Simple Does It: Weakly Supervised Instance and Semantic Segmentation

Anna Khoreva[1] Rodrigo Benenson[1] Jan Hosang[1] Matthias Hein[2] Bernt Schiele[1]

[1]Max Planck Institute for Informatics, Saarbrücken, Germany
[2]Saarland University, Saarbrücken, Germany

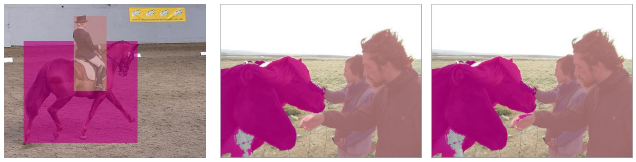
Abstract

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 a new approach that does not require modification of the segmentation training procedure. We show that when carefully designing the input labels from given bounding boxes, even a single round of training is enough to improve over previously reported weakly supervised results. Overall, our weak supervision approach reaches $\sim 95\%$ of the quality of the fully supervised model, both for semantic labelling and instance segmentation.

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 profit from extensive amounts of training data to reach top quality. However, one of their main weaknesses is that they need a large number of training samples for high quality results. This is usually mitigated by using pre-trained models (e.g. with $\sim 10^6$ training samples for ImageNet classification [37]), 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 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 expensive, requiring $\sim 15\times$ more time [25]. Cheaper and easier to define, box annotations are 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.



Training sample, with box annotations | Test image, fully supervised result | Test image, weakly supervised result

Figure 1: We propose a technique to train semantic labelling from bounding boxes, and reach 95% of the quality obtained when training from pixel-wise annotations.

Our experiments focus on the 20 Pascal classes [9] and show that using only bounding box annotations over the same training set we can reach $\sim 95\%$ 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. We also show that, when properly used, "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 recursive training of convnets for weakly supervised semantic labelling, discuss how to reach good quality results, and what are the limitations of the approach (Section 3.1).
- We show that state of the art quality can be reached when properly employing GrabCut-like algorithms to generate training labels from given bounding boxes, instead of modifying the segmentation convnet training procedure (Section 3.2).

– We report the best known results when training using bounding boxes only, both using Pascal VOC12 and VOC12+COCO training data, reaching comparable quality with the fully supervised regime (Section 4.2).

– We are the first to show that similar results can be achieved for the weakly supervised instance segmentation task (Section 6).

2. Related work

Semantic labelling. Semantic labelling may be tackled via decision forests [38] or classifiers over hand-crafted superpixel features [11]. However, convnets have proven particularly effective for semantic labelling. A flurry of variants have been proposed recently [32, 26, 5, 24, 48, 18, 46]. In this work we use DeepLab [5] 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 ($2 \sim 5$ points). The most popular technique is DenseCRF [20], but other variants are also considered [19, 2].

Weakly supervised semantic labelling. In order to keep annotation cost low, recent work has explored different forms of supervision for semantic labelling: image labels [29, 28, 27, 30, 42], points [3], scribbles [44, 23], and bounding boxes [8, 27]. [8, 27, 15] also consider the case where a fraction of images are fully supervised. [44] 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 [8, 27]. BoxSup [8] proposes a recursive training procedure, where the convnet is trained under supervision of segment object proposals and the updated network in turn improves the segments used for training. WSSL [27] proposes an expectation-maximisation algorithm with a bias to enable the network to estimate the foreground regions. We compare with these works in the result sections. Since all implementations use slightly different networks and training procedures, care should be taken during comparison. Both [8] and [27] propose new ways to train convnets under weak supervision. In contrast, in this work we show that one can reach better results without modifying the training procedure (compared to the fully supervised case) by instead carefully generating input labels for training from the bounding box annotations (Section 3).

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 [35, 16] that generate segments (such as [34, 21]) can be used for instance segmentation. Similarly, given a bounding box (e.g. selected by a detector), GrabCut [36] variants can be used to obtain an instance segmentation (e.g. [22, 7, 41, 40, 47]).
To enable end-to-end training of detection and segmentation systems, it has recently been proposed to train convnets for the task of instance segmentation [14, 33]. In this work we explore weakly supervised training of an instance segmentation convnet. We use DeepMask [33] as a reference implementation for this task. In addition we re-purpose DeepLab-v2 network [6], originally designed for semantic segmentation, for the instance segmentation task.

3. From boxes to semantic labels

The goal of this work is to provide high quality semantic labelling starting from object bounding box annotations. We design our approach aiming to exploit the available information at its best. There are two sources of information: the annotated boxes and priors about the objects. We integrate these in the following cues:

C1 Background. Since the bounding boxes are expected to be exhaustive, any pixel not covered by a box is labelled as background.

C2 Object extend. The box annotations bound the extent of each instance. Assuming a prior on the objects shapes (e.g. oval-shaped objects are more likely than thin bar or full rectangular objects), the box also gives information on the expected object area. We employ this size information during training.

C3 Objectness. Other than extent and area, there are additional object priors at hand. Two priors typically used are spatial continuity and having a contrasting boundary with the background. In general we can harness priors about object shape by using segment proposal techniques [35], which are designed to enumerate and rank plausible object shapes in an area of the image.

3.1. Box baselines

We first describe a naive baseline that serves as starting point for our exploration. Given an annotated bounding box and its class label, we label all pixels inside the box with such given class. If two boxes overlap, we assume the smaller one is in front. Any pixel not covered by boxes is labelled as background.

Figure 2 left side and Figure 3c show such example annotations. We use these labels to train a segmentation net-

work with the standard training procedure. We employ the DeepLabv1 approach from [5] (details in Section 4.1).

Recursive training. We observe that when applying the resulting model over the training set, the network outputs capture the object shape significantly better than just boxes (see Figure 2). This inspires us to follow a recursive training procedure, where these new labels are fed in as ground truth for a second training round. We name this recursive training approach `Naive`.

The recursive training is enhanced by de-noising the convnet outputs using extra information from the annotated boxes and object priors. Between each round we improve the labels with three post-processing stages:

1. Any pixel outside the box annotations is reset to background label (cue C1).

2. If the area of a segment is too small compared to its corresponding bounding box (e.g. IoU$<$ 50%), the box area is reset to its initial label (fed in the first round). This enforces a minimal area (cue C2).

3. 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 [20] with the DeepLabv1 parameters [5]). In our weakly supervised scenario, boundary-aware filtering is particularly useful to improve objects delineation (cue C3).

The recursion and these three post-processing stages are crucial to reach good performance. We name this recursive training approach `Box`, and show an example result in Figure 2.

Ignore regions. We also consider a second variant $\texttt{Box}^{\texttt{i}}$ that, instead of using filled rectangles as initial labels, we fill in the 20% inner region, and leave the remaining inner area of the bounding box as ignore regions. See Figure 3d. Following cues C2 and C3 (shape and spatial continuity priors), the 20% inner box region should have higher chances of overlapping with the corresponding object, reducing the noise in the generated input labels. The intuition is that the convnet training might benefit from trading-off lower recall (more ignore pixels) for higher precision (more pixels are correctly labelled). Starting from this initial input, we use the same recursive training procedure as for `Box`.

Despite the simplicity of the approach, as we will see in the experimental section 4, `Box` / $\texttt{Box}^{\texttt{i}}$ is already competitive with the current state of the art.

However, using rectangular shapes as training labels is clearly suboptimal. Therefore, in the next section, we propose an approach that obtains better results while avoiding multiple recursive training rounds.

3.2. Box-driven segments

The box baselines are purposely simple. A next step in complexity consists in utilising the box annotations to generate an initial guess of the object segments. We think of this as "old school meets new school": we use the noisy outputs of classic computer vision methods, box-driven figure-ground segmentation [36] and object proposal [35] techniques, to feed the training of a convnet. Although the output object segments are noisy, they are more precise than simple rectangles, and thus should provide improved results. A single training round will be enough to reach good quality.

