

# Weakly Supervised Semantic Labelling and Instance Segmentation

Anna Khoreva<sup>1</sup> Rodrigo Benenson<sup>1</sup> Jan Hosang<sup>1</sup> Matthias Hein<sup>2</sup> Bernt Schiele<sup>1</sup>

<sup>1</sup>Max Planck Institute for Informatics, Saarbrücken, Germany

<sup>2</sup>Saarland University, Saarbrücken, Germany

**Abstract** Semantic labelling and instance segmentation are two tasks that require particularly costly annotations. Starting from weak supervision in the form of bounding box detection annotations, we propose to recursively train a convnet such that outputs are improved after each iteration. We explore which aspects affect the recursive training, and which is the most suitable box-guided segmentation to use as initialisation. Our results improve significantly over previously reported ones, even when using rectangles as rough initialisation. Overall, our weak supervision approach reaches  $\sim 95\%$  of the quality of the fully supervised model, both for semantic labelling and instance segmentation.

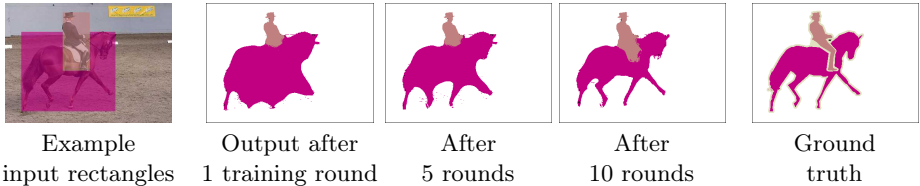


Figure 1: Example results of using only rectangle segments and recursive training (using convnet predictions as supervision for the next round), see Section 3.2.

## 1 Introduction

Convolutional networks (convnets) have become the de facto technique for pattern recognition problems in computer vision. One of their main strengths is the ability to leverage extensive amount of training data to reach top quality. However, one of their main weaknesses is that they need a large number of training samples to reach top quality. This is usually mitigated by using pre-trained models (e.g. with  $\sim 10^6$  training samples for ImageNet classification [1]), but still thousands of samples are needed to shift from the pre-training domain to the application domain. Applications such as semantic labelling (associating each image pixel to a given class) or instance segmentation (grouping all pixels belonging to the same object instance) are very expensive to annotate, and thus significant cost is involved in creating large enough training sets.

Compared to object bounding box annotations pixel-wise mask annotations are far more expansive, requiring  $\sim 15\times$  more time [2]. Cheaper and easier to define, box annotations are currently more pervasive than pixel-wise annotations.

In principle, a large number of box annotations (and images representing the background class) should convey enough information to understand which part of the box content is foreground and which is background. In this paper we explore how much one can close the gap between training a convnet using full supervision for semantic labelling (or instance segmentation) versus using only bounding box annotations.

Our experiments focus on the 20 Pascal classes and show that using only bounding box annotations over the same training set we can reach 95% of the accuracy achievable with full supervision. We show top results for (bounding box) weakly supervised semantic labelling and to the best of our knowledge for the first time report results for weakly supervised instance segmentation.

We view the problem of weak supervision as an issue of input label noise. We explore recursive training as a de-noising strategy, where convnet predictions of the previous training round are used as supervision for the next round. While recursive training on low quality inputs leads to even worse results (drift), we show that bounding box annotations contain sufficient information to avoid drift and obtain significant improvements. We also show that “classic computer vision” techniques for box-guided instance segmentation are a source of surprisingly effective supervision for convnet training.

In summary, our main contributions are:

- We explore various factors that influence the quality of recursive training of convnets for semantic labelling (Sections 3.2 and 3.3).
- As a side effect of our exploration, we report a comparison of multiple GrabCut-like algorithms on 3 400 Pascal VOC12 boxes (instead of the common 50 images from [3]) (Section 3.4).
- We report the best known results when training using bounding boxes only, both on Pascal VOC12 only and with external COCO data, reaching comparable quality with the fully supervised regime (Section 3.5).
- We are the first to show that similar results can be achieved for the weakly supervised instance segmentation task (Section 4).

## 2 Related work

**Semantic labelling.** Semantic labelling may be tackled via decision forests [4] or classifiers over hand-crafted superpixel features [5]. However, convnets have proven particularly effective for semantic labelling. A flurry of variants have been proposed recently [6,7,8,9,10,11,12]. In this work we use DeepLab [8] as our reference implementation. This network achieves state-of-the-art performance on the Pascal VOC12 semantic segmentation benchmark and the source code is available online.

Almost all these methods include a post-processing step to enforce a spatial continuity prior in the predicted segments, which provides a non-negligible improvement on the results (2~5 points). The most popular technique is DenseCRF [13], but other variants are also considered [14,15].

**Weakly supervised semantic labelling.** In order to keep annotation cost low, recent work has explored different forms of supervision for semantic labelling: image labels [16,17,18,19,20], points [21], scribbles [22], and bounding boxes [23,18]. [23,18,24] also consider the case where a fraction of images are fully supervised. [22] proposes a framework to handle all these types of annotations.

In this work we focus on box level annotations for semantic labelling of objects. The closest related work are thus [23,18]. [23] proposes a recursive training schema, where the convnet is trained under supervision of segment object proposals and the updated network in turn improves the segments used for training. [18] proposes an expectation-maximisation algorithm with a bias to enable the network to estimate the foreground regions. We compare to these works in the result sections. Since all implementations use slightly different networks and training procedures, care should be taken during comparison. Compared to [23] our method relies more directly on recursive training (see Section 3). And compared to [18] we consider longer training, explore the effect of recursion, and investigate higher quality input segments (see Section 3.4).

**Instance segmentation.** In contrast to instance agnostic semantic labelling that groups pixels by object class, instance segmentation groups pixels by object instance and ignores classes.

Object proposals [25,26] that generate segments (such as [27,28]) can be used for instance segmentation. Similarly, given a bounding box (e.g. selected by a detector), GrabCut [3] variants can be used to obtain an instance segmentation (e.g. [29,30,31,32,33]).

To enable end-to-end training of detection and segmentation systems, it has been recently proposed to train convnets for the task of instance segmentation [34,35]. In this work we explore weakly supervised training of an instance segmentation convnet. We use our re-implementation of DeepMask [35]. This network achieves state-of-the-art performance on the segmentation proposal task.

### 3 Semantic labelling

Starting from the observation that convnets are quite robust to label noise, our strategy for weakly supervised semantic labelling is to interpret weak supervision as very noisy segment annotations. We iteratively remove part of the label noise by using recursive training over a fixed convnet architecture. In the first round we train a model to convergence using noisy initial inputs (rectangles in Section 3.2, noisy segments in 3.5). We apply the trained model from round  $t - 1$  over the *training* data, de-noise the network’s predictions (see Section 3.2) and use the result as inputs for training model  $t$ .

Note that such recursive training can be thought of as an “across class GrabCut”, where the convnet serves as the appearance model (and the de-noising between rounds replaces the repeated graphcut inference). A similar recursive training strategy is used in [23] with two key differences: First, our de-noising

approach is simpler (does not relies on proposals) and leads to better quality. Second, our de-noising is applied after training until convergence (after each training round), instead of after each epoch as in [23]. The related work of [18] did not consider any recursive training strategy, and we explore different types of initial inputs (in Section 3.4).

### 3.1 Experimental setup

*Datasets.* In this work we evaluate the proposed methods on the Pascal VOC12 segmentation benchmark [36]. The dataset consists of 20 foreground object classes and one background class. The segmentation part of the VOC12 dataset contains 1 464 training, 1 449 validation and 1 456 test images. Following the previous work [8,23] we extend the training set with the annotations provided by [37], resulting in an augmented set of 10 582 training images. Our models are trained using only the training set (even when evaluating over the test set).

In some of our experiments, we use additional training images from the COCO [2] dataset. The dataset provides semantic segmentation masks for 80 object classes. We only consider images that contain any of the 20 Pascal classes, and (following [10]) only objects with bounding box area larger than 200 pixels. After this filtering 99 310 images remain (from training and validation sets), that are added to our training set. When using COCO data, we first pre-train on COCO and then fine-tune over the Pascal VOC12 training set.

All of the COCO and Pascal training images come with semantic labelling annotations (for fully supervised case), and bounding box annotations (for weakly supervised case).

*Evaluation.* We use the “comp6” evaluation protocol. The performance is measured in terms of pixel intersection-over-union averaged across 21 classes (mIoU). Most of our results are shown on the validation set, which we use to guide some of our design choices. Final results are reported on the test set (via the evaluation server) and compared with other state-of-the-art methods.

*Implementation details.* For all our experiments we use the DeepLab-LargeFOV network, using the same train and test parameters as [8]. The model is initialized from a VGG16 network pre-trained on ImageNet [38]. We use a mini-batch of 30 images for SGD and initial learning rate of 0.001 which is divided by 10 after a 2k/20k iterations (for Pascal/COCO). At test time we apply DenseCRF [13].

