Weakly- and Semi-Supervised Panoptic Segmentation

Qizhu Li*, Anurag Arnab*, and Philip H.S. Torr

University of Oxford
{liqizhu, aarnab, phst}@robots.ox.ac.uk

Abstract. We present a weakly supervised model that jointly performs both semantic- and instance-segmentation – a particularly relevant problem given the substantial cost of obtaining pixel-perfect annotation for these tasks. In contrast to many popular instance segmentation approaches based on object detectors, our method does not predict any overlapping instances. Moreover, we are able to segment both "thing" and "stuff" classes, and thus explain all the pixels in the image. "Thing" classes are weakly-supervised with bounding boxes, and "stuff" with image-level tags. We obtain state-of-the-art results on Pascal VOC, for both full and weak supervision (which achieves about 95% of fully-supervised performance). Furthermore, we present the first weakly-supervised results on Cityscapes for both semantic- and instance-segmentation. Finally, we use our weakly supervised framework to analyse the relationship between annotation quality and predictive performance, which is of interest to dataset creators.

Keywords: weak supervision, instance segmentation, semantic segmentation, scene understanding

1 Introduction

Convolutional Neural Networks (CNNs) excel at a wide array of image recognition tasks [1–3]. However, their ability to learn effective representations of images requires large amounts of labelled training data [4, 5]. Annotating training data is a particular bottleneck in the case of segmentation, where labelling each pixel in the image by hand is particularly time-consuming. This is illustrated by the Cityscapes dataset where finely annotating a single image took "more than 1.5h on average" [6]. In this paper, we address the problems of semantic- and instance-segmentation using only weak annotations in the form of bounding boxes and image-level tags. Bounding boxes take only 7 seconds to draw using the labelling method of [7], and image-level tags an average of 1 second per class [8]. Using only these weak annotations would correspond to a reduction factor of 30 in labelling a Cityscapes image which emphasises the importance of cost-effective, weak annotation strategies.

Our work differs from prior art on weakly-supervised segmentation [9–13] in two primary ways: Firstly, our model jointly produces semantic- and instance-segmentations of the image, whereas the aforementioned works only output instance-agnostic semantic segmentations. Secondly, we consider the segmentation of both "thing" and "stuff"

---

* Equal first authorship

Fig. 1. We propose a method to train an instance segmentation network from weak annotations in the form of bounding-boxes and image-level tags. Our network can explain both "thing" and "stuff" classes in the image, and does not produce overlapping instances as common detector-based approaches [23–25].

classes [14, 15], in contrast to most existing work in both semantic- and instance-segmentation which only consider "things".

We define the problem of instance segmentation as labelling every pixel in an image with both its object class and an instance identifier [16–18]. It is thus an extension of semantic segmentation, which only assigns each pixel an object class label. "Thing" classes (such as "person" and "car") are countable and are also studied extensively in object detection [19, 20]. This is because their finite extent makes it possible to annotate tight, well-defined bounding boxes around them. "Stuff" classes (such as "sky" and "vegetation"), on the other hand, are amorphous regions of homogeneous or repetitive textures [14]. As these classes have ambiguous boundaries and no well-defined shape they are not appropriate to annotate with bounding boxes [21]. Since "stuff" classes are not countable, we assume that all pixels of a stuff category belong to the same, single instance. Recently, this task of jointly segmenting "things" and "stuff" at an instance-level has also been named "Panoptic Segmentation" by [22].

Note that many popular instance segmentation algorithms which are based on object detection architectures [23–27] are not suitable for this task, as also noted by [22]. These methods output a ranked list of proposed instances, where the different proposals are allowed to overlap each other as each proposal is processed independently of the other. Consequently, these architectures are not suitable where each pixel in the image has to be explained, and assigned a unique label of either a "thing" or "stuff" class as shown in Fig. 1. This is in contrast to other instance segmentation methods such as [16, 28–31].

In this work, we use weak bounding box annotations for "thing" classes, and image-level tags for "stuff" classes. Whilst there are many previous works on semantic segmentation from image-level labels, the best performing ones [10, 32–34] used a saliency prior. The salient parts of an image are "thing" classes in popular saliency datasets [35–37] and this prior therefore does not help at all in segmenting "stuff" as in our case. We also consider the "semi-supervised" case where we have a mixture of weak- and fully-labelled annotations.

To our knowledge, this is the first work which performs weakly-supervised, non-overlapping instance segmentation, allowing our model to explain all "thing" and "stuff"

pixels in the image (Fig. 1). Furthermore, our model jointly produces semantic- and instance-segmentations of the image, which to our knowledge is the first time such a model has been trained in a weakly-supervised manner. Moreover, to our knowledge, this is the first work to perform either weakly supervised semantic- or instance-segmentation on the Cityscapes dataset. On Pascal VOC, our method achieves about 95% of fully-supervised accuracy on both semantic- and instance-segmentation. Furthermore, we surpass the state-of-the-art on fully-supervised instance segmentation as well. Finally, we use our weakly- and semi-supervised framework to examine how model performance varies with the number of examples in the training set and the annotation quality of each example, with the aim of helping dataset creators better understand the trade-offs they face in this context.