3.2.1 GrabCut baselines

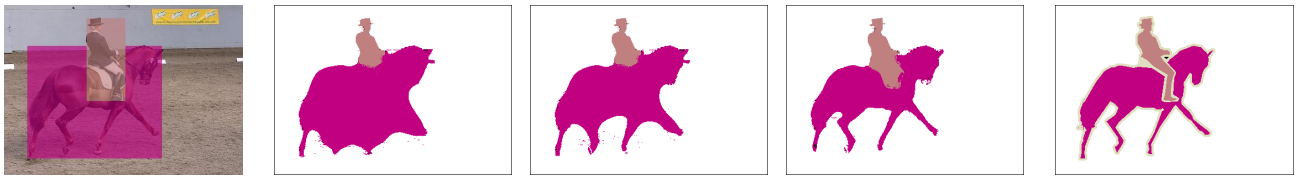
GrabCut [36] is the established technique to estimate an object segment from its bounding box. We propose to use a modified version of GrabCut, which we call `GrabCut+`, where HED boundaries [43] are used as pairwise term instead of the typical RGB colour difference. (The HED boundary detector is trained on the generic boundaries of BSDS500 [1]). We considered other GrabCut variants, such as [7, 40]; however, the proposed `GrabCut+` gives higher quality segments (see supplementary material).
Similar to $\texttt{Box}^{\texttt{i}}$, we also consider a $\texttt{GrabCut+}^{\texttt{i}}$ variant, which trades off recall for higher precision. For each annotated box we generate multiple ($\sim$ 150) perturbed `GrabCut+` outputs. If 70% of the segments mark the pixel as foreground, the pixel is set to the box object class. If less than 20% of the segments mark the pixels as foreground, the pixel is set as background, otherwise it is marked as ignore. The perturbed outputs are generated by jittering the box coordinates ($\pm$5%) as well as the size of the outer background region considered by GrabCut (from 10% to 60%). An example result of $\texttt{GrabCut+}^{\texttt{i}}$ can be seen in Figure 3g.

3.2.2 Adding objectness

With our final approach we attempt to better incorporate the object shape priors by using segment proposals [35]. Segment proposals techniques are designed to generate a soup of likely object segmentations, incorporating as many "objectness" priors as useful (cue C3).

We use the state of the art proposals from MCG [34]. As final stage the MCG algorithm includes a ranking based on a decision forest trained over the Pascal VOC 2012 dataset. We do *not* use this last ranking stage, but instead use all the (unranked) generated segments. Given a box annotation, we pick the highest overlapping proposal as a corresponding segment.

Building upon the insights from the baselines in Section 3.1 and 3.2, we use the MCG segment proposals to supplement `GrabCut+`. Inside the annotated boxes, we mark as



Example input rectangles | Output after 1 training round | After 5 rounds | After 10 rounds | Ground truth

Figure 2: Example results of using only rectangle segments and recursive training (using convnet predictions as supervision for the next round), see Section 3.1.

foreground pixels where both MCG and GrabCut+ agree; the remaining ones are marked as ignore. We denote this approach as MCG ∩ GrabCut+ or M ∩ G+ for short.

Because MCG and GrabCut+provide complementary information, we can think of M ∩ G+ as an improved version of GrabCut+^{i} providing a different trade-off between precision and recall on the generated labels (see Figure 3i).

The BoxSup method [8] also uses MCG object proposals during training; however, there are important differences. They modify the training procedure so as to denoise intermediate outputs by randomly selecting high overlap proposals. In comparison, our approach keeps the training procedure unmodified and simply generates input labels. Our approach also uses ignore regions, while BoxSup does not explore this dimension. Finally, BoxSup uses a longer training than our approach.

Section 4 shows results for the semantic labelling task, compares different methods and different supervision regimes. In Section 5 we show that the proposed approach is also suitable for the instance segmentation task.

4. Semantic labelling results

Our approach is equally suitable (and effective) for weakly supervised instance segmentation as well as for semantic labelling. However, only the latter has directly comparable related work. We thus focus our experimental comparison efforts on the semantic labelling task. Results for instance segmentation are presented in Section 6.

Section 4.1 discusses the experimental setup, evaluation, and implementation details for semantic labelling. Section 4.2 presents our main results, contrasting the methods from Section 3 with the current state of the art. Section 4.3 further expands these results with a more detailed analysis, and presents results when using more supervision (semi-supervised case).

4.1. Experimental setup

Datasets. We evaluate the proposed methods on the Pascal VOC12 segmentation benchmark [9]. 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 previous work [5, 8], we extend the training set with the annotations provided by [12], resulting in an augmented set of 10 582 training images.

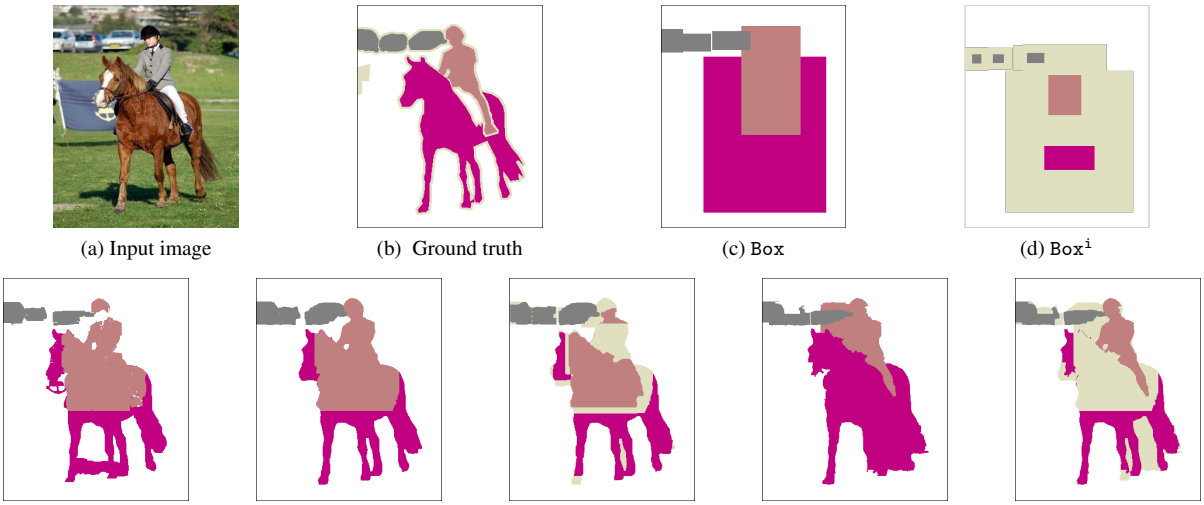
In some of our experiments, we use additional training images from the COCO [25] dataset. We only consider images that contain any of the 20 Pascal classes and (following [48]) only objects with a bounding box area larger than 200 pixels. After this filtering, 99 310 images remain (from training and validation sets), which 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 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 [5]. The model is initialized from a VGG16 network pre-trained on ImageNet [39]. 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 [20]. Our network and post-processing are comparable to the ones used in [8, 27].

Note that multiple strategies have been considered to boost test time results, such as multi-resolution or model ensembles [5, 18]. Here we keep the approach simple and fixed. In all our experiments we use a fixed training and test time procedure. Across experiments we only change the input training data that the networks gets to see.



(a) Input image (b) Ground truth (c) Box (d) Box^i

(e) GrabCut (f) GrabCut+ (g) GrabCut+^{i} (h) MCG (i) M ∩ G+

Figure 3: 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 ∩ G+ denotes MCG ∩ GrabCut+.

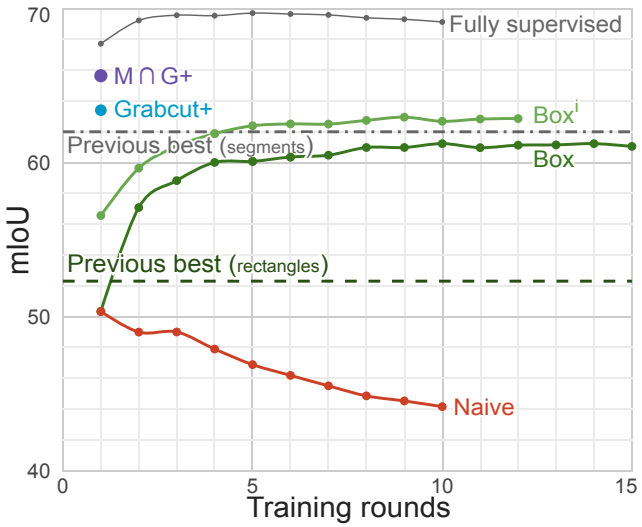


Figure 4: Segmentation quality versus training round for different approaches, see also Tables 1 and 2. Pascal VOC12 validation set results. "Previous best (rectangles/segments)" corresponds to WSSL_R/BoxSup_MCG in Table 2.