Note that multiple strategies have been considered to boost test time results, such as multi-resolution or model ensembles [8,11]. Here we keep the approach simple and fixed. In all our experiments we use a fixed training strategy, as well as a fix test time procedure. Across experiments we only change the input training data that the networks gets to see.

### 3.2 Training from rectangles

We start our exploration using a naive strategy. We train a convnet directly on the full extend of bounding box annotations as foreground labels (when two

boxes overlap the smaller one is assumed to be on top). Using this supervision and directly applying recursive training leads to significant degradation of the segmentation output quality, diverging far from the performance of the fully-supervised case, see Figure 2.

*Box enforcing.* We improve the recursive training by de-noising the convnet outputs over the training images to improve labels for the next training round. First, we notice that on the training set the bounding box annotations are available. Any outputs of network outside the boxes are unequivocal errors and thus are set back to the background label. Similarly, since the class annotation is available per box, the probability maps for other classes are set to zero inside each box. Thus inside a box, the model can output either background or the corresponding foreground class (classes when handling overlapping boxes).

*Outliers reset.* Second, we notice that the network has a tendency to generate severe mistakes. Some annotated boxes will be completely classified as background, or be flooded with a single class. We apply two rules to detect these outlier situations. Rule one, if the IoU between the bounding boxes of the input segment and the annotation is  $< 50\%$ , the segment is marked as outlier. Rule two only affects box-guided segmentations (discussed in Section 3.4). If the IoU between the segment mask and the annotation rectangle mask is  $> 98\%$ , the segment is marked as outlier. An outlier segment is reset to the annotation box initial input (a full rectangle in this case, but different in the next sections).

*CRF.* As it is common practice among semantic labelling methods, we filter the output of the network to better respect the image boundaries. We use DenseCRF [13] with the DeepLab parameters [8]. In our weakly supervised scenario, boundary-aware filtering is particularly useful to drive the different iterations forward. We also experimented with more sophisticated boundary-aware smoothing such as [11, section 5], [39] and [15] without observing a noticeable improvement.

**Results.** In Figure 2 all results use a CRF over the validation set output, but not all of them include it when applied over the training set. We see that the naive recursive training is ineffectual. However as soon as some constraints (box enforcing and outliers reset) are enforced, the quality improves dramatically after the first round of recursive training. These results already improve over previous work considering rectangles only input [23,18] (both using a similar convnet to



Figure 2: Recursive training from rectangles only as input. Validation set results. All methods use only rectangles as initial input, except the fully supervised case and “previous best (any input)“.

ours) and achieve 3 points improvement over [18] (from 52.5 to 55.6 mIoU, see Figure 2 “Box enf.+Outliers reset (no CRF)”).

Even more, when also adding CRF filtering over the training set, we see a steady grow after each round, stabilizing around 61% mIoU. This number is surprisingly close to the best results obtained using more sophisticated techniques [23], which achieve around 62% mIoU (see Figure 2 and Table 2). The improvement of the segmentation across training rounds is shown in Figure 1.

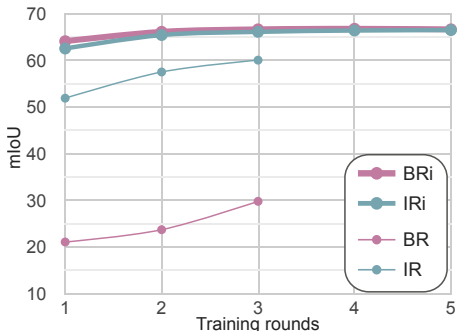
*Conclusion.* Our results indicate that recursive training of a convnet is indeed robust to input noise as soon as appropriate care is taken to de-noise the output between rounds. We use the described de-noising procedure in all following experiments.

### 3.3 Oracle cases

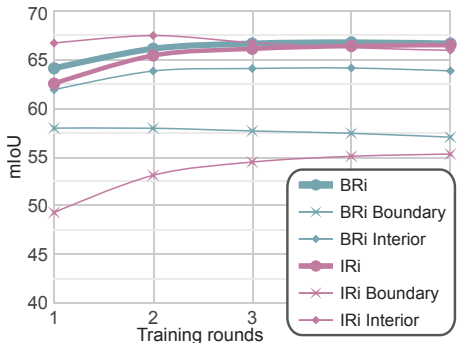
#### Boundary vs interior regions.

Section 3.2 shows that recursive training can be robust to noise. One can wonder which region is most important to have noise-free, the object boundary or the interior regions? On the one hand, the hardest part of semantic labelling might be having good boundaries, and the CRF smoothing propagates these towards the interior regions. On the other hand, labelling the bulk of an object might be the hard task (detecting), and propagation towards the border is handled during smoothing.

We analyse this question using an oracle case, where altered ground truth segmentation annotations are employed as training data. BR/IR indicates that the boundary/interior region is kept, and the rest is set as background. We adjust the boundary thickness to have similar boundary/interior areas.



(a) Ignore region matters.



(b) Recursive training improves the lacking domain.

Figure 3: Oracle cases, boundary vs interior region. BR/IR denotes training with just boundary/interior region. BRi/IRi denotes training with boundary/interior region and setting the rest of the ground truth mask to ignore label instead of background.

*Results.* Figure 3a shows that if only boundary/interior regions are provided as foreground label (and everything else background), the results are quite poor. If we introduce an ignore label performance increases in both cases significantly

(BRi/IRi sets the boundary/interior region to foreground and other one to ignore label). Note that neither ends up reaching the same performance as the fully supervised case ( $\sim 70\%$  mIoU, see Figure 2), indicating that both interior and boundary regions are needed to reach full quality.

Figure 3b evaluates the performance on the ground truth segmentation, and on the boundary and interior regions separately. One can see that the model trained on boundary/interior region starts with a better performance on this region, and as the training rounds evolve it progressively improves its lacking domain.

*Conclusion.* From Figure 3 we conclude that neither boundaries nor interior regions are critical for the recursive training procedure. However, it seems to be important to avoid systematic mistakes (since the network will learn to reproduce them), and using ignore regions is an effective way to mitigate these.

**Quality vs recall.** As a second set of oracle experiment we evaluate the effect of selecting a subset of the training data based on its quality. We filter out the input samples of two segmentation methods (MCG and GrabCut+, described in Section 3.4), based on their IoU with the ground truth segmentation annotations ( $\text{IoU} > \tau$ ,  $\tau \in \{0.9, 0.7, 0.5\}$ ). The higher the IoU threshold  $\tau$  the more boxes get replaced with ignore label (full rectangle). This allows to explore the trade-off between quality and recall of training data.

*Results.* Figure 4 shows validation set results after two rounds of recursive training. Compared to using all the data from GrabCut+, using higher quality data ( $\text{IoU} > 0.7$ ,  $\sim 65\%$  of the training data remains non-ignore) provides a noticeable gain in performance. When the recall drops below 25% ( $\text{IoU} > 0.9$ ), using fewer but better data starts hurting more than it helps. MCG shows a similar trend.

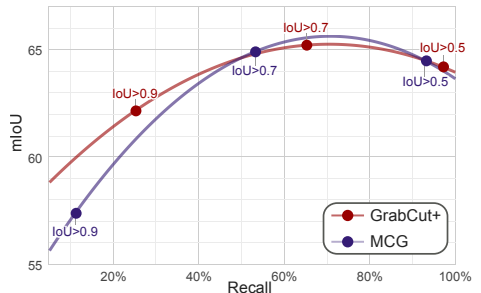


Figure 4: Oracle cases, quality versus recall trade-off.

*Conclusion.* From the oracle results, we see that there is a clear trade-off between quality and recall. The sweet-spot seems to be high quality with only a small drop in recall.