2   Related Work

Instance segmentation is a popular area of scene understanding research. Most top-performing algorithms modify object detection networks to output a ranked list of segments instead of boxes [23–27, 38]. However, all of these methods process each instance independently and thus overlapping instances are produced – one pixel can be assigned to multiple instances simultaneously. Additionally, object detection based architectures are not suitable for labelling "stuff" classes which cannot be described well by bounding boxes [21]. These limitations, common to all of these methods, have also recently been raised by Kirillov *et al.* [22]. We observe, however, that there are other instance segmentation approaches based on initial semantic segmentation networks [16, 28–30] which do not produce overlapping instances and can naturally handle "stuff" classes. Our proposed approach extends methods of this type to work with weaker supervision.

Although prior work on weakly-supervised instance segmentation is limited, there are many previous papers on weak semantic segmentation, which is also relevant to our task. Early work in weakly-supervised semantic segmentation considered cases where images were only partially labelled using methods based on Conditional Random Fields (CRFs) [39, 40]. Subsequently, many approaches have achieved high accuracy using only image-level labels [9, 10, 41, 42], bounding boxes [43, 11, 12], scribbles [21] and points [13]. A popular paradigm for these works is "self-training" [44]: a model is trained in a fully-supervised manner by generating the necessary ground truth with the model itself in an iterative, Expectation-Maximisation (EM)-like procedure [11, 12, 21, 42]. Such approaches are sensitive to the initial, approximate ground truth which is used to bootstrap training of the model. To this end, Khoreva *et al.* [43] showed how, given bounding box annotations, carefully chosen unsupervised foreground-background and segmentation-proposal algorithms could be used to generate high-quality approximate ground truth such that iterative updates to it were not required thereafter.

Our work builds on the "self-training" approach to perform instance segmentation. To our knowledge, only Khoreva *et al.* [43] have published results on weakly-supervised instance segmentation. However, the model used by [43] was not competitive with the existing instance segmentation literature in a fully-supervised setting. Moreover, [43] only considered bounding-box supervision, whilst we consider image-level labels as well. Recent work by [45] modifies Mask-RCNN [23] to train it using fully-labelled

examples of some classes, and only bounding box annotations of others. Our proposed method can also be used in a semi-supervised scenario (with a mixture of fully- and weakly-labelled training examples), but unlike [45], our approach works with only weak supervision as well. Furthermore, in contrast to [43] and [45], our method does not produce overlapping instances, handles "stuff" classes and can thus explain every pixel in an image as shown in Fig. 1.

3   Proposed Approach

We first describe how we generate approximate ground truth data to train semantic- and instance-segmentation models with in Sec. 3.1 through 3.4. Thereafter, in Sec. 3.5, we discuss the network architecture that we use.

3.1   Training with weaker supervision

In a fully-supervised setting, semantic segmentation models are typically trained by performing multinomial logistic regression independently for each pixel in the image. The loss function, the cross entropy between the ground-truth distribution and the prediction, can be written as

$$L = -\sum_{i \in \Omega} \log p(l_i | \mathbf{I}) \tag{1}$$

where l_i is the ground-truth label at pixel i, $p(l_i|\mathbf{I})$ is the probability (obtained from a softmax activation) predicted by the neural network for the correct label at pixel i of image $\mathbf{I}$ and Ω is the set of pixels in the image.
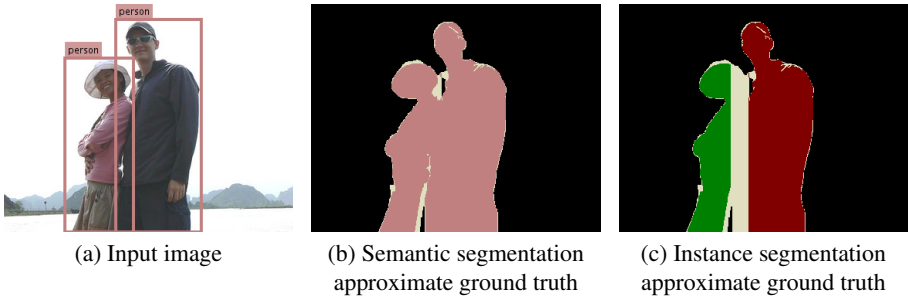
In the weakly-supervised scenarios considered in this paper, we do not have reliable annotations for all pixels in Ω. Following recent work [43, 9, 13, 42], we use our weak supervision and image priors to approximate the ground-truth for a subset $\Omega' \subset \Omega$ of the pixels in the image. We then train our network using the estimated labels of this smaller subset of pixels. Section 3.2 describes how we estimate Ω' and the corresponding labels for images with only bounding-box annotations, and Sec. 3.3 for image-level tags.