4.2. Main results

Box results. Figure 4 presents the results for the recursive training of the box baselines from Section 3.1. We see that the Naive scheme, a recursive training from rectangles disregarding post-processing stages, leads to poor quality. However, by using the suggested three post-processing stages, the Box baseline obtains a significant gain, getting tantalisingly close to the best reported results on the task [8]. Details of the contribution of each post-processing stage are presented in the supplementary material. Adding ignore regions inside the rectangles (Box → Box^i) provides a clear gain and leads by itself to state of the art results.

Method		val. mIoU
-	Fast-RCNN	44.3
	GT Boxes	62.2
Weakly supervised	Box	61.2
	Box^i	62.7
	MCG	62.6
	GrabCut+	63.4
	GrabCut+^{i}	64.3
	M ∩ G+	**65.7**
Fully supervised	DeepLab$_{ours}$ [5]	69.1

Table 1: Weakly supervised semantic labelling results for our baselines. Trained using Pascal VOC12 bounding boxes alone, validation set results. DeepLab$_{ours}$ indicates our fully supervised result.

Figure 4 also shows the result of using longer training for fully supervised case. When using ground truth semantic segmentation annotations, one training round is enough to achieve good performance; longer training brings marginal improvement. As discussed in Section 3.1, reaching good quality for Box/Box^i requires multiple training rounds instead, and performance becomes stable from round 5 onwards. Instead, GrabCut+/M ∩ G+ do not benefit from additional training rounds.

Box-driven segment results. Table 1 evaluates results on the Pascal VOC12 validation set. It indicates the Box/Box^i results after 10 rounds, and MCG/GrabCut+/GrabCut+^{i}/M∩G+ results after one round. "Fast-RCNN" is the result using detections [10] to generate semantic labels (lower-bound), "GT Boxes" considers the

box annotations as labels, and DeepLab$_{ours}$ indicates our fully supervised segmentation network result obtained with a training length equivalent to three training rounds (upper-bound for our results). We see in the results that using ignore regions systematically helps (trading-off recall for precision), and that M$\cap$G+ provides better results than MCG and GrabCut+ alone.

Table 2 indicates the box-driven segment results after 1 training round and shows comparison with other state of the art methods, trained from boxes only using either Pascal VOC12, or VOC12+COCO data. BoxSup$_{\mathrm{R}}$ and WSSL$_{\mathrm{R}}$ both feed the network with rectangle segments (comparable to Box$^{\mathrm{i}}$), while WSSL$_{\mathrm{S}}$ and BoxSup$_{\mathrm{MCG}}$ exploit arbitrary shaped segments (comparable to M $\cap$ G+). Although our network and post-processing is comparable to the ones in [8, 27], there are differences in the exact training procedure and parameters (details in supplementary material).

Overall, our results indicate that - without modifying the training procedure - M $\cap$ G+ is able to improve over previously reported results and reach 95% of the fully-supervised training quality. By training with COCO data [25] before fine-tuning for Pascal VOC12, we see that with enough additional bounding boxes we can match the full supervision from Pascal VOC 12 (68.9 versus 69.1). This shows that the labelling effort could be significantly reduced by replacing segmentation masks with bounding box annotations.

4.3. Additional results

Semi-supervised case. Table 2 compares results in the semi-supervised modes considered by [8, 27], where some of the images have full supervision, and some have only bounding box supervision. Training with 10% of Pascal VOC12 semantic labelling annotations does not bring much gain to the performance (65.7 versus 65.8), this hints at the high quality of the generated M $\cap$ G+ input data.

By using ground-truth annotations on Pascal plus bounding box annotations on COCO, we observe 2.5 points gain (69.1 $\rightarrow$ 71.6 , see Table 2). This suggests that the overall performance could be further improved by using extra training data with bounding box annotations.

Boundaries supervision. Our results from MCG, GrabCut+, and M $\cap$ G+ all indirectly include information from the BSDS500 dataset [1] via the HED boundary detector [43]. These results are fully comparable to BoxSup-MCG [8], to which we see a clear improvement. Nonetheless one would like to know how much using dense boundary annotations from BSDS500 contributes to the results. We use the weakly supervised boundary detection technique from [17] to learn boundaries directly from the Pascal VOC12 box annotations. Training M $\cap$ G+ using weakly supervised HED boundaries results in 1 point loss compared to using the BSDS500 (64.8 versus 65.7 mIoU on Pascal VOC12 validation set). We see then that although the additional supervision does bring some help, it has a minor effect and our results are still rank at the top even when we use only Pascal VOC12 + ImageNet pre-training.

Super-vision	#GT images	#Weak images	Method	val. set mIoU	test set mIoU	test set FS%
VOC12 (V)						
Weak	-	V 10k	Bearman et al. [3]	45.1	-	-
			BoxSup$_{\mathrm{R}}$ [8]	52.3	-	-
			WSSL$_{\mathrm{R}}$[27]	52.5	54.2	76.9
			WSSL$_{\mathrm{S}}$[27]	60.6	62.2	88.2
			BoxSup$_{\mathrm{MCG}}$[8]	62.0	64.6	91.6
			Box$^{\mathrm{i}}$	62.7	63.5	90.0
			M $\cap$ G+	**65.7**	**67.5**	**95.7**
Semi	V 1.4k	V 9k	WSSL$_{\mathrm{R}}$[27]	62.1	-	-
			BoxSup$_{\mathrm{MCG}}$[8]	63.5	66.2	93.9
			WSSL$_{\mathrm{S}}$[27]	65.1	66.6	94.5
			M $\cap$ G+	**65.8**	**66.9**	**94.9**
Full	V 10k	-	BoxSup [8]	63.8	-	-
			WSSL [27]	67.6	70.3	99.7
			DeepLab$_{ours}$ [5]	69.1	70.5	100
VOC12 + COCO (V+C)						
Weak	-	V+C 110k	Box$^{\mathrm{i}}$	65.3	66.7	91.1
			M $\cap$ G+	**68.9**	**69.9**	**95.5**
Semi	V 10k	C 123k C 100k	BoxSup$_{\mathrm{MCG}}$[8]	68.2	71.0	97.0
			M $\cap$ G+	**71.6**	**72.8**	**99.5**
Full	V+C 133k V+C 110k	-	BoxSup [8]	68.1	-	-
			WSSL [27]	71.7	73	99.7
			DeepLab$_{ours}$ [5]	72.3	73.2	100

Table 2: Semantic labelling results for validation and test set; under different training regimes with VOC12 (V) and COCO data (C). Underline indicates full supervision baselines, and bold are our best weakly- and semi-supervised results. FS%: performance relative to the best fully supervised model (DeepLab$_{ours}$). Discussion in Sections 4.2 and 4.3.

Different convnet results. For comparison purposes with [8, 27] we used DeepLabv1 with a VGG-16 network in our experiments. To show that our approach also generalizes across different convnets, we also trained DeepLabv2 with a ResNet101 network [6]. Table 3 presents the results.

Similar to the case with VGG-16, our weakly supervised approach M $\cap$ G+ reaches 93%/95% of the fully supervised case when training with VOC12/VOC12+COCO, and the weakly supervised results with COCO data reach similar quality to full supervision with VOC12 only.

5. From boxes to instance segmentation

Complementing the experiments of the previous sections, we also explore a second task: weakly supervised in-

Figure 5: Qualitative results on VOC12. Visually, the results from our weakly supervised method M ∩ G+ are hardly distinguishable from the fully supervised ones.

Supervision	Method	mIoU	FS%
VOC12			
Weak	M ∩ G+	69.4	93.2
Full	DeepLabv2-ResNet101 [6]	74.5	100
VOC12 + COCO			
Weak	M ∩ G+	74.2	95.5
Full	DeepLabv2-ResNet101 [6]	77.7	100

Table 3: DeepLabv2-ResNet101 network semantic labelling results on VOC12 validation set, using VOC12 or VOC12+COCO training data. FS%: performance relative to the full supervision. Discussion in Section 4.3.

stance 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 [14, 33, 31] 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 and expensive than semantic labelling, and thus there is interest in weakly supervised training.