### 3.4 Box segmentation

Based on previous observations instead of directly using bounding boxes as input (Section 3.2) we propose to employ box-guided instance segmentation to increase quality of the input data. Our goal is to have weak annotations with maximal quality and minimal loss in recall. In Section 3.2 we explored how far could we get with just using boxes as foreground labels. However, to obtain results of higher quality several rounds of recursive training are needed. Starting from less

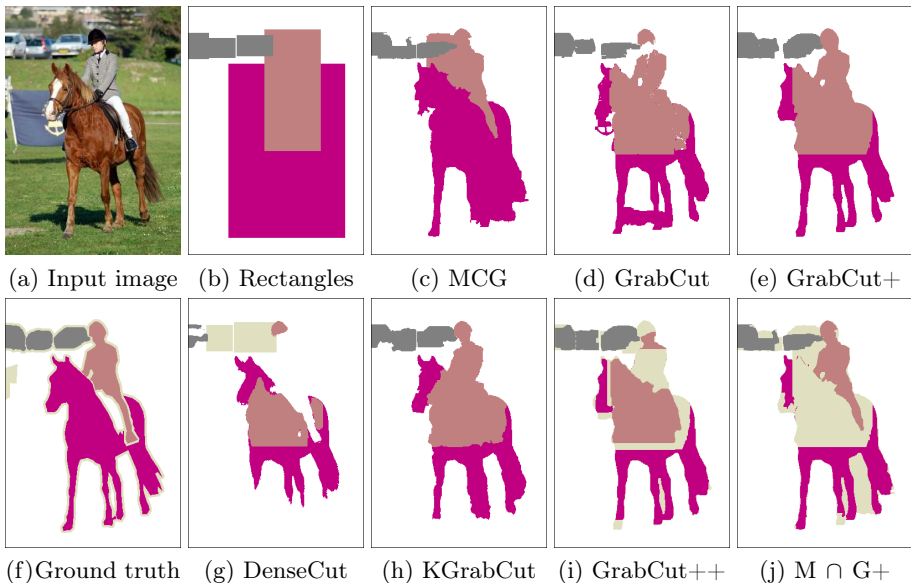


Figure 5: Example of the different segmentations obtained starting from a bounding box annotation. Grey/pink/magenta indicate different object classes, white is background, and ignore regions are beige.  $M \cap G+$  denotes  $MCG \cap \text{Grabcut}+$ .

noisier object segments we would like to reach better performance with fewer training rounds.

For this purpose we explore different GrabCut-like [3] techniques. In order to avoid systematic mistakes of one particular method and reduce the noise in the produced annotations we propose to use only the consensus between segments given by different techniques. Pixels where two methods agree are set to foreground (of the appropriate class) and, following Section 3.3, pixels with disagreement are set to ignore label.

Figure 5 presents examples of the considered weak annotations, the corresponding quantitative results are in Table 1. For evaluation we use the mean IoU measure (see Section 3.1) as well as pixel accuracy and recall inside bounding boxes. Previous work evaluated using the 50 images from the GrabCut dataset [3], or 1k images with one salient object [30]. The evaluation of Table 1 compares multiple methods over 3.4k object windows, where the objects are not salient, have diverse sizes and occlusions level. This is a more challenging scenario than usually considered for GrabCut-like methods.

*Baselines.* As a first baseline we simply fill-in the bounding box annotations with their class annotation. This is the **Rectangle** method shown in Figure 5b. This approach has 100% recall but also contains many false positive samples. Inspired by [18] and Section 3.3, we consider using the 20% centre area of each annotation and set the rest of the box to ignore. This 20% **Bbox** approach reaches



Method		No outliers reset			With outlier reset		
		mIoU	Accuracy	Recall	mIoU	Accuracy	Recall
Oracle cases	GrabCut with HED(GT)	76.0	84.9	100	79.3	87.0	93
	Best MCG [27]	80.8	89.5	100	85.4	91.4	94
Baselines	Rectangle	62.2	64.6	100	-	-	-
	MCG [27]	66.6	78.2	100	68.5	78.7	97
	20% Bbox	(83.5)	(87.1)	16	-	-	-
GrabCut variants	DenseCut [30]	52.5	69.3	100	63.2	75.9	71
	20% Bbox-Seg+CRF [18]	71.1	81.0	100	71.8	81.4	98
	GrabCut [3]	72.9	81.9	100	74.9	83.6	95
	KGrabCut [32]	73.5	82.3	100	74.8	83.3	96
	GrabCut+	75.2	83.7	100	78.6	86.2	93
Recall vs accuracy trade	GrabCut++	(78.3)	(86.9)	86	81.7	88.9	76
	KGrabcut $\cap$ Grabcut+	(80.9)	(88.3)	87	80.8	87.8	90
	MCG $\cap$ Grabcut+	(84.1)	(91.2)	76	<b>84.7</b>	<b>91.2</b>	<b>78</b>

Table 1: GrabCut variants, evaluated on Pascal VOC12 validation set. (·) denotes evaluation only on non-ignore labels, thus not directly comparable to other numbers. See Section 3.4 for details.

higher accuracy but lower recall than Rectangle.

Our final baseline MCG (inspired by [23]) uses the MCG [27] object mask proposals. The segment with highest box IoU overlap with the bounding box annotation is used, see Figure 5c. Using MCG provides higher accuracy than Rectangle.

*GrabCut variants.* **GrabCut** [3] is the established technique to estimate an object segment from its bounding box (see Figure 5d). It provides good accuracy compared to the baselines. To further improve its quality we propose to use better pairwise terms. Instead of the typical RGB colour difference the pairwise terms in **GrabCut+** are replaced by probability of boundary as generated by HED [40]. The HED boundary detector is trained on the generic boundaries of BSDS500 [41]. Moving from GrabCut to GrabCut+ brings a 2 points improvement in accuracy, see Figure 5e.

We also experimented with other variants such as **DenseCut** [30] (Figure 5g) and **KGrabCut** [32] (Figure 5h) but did not obtain significant gains.

[18] proposed to perform foreground/background segmentation by using DenseCRF and the 20% of the centre area of the bounding box as foreground prior. This approach is denoted **20% Bbox-Seg+CRF** in Table 1 and under-performs compared to GrabCut.

Table 1 also reports all methods after applying the outlier reset strategy described in Section 3.2. We can see that the outlier reset provides a slim but consistent improvement across all box segmentation methods.

*Oracle cases* To serve as reference point, we evaluate two oracle cases. MCG segment selected based on the best overlap with the ground truth segment, and GrabCut with pairwise terms based on boundaries from the HED detector trained on ground truth boundaries of Pascal VOC12.

*Consensus of segments.* To mitigate noise in the input data we explore consensus between segments. We generate  $\sim 150$  GrabCut+ segments per bounding box by jittering the box coordinates ( $\pm 5\%$ ) as well as the size of the outer background region considered (from 10% to 60%). If the agreement between the segments is higher than 70% the pixel is set to the box object class, if the agreement is less than 20% it is set to background class, otherwise it is marked as ignore. We denote this consensus variant **GrabCut++**, see Figure 5i. This approach results in higher accuracy with a small drop in recall, see Table 1.

We also explore consensus between segments given by different methods,  $\text{MCG} \cap \text{Grabcut+}$  and  $\text{KGrabcut} \cap \text{Grabcut+}$ . The intersecting pixels are set to the box object class, and the rest of the union to ignore label.  $\text{MCG} \cap \text{Grabcut+}$  shows the best accuracy overall while having reasonable recall (given the results of Figure 4). See Figure 5j.

Based on these results for the next sections we consider  $\text{MCG} \cap \text{Grabcut+}$  as the most promising candidate to obtain top segmentation quality from weak supervision.

### 3.5 Training from segments

Based on the results of Section 3.3 and 3.4, we now explore recursive training segments as input instead of full rectangles (Section 3.2). Unless otherwise specified all experiments are done using the same setup as in Section 3.2. The parameters between training rounds are all identical.

Our main results are shown in Figure 6 and Table 2 (top part). These are results obtained using only Pascal VOC12 box annotations ( $\sim 10k$  images). We also show in Table 2 (bottom part) and Table 3 results when considering additional bounding boxes from the COCO dataset ( $\sim 100k$  additional images), or when using a fraction of Pascal semantic labelling annotations (akin the fully supervised case). We evaluate the same scenarios considered by [23,18].

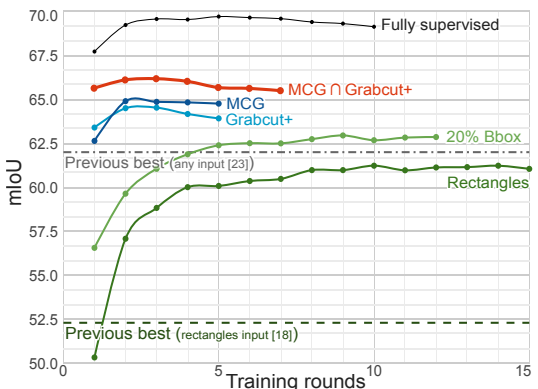


Figure 6: Segmentation quality versus training round for different input types. Validation set results. All methods only use Pascal detection bounding box annotations as input. See also Table 2.