Our approach to approximating the ground truth is based on the principle of only assigning labels to pixels which we are confident about, and marking the remaining set of pixels, $\Omega \setminus \Omega'$, as "ignore" regions over which the loss is not computed. This is motivated by Bansal *et al.* [46] who observed that sampling only 4% of the pixels in the image for computing the loss during fully-supervised training yielded about the same results as sampling all pixels, as traditionally done. This supported their hypothesis that most of the training data for a pixel-level task is statistically correlated within an image, and that randomly sampling a much smaller set of pixels is sufficient. Moreover, [47] and [48] showed improved results by respectively sampling only 6% and 12% of the hardest pixels, instead of all of them, in fully-supervised training.

3.2   Approximate ground truth from bounding box annotations

We use GrabCut [49] (a classic foreground segmentation technique given a bounding-box prior) and MCG [50] (a segment-proposal algorithm) to obtain a foreground mask

(a) Input image   (b) Semantic segmentation approximate ground truth   (c) Instance segmentation approximate ground truth

Fig. 2. An example of generating approximate ground truth from bounding box annotations for an image (a). A pixel is labelled the with the bounding-box label if it belongs to the foreground masks of both GrabCut [49] and MCG [50] (b). Approximate instance segmentation ground truth is generated using the fact that each bounding box corresponds to an instance (c). Grey regions are "ignore" labels over which the loss is not computed due to ambiguities in label assignment.

from a bounding-box annotation, following [43]. To achieve high precision in this approximate labelling, a pixel is only assigned to the object class represented by the bounding box if both GrabCut and MCG agree (Fig. 2).
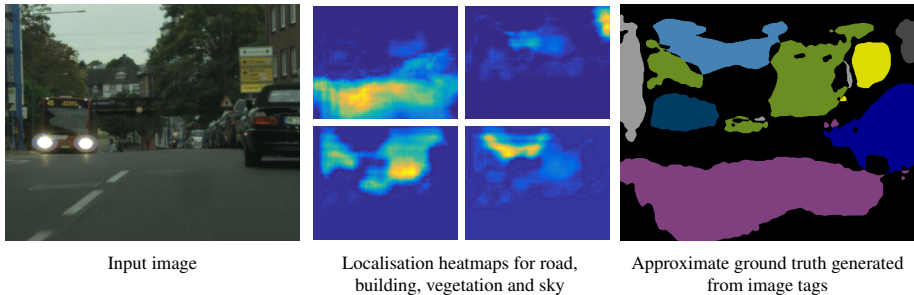
Note that the final stage of MCG uses a random forest trained with pixel-level supervision on Pascal VOC to rank all the proposed segments. We do not perform this ranking step, and obtain a foreground mask from MCG by selecting the proposal that has the highest Intersection over Union (IoU) with the bounding box annotation.

This approach is used to obtain labels for both semantic- and instance-segmentation as shown in Fig. 2. As each bounding box corresponds to an instance, the foreground for each box is the annotation for that instance. If the foreground of two bounding boxes of the same class overlap, the region is marked as "ignore" as we do not have enough information to attribute it to either instance.

3.3 Approximate ground-truth from image-level annotations

When only image-level tags are available, we leverage the fact that CNNs trained for image classification still have localisation information present in their convolutional layers [51]. Consequently, when presented with a dataset of only images and their tags, we first train a network to perform multi-label classification. Thereafter, we extract weak localisation cues for all the object classes that are present in the image (according to the image-level tags). These localisation heatmaps (as shown in Fig. 3) are thresholded to obtain the approximate ground-truth for a particular class. It is possible for localisation heatmaps for different classes to overlap. In this case, thresholded heatmaps occupying a smaller area are given precedence. We found this rule, like [9], to be effective in preventing small or thin objects from being missed.

Though this approach is independent of the weak localisation method used, we used Grad-CAM [52]. Grad-CAM is agnostic to the network architecture unlike CAM [51] and also achieves better performance than Excitation BP [53] on the ImageNet localisation task [4].

Input image   Localisation heatmaps for road, building, vegetation and sky   Approximate ground truth generated from image tags

Fig. 3. Approximate ground truth generated from image-level tags using weak localisation cues from a multi-label classification network. Cluttered scenes from Cityscapes with full "stuff" annotations makes weak localisation more challenging than Pascal VOC and ImageNet that only have "things" labels. Black regions are labelled "ignore". Colours follow Cityscapes convention.

Input Image   Iteration 0   Iteration 2   Iteration 5   Ground truth
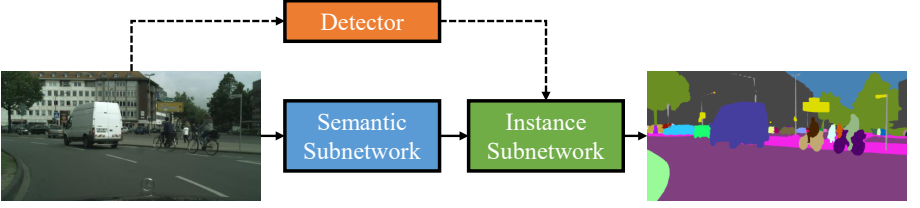
Fig. 4. By using the output of the trained network, the initial approximate ground truth produced according to Sec. 3.2 and 3.3 (Iteration 0) can be improved. Black regions are "ignore" labels over which the loss is not computed in training. Note for instance segmentation, permutations of instance labels of the same class are equivalent.