The segments used for training, as discussed in Section 3.2, 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. For each annotated bounding box, we generate a foreground versus background segmentation using the GrabCut+ method (Section 3.2), and train a convnet to regress from the image and bounding box information to the instance segment.

6. Instance segmentation results

Experimental setup. We choose a purposely simple instance segmentation pipeline, based on the "hyper-columns system 2" architecture [14]. We use Fast-RCNN [10] detections (post-NMS) with their class score, and for each detection estimate an associated foreground segment. We estimate the foreground using either some baseline method (e.g. GrabCut) or using convnets trained for the task [33, 6].

For our experiments we use a re-implementation of the DeepMask [33] architecture, and additionally we repurpose a DeepLabv2 VGG-16 network [6] for the instance segmentation task, which we name DeepLab$_{\text{BOX}}$.

Inspired by [45, 4], we modify DeepLab to accept four input channels: the input image RGB channels, plus a binary map with a bounding box of the object instance to segment. We train the network DeepLab$_{\text{BOX}}$ to output the segmentation mask of the object corresponding to the input bounding box. The additional input channel guides the network so as to segment only the instance of interest instead of all objects in the scene. The input box rectangle can also be seen as an initial guess of the desired output. We train using ground truth bounding boxes, and at test time Fast-RCNN

Supervision	Method	$mAP^r_{0.5}$	$mAP^r_{0.75}$	ABO
-	Rectangle	21.6	1.8	38.5
	Ellipse	29.5	3.9	41.7
	MCG	28.3	5.9	44.7
	GrabCut	38.5	13.9	45.8
	GrabCut+	41.1	17.8	46.4
VOC12				
Weak	DeepMask	39.4	8.1	45.8
	$\text{DeepLab}_{\text{BOX}}$	44.8	16.3	**49.1**
Full	DeepMask	41.7	9.7	47.1
	$\text{DeepLab}_{\text{BOX}}$	47.5	20.2	<u>51.1</u>
VOC12 + COCO				
Weak	DeepMask	42.9	11.5	48.8
	$\text{DeepLab}_{\text{BOX}}$	46.4	18.5	**51.4**
Full	DeepMask	44.7	13.1	49.7
	$\text{DeepLab}_{\text{BOX}}$	49.4	23.7	<u>53.1</u>

Table 4: Instance segmentation results on VOC12 validation set. Underline indicates the full supervision baseline, and bold are our best weak supervision results. Weakly supervised DeepMask and $\text{DeepLab}_{\text{BOX}}$ reach comparable results to full supervision. See Section 6 for details.

detection boxes are used.

We train DeepMask and $\text{DeepLab}_{\text{BOX}}$ using GrabCut+ results either over Pascal VOC12 or VOC12+COCO data (1 training round, no recursion like in Section 3.1), and test on the VOC12 validation set, the same set of images used in Section 4. The augmented annotation from [12] provides per-instance segments for VOC12. We do not use CRF post-processing for neither of the networks.

Following instance segmentation literature [13, 14] we report in Table 4 mAP^r at IoU threshold 0.5 and 0.75. mAP^r is similar to the traditional VOC12 evaluation, but using IoU between segments instead of between boxes. Since we have a fixed set of windows, we can also report the average best overlap (ABO) [35] metric to give a different perspective on the results.

Baselines. We consider five training-free baselines: simply filling in the detection rectangles (boxes) with foreground labels, fitting an ellipse inside the box, using the MCG proposal with best bounding box IoU, and using GrabCut and GrabCut+ (see Section 3.2), initialized from the detection box.

Analysis. The results table 4 follows the same trend as the semantic labelling results in Section 4. GrabCut+ provides the best results among the baselines considered and shows comparable performance to DeepMask, while our proposed $\text{DeepLab}_{\text{BOX}}$ outperforms both techniques. We see that our weakly supervised approach reaches $\sim 95\%$ of the quality of fully-supervised case (both on $mAP^r_{0.5}$ and ABO metrics) using two different convnets, DeepMask and $\text{DeepLab}_{\text{BOX}}$, both when training with VOC12 or VOC12+COCO.

Figure 6: Example result from our weakly supervised DeepMask (VOC12+COCO) model.

Examples of the instance segmentation results from weakly supervised DeepMask (VOC12+COCO) are shown in Figure 6. Additional example results are presented in the supplementary material.

7. Conclusion

The series of experiments presented in this paper provides new insights on how to train pixel-labelling convnets from bounding box annotations only. We showed that when carefully employing the available cues, recursive training using only rectangles as input can be surprisingly effective (Box^i). Even more, when using box-driven segmentation techniques and doing a good balance between accuracy and recall in the noisy training segments, we can reach state of the art performance without modifying the segmentation network training procedure ($\text{M}\cap\text{G}+$). 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). By employing extra training data with bounding box annotations from COCO we are able to match the full supervision results. We also report the first results for weakly supervised instance segmentation, where we also reach $\sim 95\%$ of the quality of the fully-supervised training.

Our current approach exploits existing box-driven segmentation techniques, treating each annotated box individually. In future work we would like to explore co-segmentation ideas (treating the set of annotations as a whole), and consider even weaker forms of supervision.

References

[1] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *PAMI*, 2011. 3, 6, 12

[2] J. Barron and B. Poole. The fast bilateral solver. *arXiv preprint arXiv:1511.03296*, 2015. 2

[3] A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei. What's the point: Semantic segmentation with point supervision. *arXiv preprint arXiv:1506.02106*, 2015. 2, 6

[4] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik. Human pose estimation with iterative error feedback. In *CVPR*, 2016. 7

[5] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*, 2015. 2, 3, 4, 5, 6, 11, 12, 14

[6] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv:1606.00915*, 2016. 2, 6, 7

[7] M. Cheng, V. Prisacariu, S. Zheng, P. Torr, and C. Rother. Densecut: Densely connected crfs for realtime grabcut. *Computer Graphics Forum*, 2015. 2, 3, 12

[8] J. Dai, K. He, and J. Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *ICCV*, 2015. 2, 4, 5, 6, 11, 12, 14

[9] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 2015. 1, 4

[10] R. Girshick. Fast R-CNN. In *ICCV*, 2015. 5, 7

[11] S. Gould, R. Fulton, and D. Koller. Decomposing a scene into geometric and semantically consistent regions. In *ICCV*, 2009. 2

[12] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. In *ICCV*, 2011. 4, 8

[13] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *ECCV*, 2014. 8

[14] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *CVPR*, 2015. 2, 6, 7, 8

[15] S. Hong, H. Noh, and B. Han. Decoupled deep neural network for semi-supervised semantic segmentation. In *NIPS*, 2015. 2

[16] J. Hosang, R. Benenson, P. Dollár, and B. Schiele. What makes for effective detection proposals? *PAMI*, 2015. 2

[17] A. Khoreva, R. Benenson, M. Omran, M. Hein, and B. Schiele. Weakly supervised object boundaries. In *CVPR*, 2016. 6

[18] I. Kokkinos. Pushing the boundaries of boundary detection using deep learning. In *ICLR*, 2016. 2, 4

[19] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts?. *PAMI*, 2004. 2

[20] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NIPS*. 2011. 2, 3, 4, 11

[21] P. Krähenbühl and V. Koltun. Learning to propose objects. In *CVPR*, 2015. 2

[22] V. Lempitsky, P. Kohli, C. Rother, and T. Sharp. Image segmentation with a bounding box prior. In *ICCV*, 2009. 2

[23] D. Lin, J. Dai, J. Jia, K. He, and J. Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *CVPR*, 2016. 2

[24] G. Lin, C. Shen, A. van dan Hengel, and I. Reid. Efficient piecewise training of deep structured models for semantic segmentation. In *CVPR*, 2016. 2