*Analysis.* The results of Figure 6, obtained using only bounding boxes from Pascal VOC12, raise two main observations. First, we see that the “20% Bbox” experiment, based on the insights from Sections 3.2 and 3.3, shows surprisingly good performance. After a long enough number of training rounds this setup alone improves over the best previously reported results on this task. Second, we see that when combining  $\text{MCG} \cap \text{Grabcut+}$  to provide an input with high

Training data		Method	mIoU val. set	mIoU test set
None	Oracle cases	Rectangles (GT Bbox)	62.2	-
		Rectangle (Fast-RCNN)	44.3	-
VOC12	Related work	Pathak et al. - tags and sizes [17]	42.4	45.1
		Bearman et al. - rectangles [21]	45.1	-
		BoxSup - rectangles [23]	52.3	-
		WSSL - rectangles [18]	52.5	54.2
		WSSL - 20% Bbox-Seg+CRF [18]	60.6	62.2
		BoxSup - MCG [23]	62.0	64.6
	Our weakly supervised	Rectangle	61.1	62.2
		20% Bbox	62.7	63.5
		GrabCut+	63.9	-
		MCG	64.8	-
		MCG $\cap$ Grabcut+	<b>65.7</b>	<b>67.5</b>
	Fully supervised	BoxSup [23]	63.8	-
		WSSL [18]	67.6	70.3
		Ours (DeepLab-LargeFOV [8])	<u>69.1</u>	<u>70.5</u>
VOC12 + COCO	Our weakly supervised	20% Bbox	65.3	66.7
		MCG $\cap$ Grabcut+	<b>68.9</b>	<b>69.9</b>
	Fully supervised	BoxSup [23]	68.1	-
		WSSL [18]	71.7	73
		Ours (DeepLab-LargeFOV [8])	<u>72.3</u>	<u>73.2</u>

Table 2: Semantic labelling results for methods trained using Pascal VOC12 bounding boxes alone. Validation set results. Underline indicates full supervision baselines, and bold are our best weakly supervised results.

accuracy (and  $\sim 80\%$  recall), the recursive training does not bring a consistent improvement. With this input already the first round of training reaches an accuracy that is only 3 mIoU percent points below training with full supervision and gives a significant improvement over training with rectangle inputs. Starting from less noisier input data allows to reach better performance with fewer training rounds. Notice that, coincidentally, the results of training with MCG  $\cap$  Grabcut+ are comparable to the oracle case considered in Section 3.3. As shown in Table 3 training with 10% of Pascal VOC12 semantic labelling annotations does not bring any gain to the performance. This result indicates the high quality of the proposed weak annotations.

As shown in Tables 2 (bottom part) and 3 adding more training data allows to further improve results. Using more boxes (from COCO) allows to reach the performance of training with Pascal VOC12 ground truth segmentation annotations. And adding boxes over ground truth data helps to improve over the full supervision with Pascal VOC12 by 1.5 mIoU percent points.

*Conclusion.* Our results show that good quality can be reached when the proper balance between accuracy and recall is made in the input. Using long enough recursive training 90% of the full supervision results can be obtained when feeding

Training data	Super-vision	#GT images	#Weak images	Method	mIoU val. set	mIoU test set
VOC12	weak	-	V 10k	MCG $\cap$ Grabcut+	<b>65.7</b>	<b>67.5</b>
	semi	V 1.4k	V 9k	WSSL - rectangles [18]	62.1	-
				BoxSup - MCG [23]	63.5	66.2
				WSSL - 20% Bbox-Seg+CRF [18]	65.1	66.6
				MCG $\cap$ Grabcut+	<b>65.8</b>	<b>66.9</b>
	full	V 10k	-	BoxSup [23]	63.8	-
				WSSL [18]	67.6	70.3
				Ours (DeepLab-LargeFOV [8])	<u>69.1</u>	<u>70.5</u>
VOC12 + COCO	weak	-	V + C 110k	MCG $\cap$ Grabcut+	68.9	69.9
	semi	V 10k	C 123k C 110k	BoxSup - MCG [23]	68.2	71.0
				MCG $\cap$ Grabcut+	<b>71.6</b>	<b>72.8</b>
	full	V + C 133k	-	BoxSup [23]	68.1	-
				WSSL [18]	71.7	73
				Ours (DeepLab-LargeFOV [8])	<u>72.3</u>	<u>73.2</u>

Table 3: Semantic labelling results for validation and test set. Our methods only use bounding boxes (from Pascal VOC12 and COCO). Underline indicates full supervision baselines, and bold are our best weakly- and semi-supervised results.

simple rectangles. When higher quality segments are used as input, one training round is enough to reach 95% of the fully-supervised training quality.

## 4 Instance segmentation

Complementing the experiments of Section 3, we also explored a second task: weakly supervised instance segmentation. To the best of our knowledge, these are the first reported experiments on this task.

As object detection moves forward, there is a need to provide richer output than a simple bounding box around objects. Recently [34,35] explored training convnets to output a foreground versus background segmentation of an instance inside a given bounding box. Such networks are trained using pixel-wise annotations that distinguish between instances. These annotations are more detailed/expensive than semantic labelling (for the same object classes), and thus there is strong interest in weakly supervised training.

The segments used for the training, as discussed in Section 3.5, are generated starting from individual object bounding boxes. Each segment represents a different object instance and thus can be used directly to train an instance segmentation convnet.

### 4.1 Experimental setup

Our reference network is a re-implementation of DeepMask [35] built on top of Fast-RCNN [42]. In validation experiments our implementation shows comparable results (slightly worse) to the original publications. Since we use the same network in all comparisons, we consider this a fair setup.

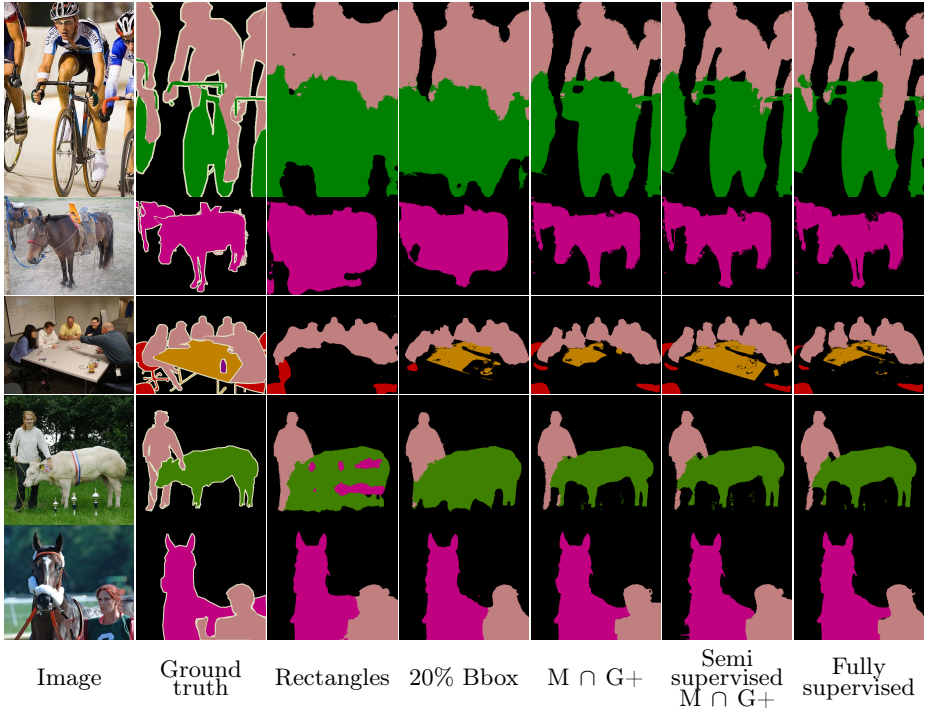


Figure 7: Qualitative results on VOC12.  $M \cap G+$  denotes the weakly supervised model trained on  $MCG \cap Grabcut+$ .

DeepMask operates by directly regressing an instance segmentation of the central object inside its square input field. As such it only considers location and scale, but ignores aspect ratio information. We compare all methods on the basis of a fixed set of Fast-RCNN detections per image (using MCG proposals [27]). These are treated as object proposal boxes. The baselines have access to the Fast-RCNN rectangle shaped proposals, while DeepMask only uses the squared box, thus having a slightly more difficult task (this setup is similar to the sliding window model used in [35]).

For training we use the augmented sets of images from Pascal VOC12 [37,36] and COCO datasets [2], the same subset used for experiments in Section 3.2. After training we perform a post-processing step, applying the same DenseCRF as for DeepLab-LargeFOV model (see Section 3.1). We constrain the pixels inside the DeepMask output segment to be foreground and set appropriately the unary terms of DenseCRF. This approach is denoted as DeepMask-CRF.

For evaluation we use the annotations from the validation set of Pascal VOC12 [36] as in Section 3.1. These are detailed annotations that also include instance segmentation information. We want to decouple the instance segmentation task from the object localisation task and thus evaluate instance segmentation using the “average best overlap” metric (ABO) [25].

Method		Training data	Average best overlap
Baselines	Rectangle	-	38.5
	Ellipse	-	41.7
	MCG	-	44.7
	GrabCut	-	45.8
	GrabCut+	-	46.4
Weakly supervised	DeepMask-CRF	V 10k	49.6
		V+C 110k	<b>52.4</b>
Fully supervised	DeepMask-CRF	V 10k	50.9
		V+C 110k	53.3

Table 4: Instance segmentation results on Pascal VOC12 validation set. See 4.1 for details of the evaluation setup. Weakly supervised DeepMask-CRF trained with GrabCut+ reaches comparable results to full supervision.



Figure 8: Example result from our DeepMask-CRF model trained with Pascal VOC12 and COCO weak supervision.