We cannot differentiate different instances of the same class from only image tags as the number of instances is unknown. This form of weak supervision is thus appropriate for "stuff" classes which cannot have multiple instances. Note that saliency priors, used by many works such as [10, 32, 33] on Pascal VOC, are not suitable for "stuff" classes as popular saliency datasets [35–37] only consider "things" to be salient.

3.4   Iterative ground truth approximation

The ground truth approximated in Sec. 3.2 and 3.3 can be used to train a network from random initialisation. However, the ground truth can subsequently be iteratively refined by using the outputs of the network on the training set as the new approximate ground truth as shown in Fig 4. The network's output is also post-processed with DenseCRF [54] using the parameters of Deeplab [55] (as also done by [9, 43]) to improve the predictions at boundaries. Moreover, any pixel labelled a "thing" class that is outside the bounding-box of the "thing" class is set to "ignore" as we are certain that a pixel for a thing class cannot be outside its bounding box. For a dataset such as Pascal VOC, we can set these pixels to be "background" rather than "ignore". This is because "background" is the only "stuff" class in the dataset.

Fig. 5. Overview of the network architecture. An initial semantic segmentation is partitioned into an instance segmentation, using the output of an object detector as a cue. Dashed lines indicate paths which are not backpropagated through during training.

3.5 Network Architecture

Using the approximate ground truth generation method described in this section, we can train a variety of segmentation models. Moreover, we can trivially combine this with full human-annotations to operate in a semi-supervised setting. We use the architecture of Arnab *et al.* [16] as it produces both semantic- and instance-segmentations, and can be trained end-to-end, given object detections. This network consists of a semantic segmentation subnetwork, followed by an instance subnetwork which partitions the initial semantic segmentation into an instance segmentation with the aid of object detections, as shown in Fig. 5.

We denote the output of the first module, which can be any semantic segmentation network, as $\mathbf{Q}$ where $Q_i(l)$ is the probability of pixel i of being assigned semantic label l. The instance subnetwork has two inputs – $\mathbf{Q}$ and a set of object detections for the image. There are D detections, each of the form (l_d, s_d, B_d) where l_d is the detected class label, $s_d \in [0, 1]$ the score and B_d the set of pixels lying within the bounding box of the dth detection. This model assumes that each object detection represents a possible instance, and it assigns every pixel in the initial semantic segmentation an instance label using a Conditional Random Field (CRF). This is done by defining a multinomial random variable, X_i, at each of the N pixels in the image, with $\mathbf{X} = [X_1, X_2 \ldots, X_N]^\top$. This variable takes on a label from the set $\{1, \ldots, D\}$ where D is the number of detections. This formulation ensures that each pixel can only be assigned one label. The energy of the assignment $\mathbf{x}$ to all instance variables $\mathbf{X}$ is then defined as

$$E(\mathbf{X} = \mathbf{x}) = -\sum_{i}^{N} \ln \left(w_1 \psi_{Box}(x_i) + w_2 \psi_{Global}(x_i) + \epsilon \right) + \sum_{i<j}^{N} \psi_{Pairwise}(x_i, x_j). \tag{2}$$

The first unary term, the box term, encourages a pixel to be assigned to the instance represented by a detection if it falls within its bounding box,

$$\psi_{Box}(X_i = k) = \begin{cases} s_k Q_i(l_k) & \text{if } i \in B_k \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

Note that this term is robust to false-positive detections [16] since it is low if the semantic segmentation at pixel i, $Q_i(l_k)$ does not agree with the detected label, l_k. The global

term,

$$\psi_{Global}(X_i = k) = Q_i(l_k), \tag{4}$$

is independent of bounding boxes and can thus overcome errors in mislocalised bounding boxes not covering the whole instance. Finally, the pairwise term is the common densely-connected Gaussian and bilateral filter [54] encouraging appearance and spatial consistency.

In contrast to [16], we also consider stuff classes (which object detectors are not trained for), by simply adding "dummy" detections covering the whole image with a score of 1 for all stuff classes in the dataset. This allows our network to jointly segment all "things" and "stuff" classes at an instance level. As mentioned before, the box and global unary terms are not affected by false-positive detections arising from detections for classes that do not correspond to the initial semantic segmentation **Q**. The Maximum-a-Posteriori (MAP) estimate of the CRF is the final labelling, and this is obtained by using mean-field inference, which is formulated as a differentiable, recurrent network [56, 57].