[25] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 1, 4, 6

[26] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 2

[27] G. Papandreou, L. Chen, K. Murphy, , and A. L. Yuille. Weakly- and semi-supervised learning of a dcnn for semantic image segmentation. In *ICCV*, 2015. 2, 4, 6, 11, 12, 14

[28] D. Pathak, P. Kraehenbuehl, and T. Darrell. Constrained convolutional neural networks for weakly supervised segmentation. In *ICCV*, 2015. 2

[29] D. Pathak, E. Shelhamer, J. Long, and T. Darrell. Fully convolutional multi-class multiple instance learning. In *ICLR workshop*, 2015. 2

[30] P. Pinheiro and R. Collobert. From image-level to pixel-level labeling with convolutional network. In *CVPR*, 2015. 2

[31] P. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár. Learning to refine object segments. In *ECCV*, 2016. 6

[32] P. O. Pinheiro and R. Collobert. Recurrent convolutional neural networks for scene labeling. In *ICML*, 2014. 2

[33] P. O. Pinheiro, R. Collobert, and P. Dollár. Learning to segment object candidates. In *NIPS*, 2015. 2, 6, 7

[34] J. Pont-Tuset, P. Arbeláez, J. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping for image segmentation and object proposal generation. *arXiv preprint arXiv:1503.00848*, 2015. 2, 3

[35] J. Pont-Tuset and L. V. Gool. Boosting object proposals: From pascal to coco. In *ICCV*, 2015. 2, 3, 8

[36] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM Trans. Graphics*, 2004. 2, 3, 12

[37] 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. *IJCV*, 2015. 1

[38] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *IJCV*, 2009. 2

[39] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 4

[40] M. Tang, I. Ben Ayed, D. Marin, and Y. Boykov. Secrets of grabcut and kernel k-means. In *ICCV*, 2015. 2, 3, 12

[41] T. Taniai, Y. Matsushita, and T. Naemura. Superdifferential cuts for binary energies. In *CVPR*, 2015. 2

[42] Y. Wei, X. Liang, Y. Chen, X. Shen, M.-M. Cheng, Y. Zhao, and S. Yan. Stc: A simple to complex framework for weakly-supervised semantic segmentation. *arXiv preprint arXiv:1509.03150*, 2015. 2

[43] S. Xie and Z. Tu. Holistically-nested edge detection. In *ICCV*, 2015. 3, 6, 12

[44] J. Xu, A. Schwing, and R. Urtasun. Learning to segment under various forms of weak supervision. In *CVPR*, 2015. 2

[45] N. Xu, B. Price, S. Cohen, J. Yang, and T. S. Huang. Deep interactive object selection. In *CVPR*, 2016. 7

[46] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016. 2

[47] H. Yu, Y. Zhou, H. Qian, M. Xian, Y. Lin, D. Guo, K. Zheng, K. Abdelfatah, and S. Wang. Loosecut: Interactive image segmentation with loosely bounded boxes. *arXiv preprint arXiv:1507.03060*, 2015. 2

[48] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. Torr. Conditional random fields as recurrent neural networks. In *ICCV*, 2015. 2, 4

Supplementary material

A. Content

This supplementary material provides additional quantitative and qualitative results:

- Section B analyses the contribution of the post-processing stages during recursive training (Figure S1).
- Section C discusses training differences of our approach in contrast to the related work.
- We report a comparison of different GrabCut-like methods on Pascal VOC12 boxes in Section D.
- Section E (Figure S2) shows visualization of the different variants of the proposed segmentation inputs obtained from bounding box annotations for weakly supervised semantic segmentation.
- Detailed performance of each class for semantic labelling is reported in Section F (Table S2).
- Section G provides additional qualitative results for weakly supervised semantic segmentation on Pascal VOC12 (Figure S3).
- Qualitative results for instance segmentation are shown in Section H (Figure S4 and Figure S5).

B. Recursive training with boxes

In Section 3 of the main paper we recursively train a convnet directly on the full extend of bounding box annotations as foreground labels, disregarding post-processing stages. We name this recursive training approach `Naive`. Using this supervision and directly applying recursive training leads to significant degradation of the segmentation output quality, see Figure S1.

To improve the labels between the training rounds three post-processing stages are proposed. Here we discuss them in more detail:

1. **Box enforcing**: Any pixel outside the box annotations is reset to background label (cue C1, see Section 3 in the main paper).
2. **Outliers reset**: If the area of a segment is too small compared to its corresponding bounding box (e.g. IoU$<$ 50%), the box area is reset to its initial label (fed in the first round). This enforces a minimal area (cue C2).
3. **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 [20] with the DeepLabv1 parameters [5]). In our weakly supervised scenario, boundary-aware filtering is particularly useful to improve objects delineation (cue C3).

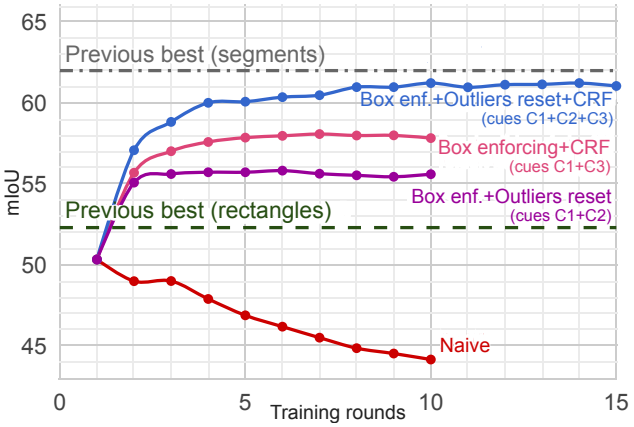


Figure S1: Recursive training from rectangles only as input. Validation set results. All methods use only rectangles as initial input, except "previous best (segments)".

Results. Figure S1 presents results of the recursive training using boxes as input and shows the contribution of the post-processing stages. We see that the naive recursive training is ineffectual. However as soon as some constraints (box enforcing and outliers reset, cues C1+C2) are enforced, the quality improves dramatically after the first round of recursive training. These results already improve over previous work considering rectangles only input [8, 27] (both using a similar convnet to ours) and achieve 3 points improvement over [27] (from 52.5 to 55.6 mIoU, see Figure S1 "Box enf.+Outliers reset").

Even more, when also adding CRF filtering (+ cue C3) 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 [8], which achieve around 62% mIoU (see Figure S1 and Table S2).

Our results indicate that recursive training of a convnet is robust to input noise as soon as appropriate care is taken to de-noise the output between rounds, enabled by given bounding boxes and object priors.

C. Training details in comparison with BoxSup and WSSL

In this work we focus on box level annotations for semantic labelling of objects. The closest related work are thus [8, 27]. Since all implementations use slightly different networks and training procedures, care should be taken

during comparison. Both [8] and [27] propose new ways to train convnets under weak supervision. Both of the approaches build upon the DeepLab network [5], however, there are a few differences in the network architecture.

WSSL [27] employs 2 different variants of the DeepLab architecture with small and large receptive field of view (FOV) size. For each experiment WSSL evaluates with both architectures and reports the best result obtained (using boxes or segments as input). BoxSup [8] uses their own implementation of the DeepLab with the small FOV. In our approach all the experiments employ the DeepLab architecture with the large FOV.

There are also differences in the training procedure. For SGD WSSL uses a mini-batch of 20-30 images and finetunes the network for about 12 hours (number of epochs is not specified) with the standard learning parameters (following [5]). In the SGD training BoxSup uses a mini-batch size of 20 and the learning rate is divided by 10 after every 15 epochs. The training is terminated after 45 epochs. We use a mini-batch of 30 images for SGD and the learning rate is divided by 10 after every 2k iterations, ~6 epochs. Our network is trained for 6k iterations, ~18 epochs.

Similarly to our approach, the BoxSup method [8] uses MCG object proposals during training. However, there are important differences. They modify the training procedure so as to denoise intermediate outputs by randomly selecting high overlap proposals. In comparison, our approach keeps the training procedure unmodified and simply generates input labels. Our approach also uses ignore regions, while BoxSup does not explore this dimension.

WSSL [27] proposes an expectation-maximisation algorithm with a bias to enable the network to estimate the foreground regions. In contrast, in our work we show that one can reach better results without modifying the training procedure (compared to the fully supervised case) by instead carefully generating input labels for training from the bounding box annotations (Section 3.2 in the main paper).