## 4.2 Results

We consider five training-free baselines. Simply filling in the proposal rectangles (boxes) with foreground label, fitting an ellipse inside the box, using the MCG proposal with best bounding box IoU, and using GrabCut or GrabCut+ as described in Section 3.4, initialized from the proposal box. In these experiments only one training round is used (no recursive training).

*Analysis.* In Table 4 we see that GrabCut+ performs rather well on this task: with 46.4 ABO it is only 4 ABO points behind the fully supervised DeepMask-CRF model trained on Pascal VOC12. In contrast to GrabCut+, DeepMask-CRF is trained with and benefits from additional training data. When moving from 10k training images (Pascal VOC12 only) to 110k (Pascal VOC12 and COCO) the weakly supervised DeepMask-CRF performance improves from 49.6 to 52.4 ABO. Compared to the fully supervised variant, this is a slightly bigger improvement, suggesting that the benefit from more accurate annotations diminish while increasing amount of training data. On both training sets the weakly supervised model performs almost as good as the fully supervised model, lacking 1 ABO point behind. This highlights the quality of weakly supervised segments given by GrabCut+ and the robustness to label noise of convnets.

An example of the segmentation result from weakly supervised DeepMask-CRF (V+C) is shown in Figure 8. Additional example results are presented in the supplementary material.

*Conclusion.* Overall we see a similar trend for instance segmentation as for semantic labelling. From only bounding boxes, we generate good enough noisy input segments and are able to train a model that reaches more than 95% of the quality of fully-supervised case.

## 5 Conclusion

The series of experiments presented in this paper provide new insights on how to train pixel-labelling convnets from bounding box annotations only. We discussed which ingredients make recursive training tick, and the importance of a good balance between accuracy and recall in the noisy training data. Our results improve over previously reported ones on the semantic labelling task and reach  $\sim 95\%$  of the quality of the same network trained on the ground truth segmentation annotations (over the same data). We also report the first results for weakly supervised instance segmentation, where we also reach  $95\%+$  of the quality of the fully-supervised training.

Our current approach exploits existing GrabCut variants. In future work we would like to exploit the class labels of the annotations (closer to co-segmentation), and consider even weaker forms of supervision.

## References

1. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. *IJCV* (2015)
2. Lin, T., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: *ECCV*. (2014)
3. Rother, C., Kolmogorov, V., Blake, A.: Grabcut: Interactive foreground extraction using iterated graph cuts. In: *ACM Trans. Graphics*. (2004)
4. Shotton, J., Winn, J., Rother, C., Criminisi, A.: Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *IJCV* (2009)
5. Gould, S., Fulton, R., Koller, D.: Decomposing a scene into geometric and semantically consistent regions. In: *ICCV*. (2009)
6. Pinheiro, P.O., Collobert, R.: Recurrent convolutional neural networks for scene labeling. In: *ICML*. (2014)
7. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: *CVPR*. (2015)
8. Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.: Semantic image segmentation with deep convolutional nets and fully connected crfs. In: *ICLR*. (2015)
9. Lin, G., Shen, C., van den Hengel, A., Reid, I.: Efficient piecewise training of deep structured models for semantic segmentation. In: *CVPR*. (2016)
10. Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., Torr, P.: Conditional random fields as recurrent neural networks. In: *ICCV*. (2015)
11. Kokkinos, I.: Pushing the boundaries of boundary detection using deep learning. In: *ICLR*. (2016)
12. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. In: *ICLR*. (2016)
13. Krähenbühl, P., Koltun, V.: Efficient inference in fully connected crfs with gaussian edge potentials. In: *NIPS*. (2011)

14. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts?. *PAMI* (2004)
15. Barron, J., Poole, B.: The fast bilateral solver. *arXiv preprint arXiv:1511.03296* (2015)
16. Pathak, D., Shelhamer, E., Long, J., Darrell, T.: Fully convolutional multi-class multiple instance learning. In: *ICLR workshop*. (2015)
17. Pathak, D., Kraehenbuehl, P., Darrell, T.: Constrained convolutional neural networks for weakly supervised segmentation. In: *ICCV*. (2015)
18. Papandreou, G., Chen, L., Murphy, K., Yuille, A.L.: Weakly- and semi-supervised learning of a dcnn for semantic image segmentation. In: *ICCV*. (2015)
19. Pinheiro, P., Collobert, R.: From image-level to pixel-level labeling with convolutional network. In: *CVPR*. (2015)
20. Wei, Y., Liang, X., Chen, Y., Shen, X., Cheng, M.M., Zhao, Y., Yan, S.: Stc: A simple to complex framework for weakly-supervised semantic segmentation. *arXiv preprint arXiv:1509.03150* (2015)
21. Bearman, A., Russakovsky, O., Ferrari, V., Fei-Fei, L.: What’s the point: Semantic segmentation with point supervision. *arXiv preprint arXiv:1506.02106* (2015)
22. Xu, J., Schwing, A., Urtasun, R.: Learning to segment under various forms of weak supervision. In: *CVPR*. (2015)
23. Dai, J., He, K., Sun, J.: Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In: *ICCV*. (2015)
24. Hong, S., Noh, H., Han, B.: Decoupled deep neural network for semi-supervised semantic segmentation. In: *NIPS*. (2015)
25. Pont-Tuset, J., Gool, L.V.: Boosting object proposals: From pascal to coco. In: *ICCV*. (2015)
26. Hosang, J., Benenson, R., Dollár, P., Schiele, B.: What makes for effective detection proposals? *PAMI* (2015)
27. Pont-Tuset, J., Arbeláez, P., Barron, J., Marques, F., Malik, J.: Multiscale combinatorial grouping for image segmentation and object proposal generation. *arXiv preprint arXiv:1503.00848* (2015)
28. Krähenbühl, P., Koltun, V.: Learning to propose objects. In: *CVPR*. (2015)
29. Lempitsky, V., Kohli, P., Rother, C., Sharp, T.: Image segmentation with a bounding box prior. In: *ICCV*. (2009)
30. Cheng, M., Prisacariu, V., Zheng, S., Torr, P., Rother, C.: Densecut: Densely connected crfs for realtime grabcut. *Computer Graphics Forum* (2015)
31. Taniai, T., Matsushita, Y., Naemura, T.: Superdifferential cuts for binary energies. In: *CVPR*. (2015)
32. Tang, M., Ben Ayed, I., Marin, D., Boykov, Y.: Secrets of grabcut and kernel k-means. In: *ICCV*. (2015)
33. Yu, H., Zhou, Y., Qian, H., Xian, M., Lin, Y., Guo, D., Zheng, K., Abdelfatah, K., Wang, S.: Loosecut: Interactive image segmentation with loosely bounded boxes. *arXiv preprint arXiv:1507.03060* (2015)
34. Hariharan, B., Arbeláez, P., Girshick, R., Malik, J.: Hypercolumns for object segmentation and fine-grained localization. In: *CVPR*. (2015)
35. Pinheiro, P.O., Collobert, R., Dollar, P.: Learning to segment object candidates. In: *NIPS*. (2015)
36. Everingham, M., Eslami, S.M.A., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes challenge: A retrospective. *IJCV* (2015)
37. Hariharan, B., Arbeláez, P., Bourdev, L., Maji, S., Malik, J.: Semantic contours from inverse detectors. In: *ICCV*. (2011)



38. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR. (2015)
39. Baque, P., Bagautdinov, T., Fleuret, F., Fua, P.: Principled parallel mean-field inference for discrete random fields. arXiv preprint arXiv:1511.06103 (2015)
40. Xie, S., Tu, Z.: Holistically-nested edge detection. In: ICCV. (2015)
41. Arbeláez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. PAMI (2011)
42. Girshick, R.: Fast R-CNN. In: ICCV. (2015)



# Supplementary material

## A Content

This supplementary material provides additional quantitative and qualitative results:

- Section B provides additional results for box-guided input segmentations used as weak supervision for semantic labelling (Figure 9 and Figure 2).
- Detailed performance of each class for semantic labelling is reported in Section C (Table 5).
- Section D provides additional qualitative results for weakly supervised semantic labelling on Pascal VOC12 (Figure 3).
- Detailed results for instance segmentation of DeepMask-CRF are shown in Section E (Figure 5 and Figure 4).

## B Box-guided input segmentations

In this work instead of directly using bounding boxes as input we propose to employ box-guided instance segmentation to increase quality of the input training data. For this purpose we explore different GrabCut-like techniques [3,27,18,30,32].

Figure 9 shows the evaluation of the considered in the main paper box-guided segmentation methods (without outlier reset). For evaluation we use the pixel accuracy of the foreground segments for each box versus recall. Figure 9 complements Table 1 in the main paper.

In order to avoid mistakes of one particular method and reduce the noise in the produced annotations we propose to use the consensus between segments given by different techniques, such as  $MCG \cap Grabcut+$ . We consider  $MCG \cap Grabcut+$  as the most promising candidate to obtain top segmentation quality from weak supervision.