We first train the semantic segmentation subnetwork using a standard cross-entropy loss with the approximate ground truth described in Sec 3.2 and 3.3. Thereafter, we append the instance subnetwork and finetune the entire network end-to-end. For the instance subnetwork, the loss function must take into account that different permutations of the same instance labelling are equivalent. As a result, the ground truth is "matched" to the prediction before the cross-entropy loss is computed as described in [16].

4   Experimental Evaluation

4.1   Experimental Set-up

Datasets and weak supervision We evaluate on two standard segmentation datasets, Pascal VOC [19] and Cityscapes [6]. Our weakly- and fully-supervised experiments are trained with the same images, but in the former case, pixel-level ground truth is approximated as described in Sec. 3.1 through 3.4.

Pascal VOC has 20 "thing" classes annotated, for which we use bounding box supervision. There is a single "background" class for all other object classes. Following common practice on this dataset, we utilise additional images from the SBD dataset [58] to obtain a training set of 10582 images. In some of our experiments, we also use 54000 images from Microsoft COCO [20] only for the initial pretraining of the semantic subnetwork. We evaluate on the validation set, of 1449 images, as the evaluation server is not available for instance segmentation.

Cityscapes has 8 "thing" classes, for which we use bounding box annotations, and 11 "stuff" class labels for which we use image-level tags. We train our initial semantic segmentation model with the images for which 19998 coarse and 2975 fine annotations are available. Thereafter, we train our instance segmentation network using the 2975 images with fine annotations available as these have instance ground truth labelled. Details of the multi-label classification network we trained in order to obtain weak localisation cues from image-level tags (Sec. 3.3) are described in the supplementary. When using Grad-CAM, the original authors originally used a threshold of 15% of

the maximum value for weak localisation on ImageNet. However, we increased the threshold to 50% to obtain higher precision on this more cluttered dataset.

Network training Our underlying segmentation network is a reimplementation of PSP-Net [59]. For fair comparison to our weakly-supervised model, we train a fully-supervised model ourselves, using the same training hyperparameters (detailed in the supplementary) instead of using the authors' public, fully-supervised model. The original PSP-Net implementation [59] used a large batch size synchronised over 16 GPUs, as larger batch sizes give better estimates of batch statistics used for batch normalisation [59, 60]. In contrast, our experiments are performed on a single GPU with a batch size of one 521×521 image crop. As a small batch size gives noisy estimates of batch statistics, our batch statistics are "frozen" to the values from the ImageNet-pretrained model as common practice [61, 62]. Our instance subnetwork requires object detections, and we train Faster-RCNN [3] for this task. All our networks use a ResNet-101 [1] backbone.

Evaluation Metrics We use the AP^r metric [38], commonly used in evaluating instance segmentation. It extends the AP, a ranking metric used in object detection [19], to segmentation where a predicted instance is considered correct if its Intersection over Union (IoU) with the ground truth instance is more than a certain threshold. We also report the AP^r_{vol} which is the mean AP^r across a range of IoU thresholds. Following the literature, we use a range of 0.1 to 0.9 in increments of 0.1 on VOC, and 0.5 to 0.95 in increments of 0.05 on Cityscapes.

However, as noted by several authors [63, 16, 28, 22], the AP^r is a ranking metric that does not penalise methods which predict more instances than there actually are in the image as long as they are ranked correctly. Moreover, as it considers each instance independently, it does not penalise overlapping instances. As a result, we also report the Panoptic Quality (PQ) recently proposed by [22],

$$\mathrm{PQ} = \underbrace{\frac{\sum_{(p,g) \in TP} \mathrm{IoU}(p, g)}{|TP|}}_{\text{Segmentation Quality (SQ)}} \times \underbrace{\frac{|TP|}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|}}_{\text{Detection Quality (DQ)}}, \quad (5)$$

where p and g are the predicted and ground truth segments, and TP, FP and FN respectively denote the set of true positives, false positives and false negatives.

4.2 Results on Pascal VOC

Tables 1 and 2 show the state-of-art results of our method for semantic- and instance-segmentation respectively. For both semantic- and instance-segmentation, our weakly supervised model obtains about 95% of the performance of its fully-supervised counterpart, emphasising that accurate models can be learned from only bounding box annotations, which are significantly quicker and cheaper to obtain than pixelwise annotations. Table 2 also shows that our weakly-supervised model outperforms some recent fully supervised instance segmentation methods such as [17] and [65]. Moreover, our fully-supervised instance segmentation model outperforms all previous work on this dataset. The main difference of our model to [16] is that our network is based on the PSPNet architecture using ResNet-101, whilst [16] used the network of [66] based on VGG [2].

Table 1. Comparison of semantic segmentation performance to recent methods using only weak, bounding-box supervision on Pascal VOC. Note that [12] and [11] use the less accurate VGG network, whilst we and [43] use ResNet-101. "FS%" denotes the percentage of fully-supervised performance.