D. GrabCut variants

As discussed in Section 3.2 in the main paper 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 3.1 in the main paper 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 noisier object segments we would like to reach better performance with just one training round.

For this purpose we explore different GrabCut-like [36] techniques, the corresponding quantitative results are in Table S1. For evaluation we use the mean IoU measure. Previous work evaluated using the 50 images from the GrabCut dataset [36], or 1k images with one salient object [7]. The evaluation of Table S1 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.

	Method	mIoU
GrabCut variants	DenseCut [7]	52.5
	Bbox-Seg+CRF [27]	71.1
	GrabCut [36]	72.9
	KGrabCut [40]	73.5
	GrabCut+	75.2

Table S1: GrabCut variants, evaluated on Pascal VOC12 validation set. See Section D for details.

`GrabCut` [36] is the established technique to estimate an object segment from its bounding box. To further improve its quality we propose to use better pairwise terms. We name this variant `GrabCut+`. Instead of the typical RGB colour difference the pairwise terms in `GrabCut+` are replaced by probability of boundary as generated by HED [43]. The HED boundary detector is trained on the generic boundaries of BSDS500 [1]. Moving from `GrabCut` to `GrabCut+` brings a $\sim$ 2 points improvement, see Table S1.

We also experimented with other variants such as `DenseCut` [7] and `KGrabCut` [40] but did not obtain significant gains.

[27] 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 `Bbox-Seg+CRF` in Table S1 and underperforms compared to `GrabCut` and `GrabCut+`.

E. Examples of input segmentations

Figure S2 presents examples of the considered weak annotations. This figure extends Figure 3 of the main paper.

F. Detailed test set results for semantic labelling

In Table S2, we present per class results on the Pascal VOC12 test set for the methods reported in the main paper in Table 2.

On average with our weakly supervised results we achieve $\sim$ 95% quality of full supervision across all classes when training with VOC12 only or VOC12+COCO.

G. Qualitative results for semantic labelling

Figure S3 presents qualitative results for semantic labelling on Pascal VOC12. The presented semantic la-

belling examples show that high quality segmentation can be achieved using only detection bounding box annotations. This figure extends Figure 5 of the main paper.

H. Qualitative results for instance segmentations

Figure S4 illustrates additional qualitative results for instance segmentations given by the weakly supervised DeepMask and $\text{DeepLab}_{\text{BOX}}$ models. This figure complements Figure 6 from the main paper.

Figure S5 shows examples of instance segmentation given by different methods. Our proposed weakly supervised DeepMask model achieves competitive performance with fully supervised results and provides higher quality output in comparison with box-guided segmentation techniques. The $\text{DeepLab}_{\text{BOX}}$ model also provides similar results, see Table 4 in the main paper.

Training data	Super-vision	Method	mean	plane	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	motor bike	per son	plant	sheep	sofa	train	tv
VOC12	weak	Box	62.2	62.6	24.5	63.7	56.7	68.1	84.3	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
		Box^{i}	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+	**67.5**	78.1	31.1	72.4	61.0	67.2	84.2	78.2	81.7	27.6	68.5	62.1	76.9	70.8	78.0	76.3	51.7	78.3	48.3	74.2	58.6
	semi	M $\cap$ G+	66.9	75.8	32.3	75.9	60.1	65.7	82.9	75.0	79.5	29.5	68.5	60.6	76.2	68.6	76.9	75.2	53.2	76.6	49.5	73.8	58.6
	full	WSSL [27]	70.3	83.5	36.6	82.5	62.3	66.5	85.4	78.5	83.7	30.4	72.9	60.4	78.5	75.5	82.1	79.7	58.2	82.0	48.8	73.7	63.3
		DeepLab$_{ours}$ [5]	70.5	85.3	38.3	79.4	61.4	68.9	86.4	82.1	83.6	30.3	74.5	53.8	78.0	77.0	83.7	81.8	55.6	79.8	45.9	79.3	63.4
VOC12 + COCO	weak	Box^{i}	66.7	69.0	27.5	77.1	61.9	65.3	84.2	75.5	83.2	25.7	73.6	63.6	78.2	69.3	75.3	75.2	51.0	73.5	46.2	74.4	60.4
		M $\cap$ G+	**69.9**	82.5	33.4	82.5	59.5	65.8	85.3	75.6	86.4	29.3	77.1	60.8	80.7	79.0	80.5	77.6	55.9	78.4	48.6	75.2	61.5
	semi	BoxSup [8]	71.0	86.4	35.5	79.7	65.2	65.2	84.3	78.5	83.7	30.5	76.2	62.6	79.3	76.1	82.1	81.3	57.0	78.2	55.0	72.5	68.1
		M $\cap$ G+	**72.8**	87.6	37.7	86.7	65.5	67.3	86.8	81.1	88.3	30.7	77.3	61.6	82.7	79.4	84.1	82.0	60.3	84.0	49.4	77.8	64.7
	full	WSSL [27]	72.7	89.1	38.3	88.1	63.3	69.7	87.1	83.1	85.0	29.3	76.5	56.5	79.8	77.9	85.8	82.4	57.4	84.3	54.9	80.5	64.1
		DeepLab$_{ours}$ [5]	73.2	88.8	37.3	83.8	66.5	70.1	89.0	81.4	87.3	30.2	78.8	61.6	82.4	82.3	84.4	82.2	59.1	85.0	50.8	79.7	63.8

Table S2: 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+.