Figure 2 presents examples of the considered weak annotations. This figure extends Figure 5 of the main paper.

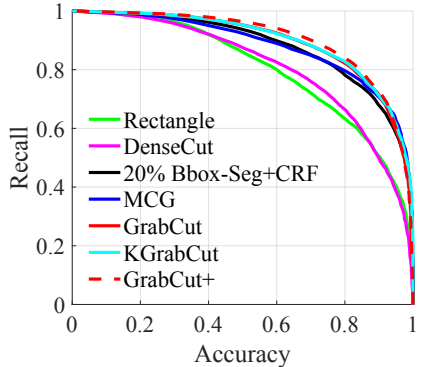


Figure9: Evaluation of box-guided segmentations. Validation set results.

## C Detailed test set results for semantic labelling

In Table 5, we present more detailed results on the Pascal VOC12 test set for the methods reported in the main paper in Table 2 and Table 3.

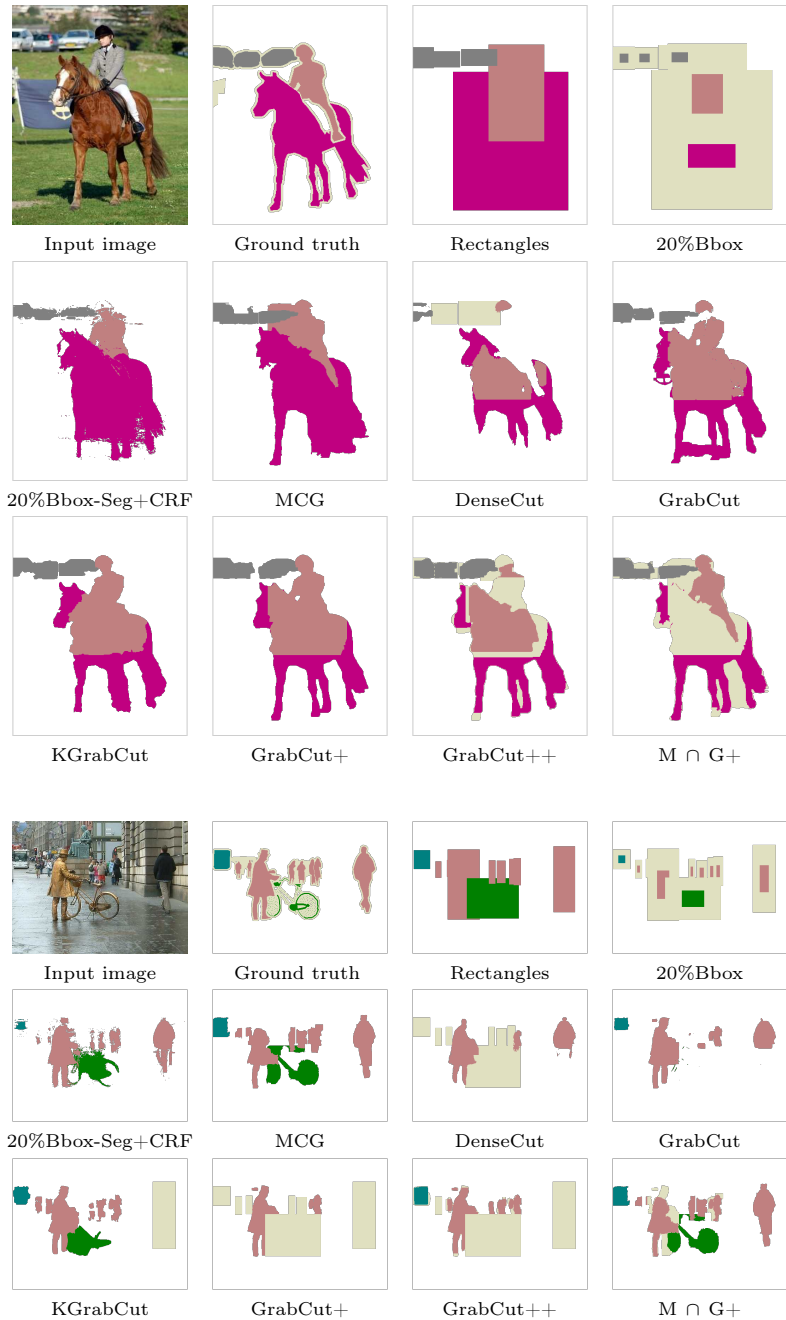


Figure 10: Different segmentations obtained starting from a bounding box. White is background and ignore regions are beige.  $M \cap G+$  denotes  $MCG \cap Grabcut+$ .



Figure 2: Different segmentations obtained starting from a bounding box. White is background and ignore regions are beige.  $M \cap G+$  denotes  $MCG \cap Grabcut+$ .

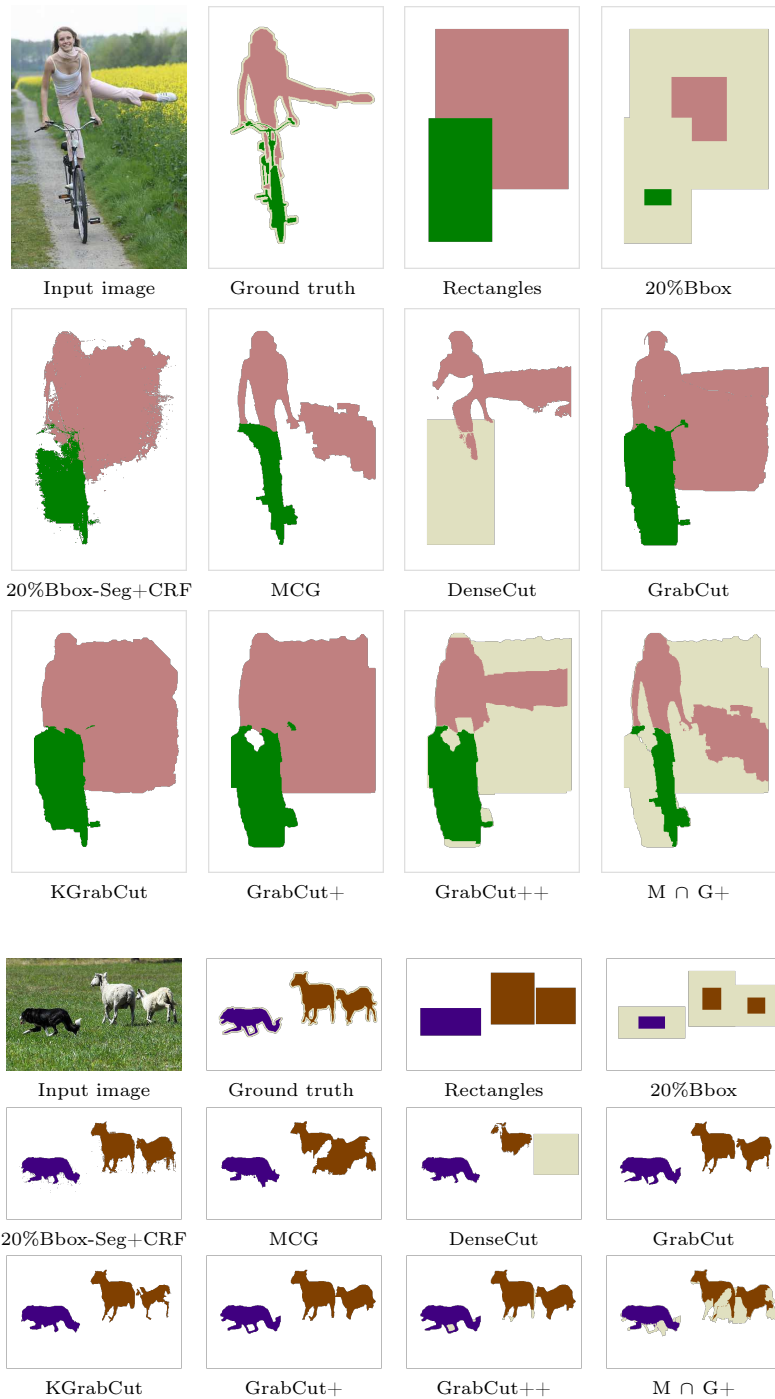


Figure 2: Different segmentations obtained starting from a bounding box. White is background and ignore regions are beige.  $M \cap G+$  denotes  $MCG \cap Grabcut+$ .