Method	Validation set			Test set		
	IoU (weak)	IoU (full)	FS%	IoU (weak)	IoU (full)	FS%
Without COCO annotations						
BoxSup [12]	62.0	63.8	**97.2**	64.6	–	–
Deeplab WSSL [11]	60.6	67.6	89.6	62.2	70.3	88.5
SDI [43]	69.4	74.5	93.2	–	–	–
Ours	**74.3**	**77.3**	96.1	**75.5**	**78.6**	**96.3**
With COCO annotations						
SDI [43]	74.2	77.7	95.5	–	–	–
Ours	**75.7**	**79.0**	**95.8**	**76.7**	**79.4**	**96.6**

We can obtain semantic segmentations from the output of our semantic subnetwork, or from the final instance segmentation (as we produce non-overlapping instances) by taking the union of all instances which have the same semantic label. We find that the IoU obtained from the final instance segmentation, and the initial pretrained semantic subnetwork to be very similar, and report the latter in Tab.1. Further qualitative and quantitative results, including success and failure cases, are included in the supplement.

End-to-end training of instance subnetwork Our instance subnetwork can be trained in a piecewise fashion, or the entire network including the semantic subnetwork can be trained end-to-end. End-to-end training was shown to obtain higher performance by [16] for full supervision. We also observe this effect for weak supervision from bounding box annotations. A weakly supervised model, trained with COCO annotations improves from an AP^r_{vol} of 53.3 to 55.5. When not using COCO for training the initial semantic subnetwork, a slightly higher increase by 3.9 from 51.7 is observed. This emphasises that our training strategy (Sec. 3.1) is effective for both semantic- and instance-segmentation.

Iterative training The approximate ground truth used to train our model can also be generated in an iterative manner, as discussed in Sec. 3.4. However, as the results from a single iteration (Tab. 1 and 2) are already very close to fully-supervised performance, this offers negligible benefit. Iterative training is, however, crucial for obtaining good results on Cityscapes as discussed in Sec. 4.3.

Semi-Supervision We also consider the case where we have a combination of weak and full annotations. As shown in Tab. 3, we consider all combinations of weak- and full-supervision of the training data from Pascal VOC and COCO. Table 3 shows that training with fully-supervised data from COCO and weakly-supervised data from VOC performs about the same as weak supervision from both datasets for both semantic- and instance-segmentation. Furthermore, training with fully annotated VOC data and weakly labelled COCO data obtains similar results to full supervision from both datasets.

Table 2. Comparison of instance segmentation performance to recent (fully- and weakly-supervised) methods on the VOC 2012 validation set.

| Method | AP^r | | | | | AP^r_{vol} | PQ |
	0.5	0.6	0.7	0.8	0.9		
Weakly supervised without COCO							
SDI [43]	44.8	–	–	–	–	–	–
Ours	**60.5**	**55.2**	**47.8**	**37.6**	**21.6**	**55.6**	**59.0**
Fully supervised without COCO							
SDS [38]	43.8	34.5	21.3	8.7	0.9	–	–
Chen *et al.* [64]	46.3	38.2	27.0	13.5	2.6	–	–
PFN [65]	58.7	51.3	42.5	31.2	15.7	52.3	–
Ours (fully supervised)	**63.6**	**59.5**	**53.8**	**44.7**	**30.2**	**59.2**	**62.7**
Weakly supervised with COCO							
SDI [43]	46.4	–	–	–	–	–	–
Ours	**60.9**	**55.9**	**48.0**	**37.2**	**21.7**	**55.5**	**59.5**
Fully supervised with COCO							
Arnab *et al.* [17]	58.3	52.4	45.4	34.9	20.1	53.1	–
MPA [27]	62.1	56.6	47.4	36.1	18.5	56.5	–
Arnab *et al.* [16]	61.7	55.5	48.6	39.5	25.1	57.5	–
SGN [31]	61.4	55.9	49.9	42.1	26.9	–	–
Ours (fully supervised)	**63.9**	**59.3**	**54.3**	**45.4**	**30.2**	**59.5**	**63.1**

We have qualitatively observed that the annotations in Pascal VOC are of higher quality than those of Microsoft COCO (random samples from both datasets are shown in the supplementary). And this intuition is evident in the fact that there is not much difference between training with weak or full annotations from COCO. This suggests that in the case of segmentation, per-pixel labelling of additional images is not particularly useful if they are not labelled to a high standard, and that labelling fewer images at a higher quality (Pascal VOC) is more beneficial than labelling many images at a lower quality (COCO). This is because Tab. 3 demonstrates how both semantic- and instance-segmentation networks can be trained to achieve similar performance by using only bounding box labels instead of low-quality segmentation masks. The average annotation time can be considered a proxy for segmentation quality. While a COCO instance took an average of 79 seconds to segment [20], this figure is not mentioned for Pascal VOC [19, 67].