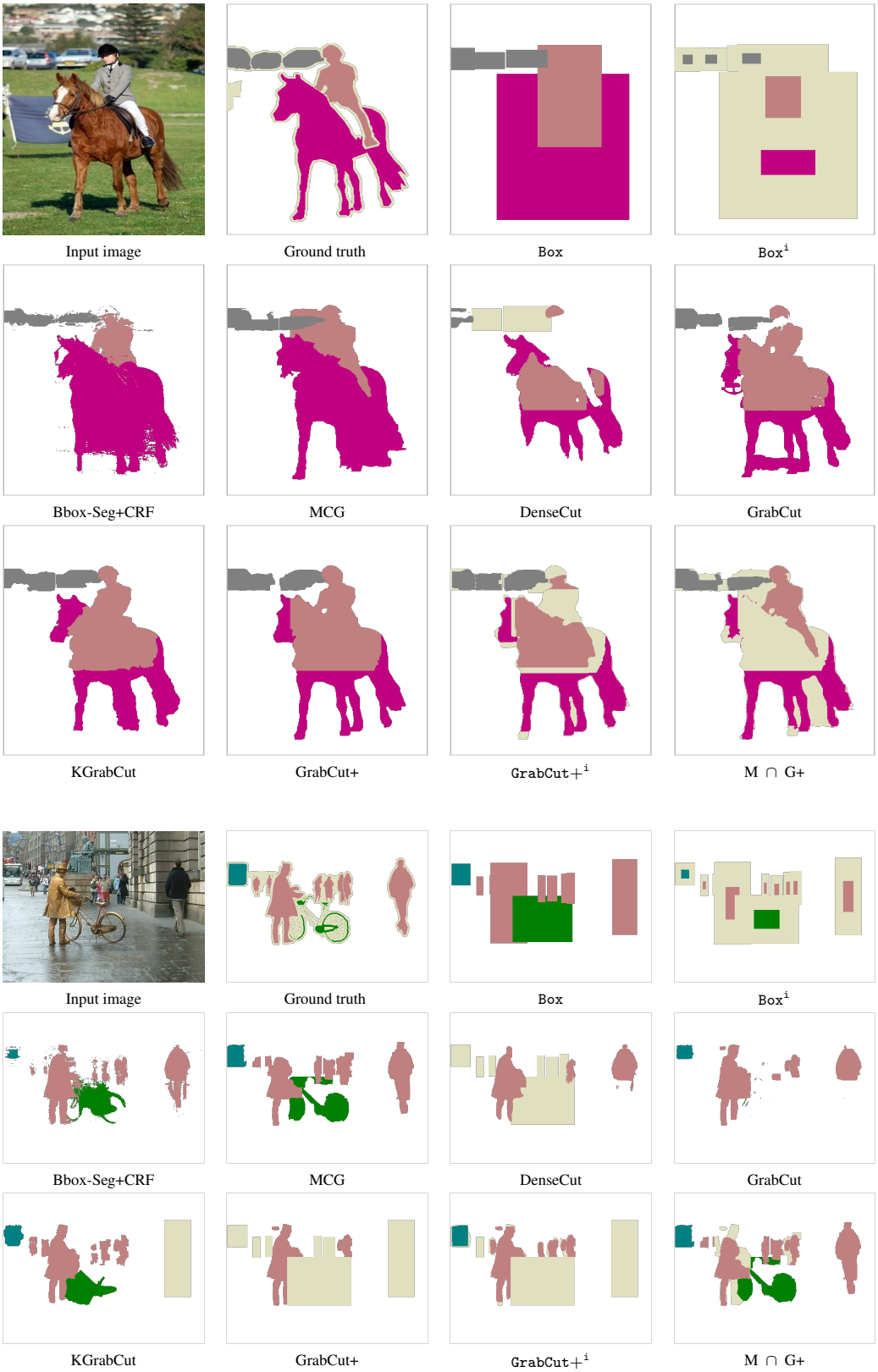


Figure S2: Different segmentations obtained starting from a bounding box. White is background and ignore regions are beige. M ∩ G+ denotes MCG ∩ Grabcut+.

Figure S2: Different segmentations obtained starting from a bounding box. White is background and ignore regions are beige. M ∩ G+ denotes MCG ∩ Grabcut+.

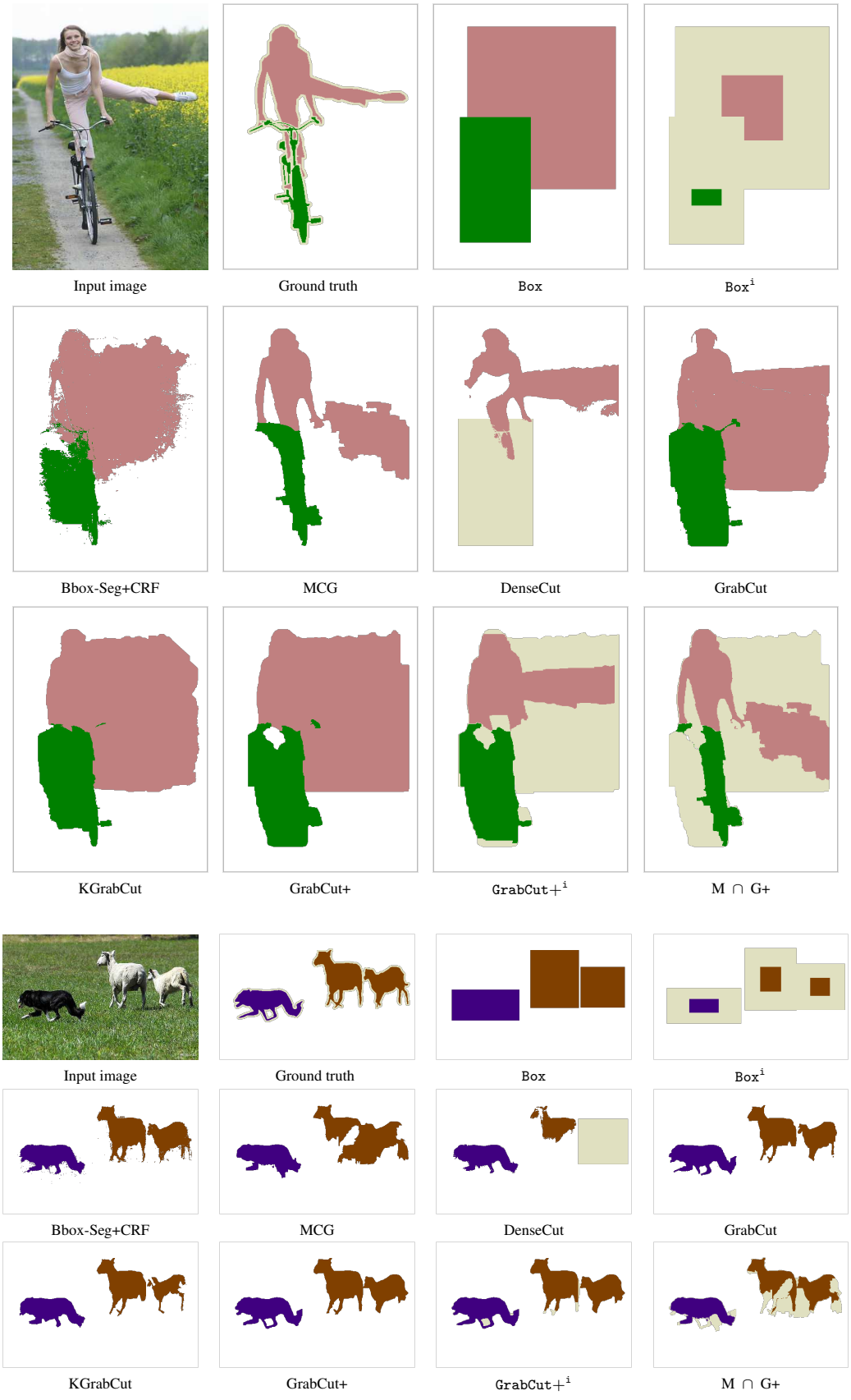


Figure S2: Different segmentations obtained starting from a bounding box. White is background and ignore regions are beige. M ∩ G+ denotes MCG ∩ Grabcut+.