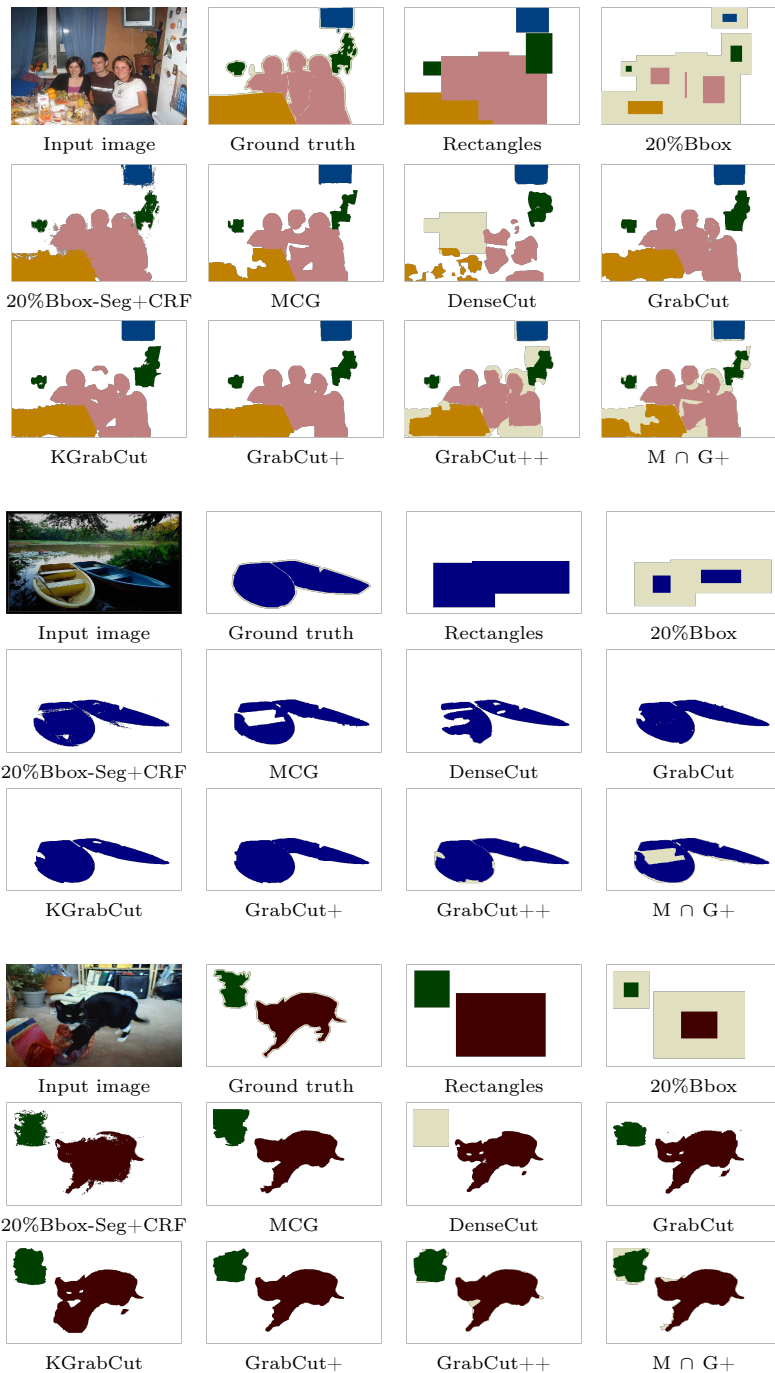


Figure 2: Different segmentations obtained starting from a bounding box. White is background and ignore regions are beige.  $M \cap G+$  denotes  $MCG \cap Grabcut+$ .

Training data	Super-vision	Method	mean	aero plane	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	motor bike	per son	potted plant	sheep	sofa	train	tv
VOC12	weak	WSSL rect.[18]	54.2	43.6	22.5	50.5	45.0	62.5	76.0	66.5	61.2	25.3	55.8	52.1	56.6	48.1	60.1	58.2	49.5	58.3	40.7	62.3	<b>61.1</b>
		WSSL segm.[18]	62.2	64.4	27.3	67.6	55.1	64.0	81.6	70.5	76.0	24.1	63.8	58.2	72.1	59.8	73.5	71.4	47.4	76.0	44.2	68.9	50.9
		BoxSup [23]	64.6	<b>80.3</b>	<b>31.3</b>	<b>82.1</b>	47.4	62.6	75.4	75.0	74.5	24.5	68.3	56.4	73.7	69.4	75.2	75.1	47.4	70.8	45.7	71.1	58.8
		Rectangle	62.2	62.6	24.5	63.7	56.7	<b>68.1</b>	<b>84.3</b>	75.0	72.3	27.2	63.5	61.7	68.2	56.0	70.9	72.8	49.0	66.7	45.2	71.8	58.3
		20% Bbox	63.5	67.7	25.5	67.3	58.0	62.8	83.1	75.1	78.0	25.5	64.7	60.8	74.0	62.9	74.6	73.3	50.0	68.5	43.5	71.6	56.7
		M $\cap$ G+	<b>67.5</b>	78.1	31.1	72.4	<b>61.0</b>	67.2	84.2	<b>78.2</b>	<b>81.7</b>	<b>27.6</b>	<b>68.5</b>	<b>62.1</b>	<b>76.9</b>	<b>70.8</b>	<b>78.0</b>	<b>76.3</b>	<b>51.7</b>	<b>78.3</b>	<b>48.3</b>	<b>74.2</b>	58.6
	semi	BoxSup [23]	66.2	<b>82.0</b>	<b>33.6</b>	74.0	55.8	57.5	81.0	74.6	<b>80.7</b>	27.6	<b>70.9</b>	50.4	71.6	<b>70.8</b>	78.2	<b>76.9</b>	<b>53.5</b>	72.6	<b>50.1</b>	72.3	<b>64.4</b>
		WSSL segm.[18]	66.6	75.3	29.9	74.4	59.8	64.6	<b>84.3</b>	<b>76.2</b>	79.0	27.9	69.1	56.5	73.8	66.7	<b>78.8</b>	76.0	51.8	<b>80.8</b>	47.5	73.6	60.5
	full	M $\cap$ G+	<b>66.9</b>	75.8	32.3	<b>75.9</b>	<b>60.1</b>	<b>65.7</b>	82.9	75.0	79.5	<b>29.5</b>	68.5	<b>60.6</b>	<b>76.2</b>	68.6	76.9	75.2	53.2	76.6	49.5	<b>73.8</b>	58.6
		WSSL [18]	70.3	83.5	36.6	<b>82.5</b>	<b>62.3</b>	66.5	85.4	78.5	<b>83.7</b>	<b>30.4</b>	72.9	<b>60.4</b>	<b>78.5</b>	75.5	82.1	79.7	<b>58.2</b>	<b>82.0</b>	<b>48.8</b>	73.7	63.3
		Ours	<b>70.5</b>	<b>85.3</b>	<b>38.3</b>	79.4	61.4	<b>68.9</b>	<b>86.4</b>	<b>82.1</b>	83.6	30.3	<b>74.5</b>	53.8	78.0	<b>77.0</b>	<b>83.7</b>	<b>81.8</b>	55.6	79.8	45.9	<b>79.3</b>	<b>63.4</b>
VOC12 + COCO	weak	20% Bbox	66.7	69.0	27.5	77.1	<b>61.9</b>	65.3	84.2	75.5	83.2	25.7	73.6	<b>63.6</b>	78.2	69.3	75.3	75.2	51.0	73.5	46.2	74.4	60.4
		M $\cap$ G+	<b>69.9</b>	<b>82.5</b>	<b>33.4</b>	<b>82.5</b>	59.5	<b>65.8</b>	<b>85.3</b>	<b>75.6</b>	<b>86.4</b>	<b>29.3</b>	<b>77.1</b>	60.8	<b>80.7</b>	<b>79.0</b>	<b>80.5</b>	<b>77.6</b>	<b>55.9</b>	<b>78.4</b>	<b>48.6</b>	<b>75.2</b>	<b>61.5</b>
	semi	BoxSup [23]	71.0	86.4	35.5	79.7	65.2	65.2	84.3	78.5	83.7	30.5	76.2	<b>62.6</b>	79.3	76.1	82.1	81.3	57.0	78.2	<b>55.0</b>	72.5	<b>68.1</b>
		M $\cap$ G+	<b>72.8</b>	<b>87.6</b>	<b>37.7</b>	<b>86.7</b>	<b>65.5</b>	<b>67.3</b>	<b>86.8</b>	<b>81.1</b>	<b>88.3</b>	<b>30.7</b>	<b>77.3</b>	61.6	<b>82.7</b>	<b>79.4</b>	<b>84.1</b>	<b>82.0</b>	<b>60.3</b>	<b>84.0</b>	49.4	<b>77.8</b>	64.7
	full	WSSL [18]	73.0	88.5	35.9	88.5	62.3	68.0	87.0	81.0	86.8	<b>32.2</b>	<b>80.8</b>	60.4	81.1	81.1	83.5	81.7	55.1	84.6	<b>57.1</b>	75.7	<b>67.2</b>
		Ours	<b>73.2</b>	<b>88.8</b>	<b>37.3</b>	<b>83.8</b>	<b>66.5</b>	<b>70.1</b>	<b>89.0</b>	<b>81.4</b>	<b>87.3</b>	30.2	78.8	<b>61.6</b>	<b>82.4</b>	<b>82.3</b>	<b>84.4</b>	<b>82.2</b>	<b>59.1</b>	<b>85.0</b>	50.8	<b>79.7</b>	63.8

Table 5: Per class semantic labelling results for methods trained using Pascal VOC12 and COCO. Test set results. Bold indicates the best performance with the same supervision and training data. M  $\cap$  G+ denotes the weakly or semi supervised model trained with MCG  $\cap$  Grabcut+.