4.3   Results on Cityscapes

Tables 4 and 5 present, what to our knowledge is, the first weakly supervised results for either semantic or instance segmentation on Cityscapes. Table 4 shows that, as expected for semantic segmentation, our weakly supervised model performs better, relative to the fully-supervised model, for "thing" classes compared to "stuff" classes. This is because we have more informative bounding box labels for "things", compared to only image-level tags for "stuff". For semantic segmentation, we obtain about 97% of

Table 3. Semantic- and instance-segmentation performance on Pascal VOC with varying levels of supervision from the Pascal and COCO datasets. The former is measured by the IoU, and latter by the AP_{vol}^r and PQ.

Dataset		IoU	AP_{vol}^r	PQ
VOC	COCO			
Weak	Weak	75.7	55.5	59.5
Weak	Full	75.8	56.1	59.8
Full	Weak	77.5	58.9	62.7
Full	Full	79.0	59.5	63.1

Table 4. Semantic segmentation performance on the Cityscapes validation set. We use more informative, bounding-box annotations for "thing" classes, and this is evident from the higher IoU than on "stuff" classes for which we only have image-level tags.

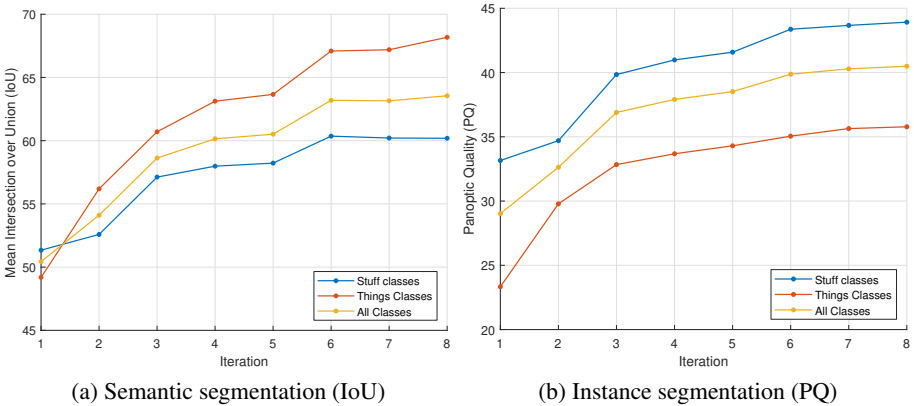| Method	IoU (weak)	IoU (full)	FS%
Ours (thing classes)	68.2	70.4	96.9
Ours (stuff classes)	60.2	72.4	83.1
Ours (overall)	63.6	71.6	88.8

Table 5. Instance-level segmentation results on Cityscapes. On the validation set, we report results for both "thing" (th.) and "stuff" (st.) classes. The online server, which evaluates the test set, only computes the AP^r for "thing" classes. We compare to other fully-supervised methods which produce non-overlapping instances. To our knowledge, no published work has evaluated on both "thing" and "stuff" classes. Our fully supervised model, initialised from the public PSPNet model [59] is equivalent to our previous work [16], and competitive with the state-of-art. Note that we cannot use the public PSPNet pretrained model in a weakly-supervised setting.

| Method | Validation | | | | | | Test |
	AP_{vol}^r th.	AP_{vol}^r st.	AP_{vol}^r all	PQ th.	PQ st.	PQ all	AP th.
Ours (weak, ImageNet init.)	17.0	33.1	26.3	35.8	43.9	40.5	12.8
Ours (full, ImageNet init.)	24.3	42.6	34.9	39.6	52.9	47.3	18.8
Ours (full, PSPNet init.) [16]	28.6	52.6	42.5	42.5	62.1	53.8	23.4
Pixel Encoding [68]	9.9	–	–	–	–	–	8.9
RecAttend [69]	–	–	–	–	–	–	9.5
InstanceCut [30]	–	–	–	–	–	–	13.0
DWT [28]	21.2	–	–	–	–	–	19.4
SGN [31]	29.2	–	–	–	–	–	25.0

fully-supervised performance for "things" (similar to our results on Pascal VOC) and 83% for "stuff". Note that we evaluate images at a single-scale, and higher absolute scores could be obtained by multi-scale ensembling [59, 61].

For instance-level segmentation, the fully-supervised ratios for the PQ are similar to the IoU ratio for semantic segmentation. In Tab. 5, we report the AP_{vol}^r and PQ for both thing and stuff classes, assuming that there is only one instance of a "stuff" class in the image if it is present. Here, the AP_{vol}^r for "stuff" classes is higher than that for "things". This is because there can only be one instance of a "stuff" class, which makes instances easier to detect, particularly for classes such as "road" which typically occupy a large portion of the image. The Cityscapes evaluation server, and previous work on this dataset, only report the AP_{vol}^r for "thing" classes. As a result, we report results for "stuff" classes only on the validation set. Table 5 also compares our results to existing work which produces non-overlapping instances on this dataset, and shows that both our fully- and weakly-supervised models are competitive with recently pub-

(a) Semantic segmentation (IoU)   (b) Instance segmentation (PQ)

Fig. 6. Iteratively refining our approximate ground truth during training improves both semantic and instance segmentation on the Cityscapes validation set.

lished work on this dataset. We also include the results of our fully-supervised model, initialised from the public PSPNet model [59] released by the authors, and show that this is competitive with the state-of-art [31] among methods producing non-overlapping segmentations (note that [31] also uses the same PSPNet model). Figure 7 shows some predictions of our weakly supervised model; further results are in the supplementary.

