest annotated objects of COCO, and especially those semi-transparent, are extremely challenging to segment. We have shown that, in fact, they are not even captured by superpixels of a very fine granularity. This can push us into the meeting point between detection and segmentation. Would we better *segmenting* these type of objects simply by *detecting* them?
 - The Pareto combination of proposals relies on getting the *top* proposals from each technique, emulated by a random sampling when a ranking is not available. We believe that having better rankings, or *diverse* rankings, could provide further improvement to the combination of techniques.

References

- [1] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *CVPR*, pages 73–80, 2010. [1](#)
- [2] P. Arbeláez, B. Hariharan, C. Gu, S. Gupta, L. Bourdev, and J. Malik. Semantic segmentation using regions and parts. In *CVPR*, 2012. [2](#)
- [3] P. Arbeláez, M. Maire, C. C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *TPAMI*, 33(5):898–916, 2011. [7](#)
- [4] P. Arbeláez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. In *CVPR*, 2014. [1](#), [2](#), [6](#), [8](#)
- [5] J. Carreira and C. Sminchisescu. Constrained parametric min-cuts for automatic object segmentation. In *CVPR*, 2010. [1](#), [2](#)
- [6] J. Carreira and C. Sminchisescu. CPMC: Automatic object segmentation using Constrained Parametric Min-Cuts. *TPAMI*, 34(7):1312–1328, 2012. [1](#), [2](#), [5](#)
- [7] 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. [1](#)
- [8] P. Dollár and C. Zitnick. Structured forests for fast edge detection. *ICCV*, 2013. [2](#)
- [9] I. Endres and D. Hoiem. Category independent object proposals. In *ECCV*, pages 575–588, 2010. [1](#), [2](#)
- [10] I. Endres and D. Hoiem. Category-independent object proposals with diverse ranking. *TPAMI*, 36(2):222–234, 2014. [1](#), [2](#), [5](#)
- [11] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. <http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html>. [1](#), [2](#)
- [12] B. Hariharan, P. Arbelaez, L. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. In *ICCV*, 2011. [1](#), [2](#)
- [13] D. Hoiem, Y. Chodpathumwan, and Q. Dai. Diagnosing error in object detectors. In *ECCV*, 2012. [2](#)
- [14] J. Hosang, R. Benenson, P. Dollár, and B. Schiele. What makes for effective detection proposals? In *arXiv:1502.05082*, 2015. [2](#), [4](#)
- [15] A. Humayun, F. Li, and J. M. Rehg. RIGOR: Recycling Inference in Graph Cuts for generating Object Regions. In *CVPR*, 2014. [1](#), [2](#), [5](#), [6](#), [8](#)
- [16] C. Z. J. Lim and P. Dollár. Sketch tokens: A learned mid-level representation for contour and object detection. *CVPR*, 2013. [2](#)
- [17] J. Kim and K. Grauman. Shape sharing for object segmentation. In *ECCV*, 2012. [1](#), [2](#), [5](#)
- [18] P. Krähenbühl and V. Koltun. Geodesic object proposals. In *ECCV*, 2014. [1](#), [2](#), [4](#), [5](#), [6](#), [8](#)
- [19] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. Zitnick. Microsoft COCO: Common Objects in Context. In *ECCV*, 2014. [1](#), [2](#)
- [20] S. Manén, M. Guillaumin, and L. Van Gool. Prime Object Proposals with Randomized Prim’s Algorithm. In *ICCV*, 2013. [1](#)
- [21] J. Pont-Tuset, P. Arbeláez, J. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping for image segmentation and object proposal generation. *arXiv:1503.00848*, March 2015. [2](#), [5](#), [6](#), [8](#)
- [22] P. Rantalankila, J. Kannala, and E. Rahtu. Generating object segmentation proposals using global and local search. In *CVPR*, 2014. [1](#), [2](#), [5](#), [6](#), [8](#)
- [23] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 2015. [1](#)
- [24] A. Torralba and A. A. Efros. Unbiased Look at Dataset Bias. In *CVPR*, 2011. [2](#)
- [25] 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):154–171, 2013. [1](#), [2](#), [5](#), [6](#), [8](#)
- [26] M. Van den Bergh, X. Boix, G. Roig, and L. Van Gool. SEEDS: Superpixels Extracted via Energy-Driven Sampling. *IJCV*, 111(3):298–314, 2015. [7](#)
- [27] P. Viola and M. Jones. Robust real-time face detection. *IJCV*, 57(2), 2004. [1](#)
- [28] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, 2014. [1](#)