## D Qualitative results for semantic labelling

Figure 3 presents qualitative results for semantic labelling on Pascal VOC12. The presented semantic labelling examples show that high quality segmentation can be achieved using only detection bounding box annotations. This figure extends Figure 7 of the main paper.

## E Qualitative results for instance segmentations

Figure 4 illustrates additional qualitative results for instance segmentations given by the weakly supervised DeepMask-CRF model. This figure complements Figure 9 from the main paper.

Figure 5 shows examples of instance segmentation given by different methods. This figure complements Figure 8 from the main paper. Our proposed weak-supervision DeepMask-CRF model achieves competitive performance with fully supervised results and provides higher quality output in comparison with box-guided segmentation techniques.

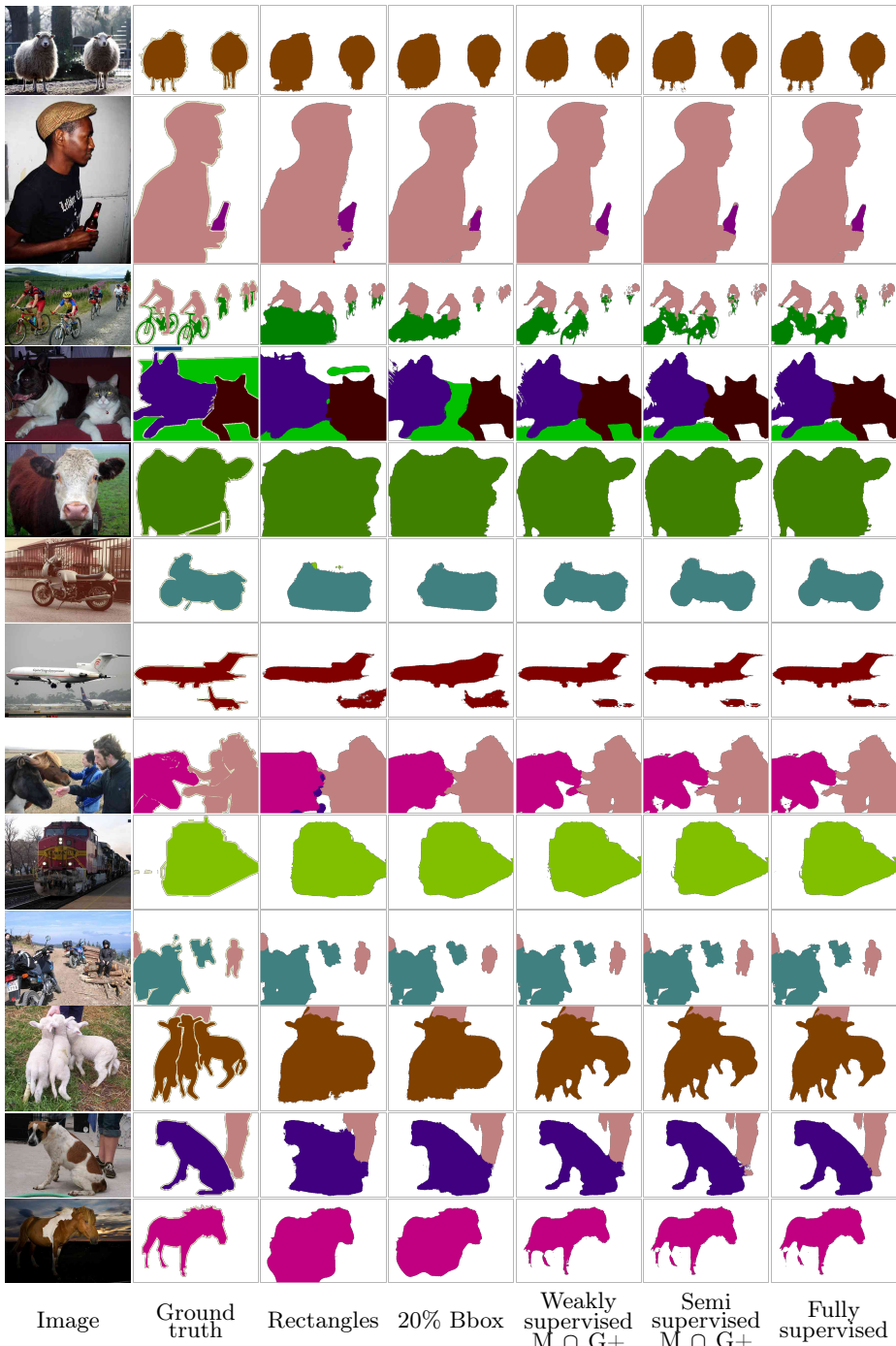


Figure 3: Qualitative results on VOC12.  $M \cap G+$  denotes the weakly or semi supervised model trained with  $M \cap G+$  Grabcut+.



Figure 4: Example result from the weakly supervised DeepMask-CRF model trained with Pascal VOC12 and COCO supervision. White boxes illustrate Fast-RCNN detection proposals used to output the segments which have the best overlap with the ground truth segmentation mask.

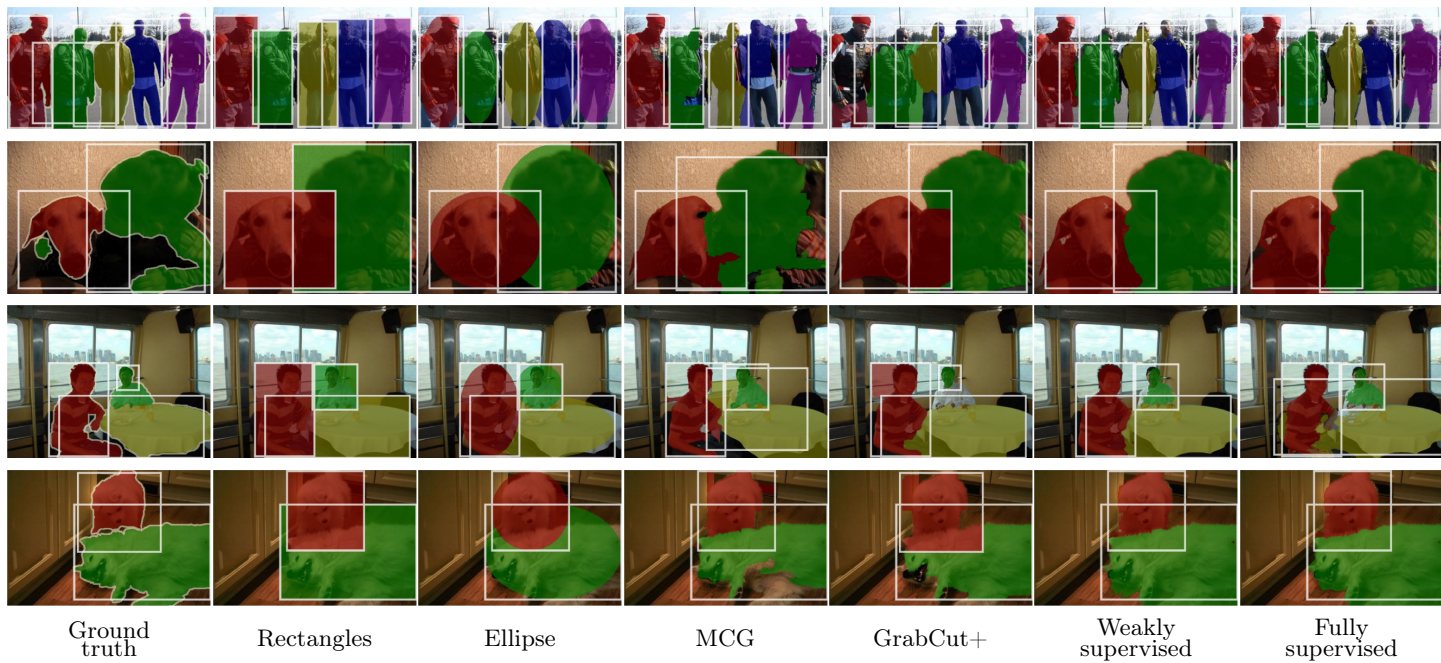


Figure 5: Qualitative results of instance segmentation on VOC12. Weakly and fully supervised refer to results of DeepMask-CRF trained with Pascal VOC12 and COCO supervision. White boxes illustrate Fast-RCNN detection proposals used to output the segments which have the best overlap with the ground truth segmentation mask.