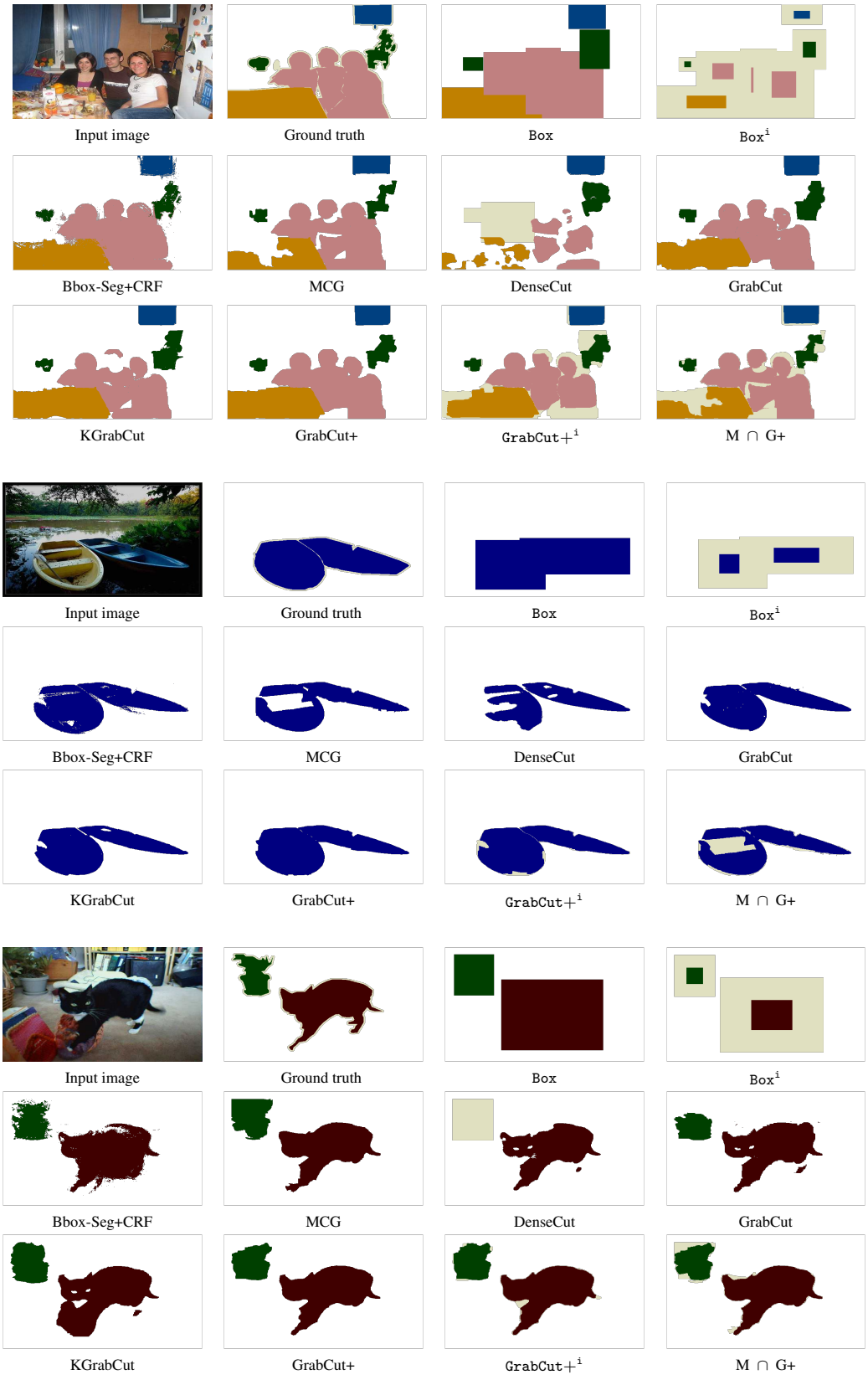


Figure S2: Different segmentations obtained starting from a bounding box. White is background and ignore regions are beige. M ∩ G+ denotes MCG ∩ Grabcut+.

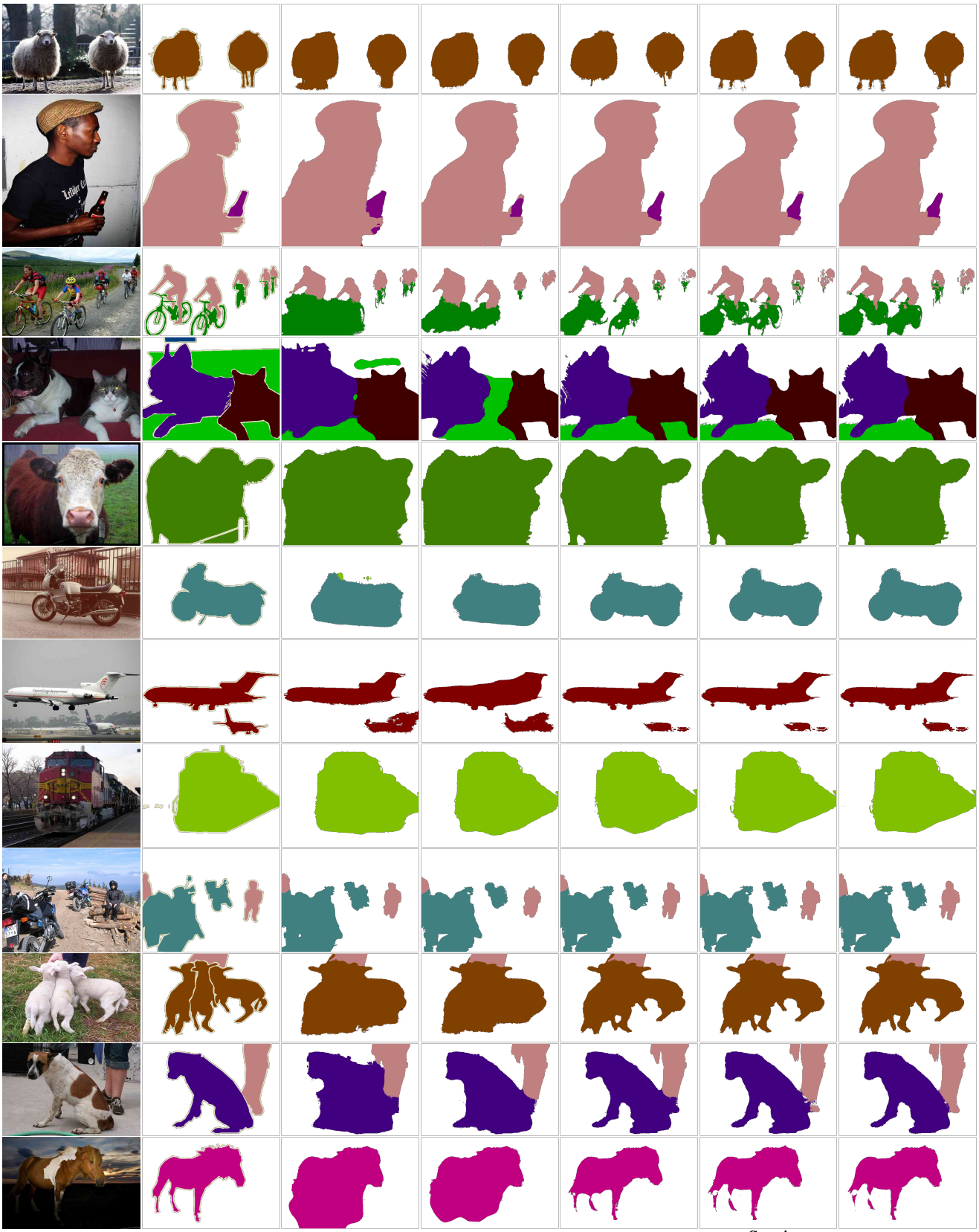


Figure S3: Qualitative results on VOC12. M ∩ G+ denotes the weakly supervised model trained on MCG ∩ Grabcut+.

DeepMask

$\text{DeepLab}_{\text{BOX}}$

Figure S4: Example results from the DeepMask and $\text{DeepLab}_{\text{BOX}}$ models trained with Pascal VOC12 and COCO using box supervision. White boxes illustrate Fast-RCNN detection proposals used to output the segments which have the best overlap with the ground truth segmentation mask.

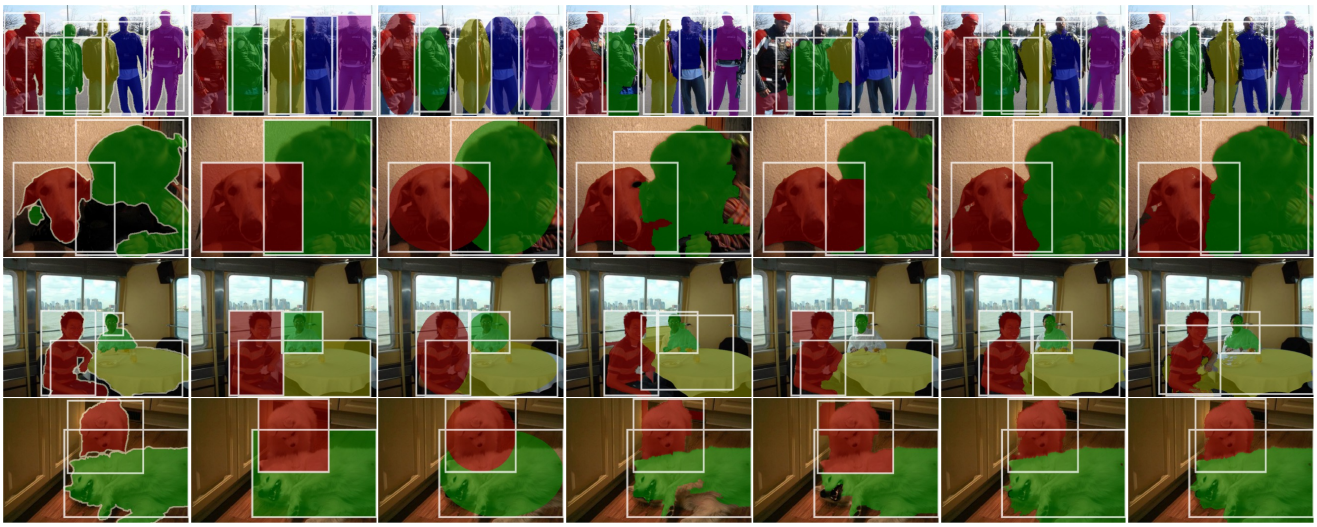


Figure S5: Qualitative results of instance segmentation on VOC12. Example result from the DeepMask model are trained with Pascal VOC12 and COCO supervision. White boxes illustrate Fast-RCNN detection proposals used to output the segments which have the best overlap with the ground truth segmentation mask.