Iterative training Iteratively refining our approximate ground truth during training, as described in Sec. 3.4, greatly improves our performance on both semantic- and instance-segmentation as shown in Fig. 6. We trained the network for 150 000 iterations before regenerating the approximate ground truth using the network's own output on the training set. Unlike on Pascal VOC, iterative training is necessary to obtain good performance on Cityscapes as the approximate ground truth generated on the first iteration is not sufficient to obtain high accuracy. This was expected for "stuff" classes, since we began from weak localisation cues derived from the image-level tags. However, as shown in Fig. 6, "thing" classes also improved substantially with iterative training, unlike on Pascal VOC where there was no difference. Compared to VOC, Cityscapes is a more cluttered dataset, and has large scale variations as the distance of an object from the car-mounted camera changes. These dataset differences may explain why the image priors employed by the methods we used (GrabCut [49] and MCG [50]) to obtain approximate ground truth annotations from bounding boxes are less effective. Furthermore, in contrast to Pascal VOC, Cityscapes has frequent co-occurences of the same objects in many different images, making it more challenging for weakly supervised methods.

Effect of ranking methods on the AP^r The AP^r metric is a ranking metric derived from object detection. It thus requires predicted instances to be scored such that they are ranked in the correct relative order. As our network uses object detections as an additional input and each detection represents a possible instance, we set the score of a predicted instance to be equal to the object detection score. For the case of stuff classes,

Table 6. The effect of different instance ranking methods on the AP_{vol}^r of our weakly supervised model computed on the Cityscapes validation set.

Ranking Method	AP_{vol}^r th.	AP_{vol}^r st.	PQ all
Detection score	17.0	26.7	40.5
Mean seg. confidence	14.6	33.1	40.5
Oracle	21.6	37.0	40.5

Fig. 7. Example results on Cityscapes of our weakly supervised model.

which object detectors are not trained for, we use a constant detection score of 1 as described in Sec. 3.5. An alternative to using a constant score for "stuff" classes is to take the mean of the softmax-probability of all pixels within the segmentation mask. Table 6 shows that this latter method improves the AP^r for stuff classes. For "things", ranking with the detection score performs better and comes closer to oracle performance which is the maximum AP^r that could be obtained with the predicted instances.

Changing the score of a segmented instance does not change the quality of the actual segmentation, but does impact the AP^r greatly as shown in Tab. 6. The PQ, which does not use scores, is unaffected by different ranking methods, and this suggests that it is a better metric for evaluating non-overlapping instance segmentation where each pixel in the image is explained.

5   Conclusion and Future Work

We have presented, to our knowledge, the first weakly-supervised method that jointly produces non-overlapping instance and semantic segmentation for both "thing" and "stuff" classes. Using only bounding boxes, we are able to achieve 95% of state-of-art fully-supervised performance on Pascal VOC. On Cityscapes, we use image-level annotations for "stuff" classes and obtain 88.8% of fully-supervised performance for semantic segmentation and 85.6% for instance segmentation (measured with the PQ). Crucially, the weak annotations we use incur only about 3% of the time of full labelling. As annotating pixel-level segmentation is time consuming, there is a dilemma between labelling few images with high quality or many images with low quality. Our semi-supervised experiment suggests that the latter is not an effective use of annotation budgets as similar performance can be obtained from only bounding-box annotations.

Future work is to perform instance segmentation using only image-level tags and the number of instances of each object present in the image as supervision. This will require a network architecture that does not use object detections as an additional input.

References

1. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. (2016)
2. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR. (2015)
3. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: NIPS. (2015)
4. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. IJCV (2015)
5. Sun, C., Shrivastava, A., Singh, S., Gupta, A.: Revisiting unreasonable effectiveness of data in deep learning era. In: ICCV, IEEE (2017) 843–852
6. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: CVPR. (2016)
7. Papadopoulos, D.P., Uijlings, J.R., Keller, F., Ferrari, V.: Extreme clicking for efficient object annotation. In: ICCV, IEEE (2017) 4940–4949
8. Papadopoulos, D.P., Clarke, A.D., Keller, F., Ferrari, V.: Training object class detectors from eye tracking data. In: ECCV, Springer (2014) 361–376
9. Kolesnikov, A., Lampert, C.H.: Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In: ECCV. (2016)
10. Wei, Y., Feng, J., Liang, X., Cheng, M.M., Zhao, Y., Yan, S.: Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In: CVPR. (2017)
11. Papandreou, G., Chen, L., Murphy, K., Yuille, A.L.: Weakly- and semi-supervised learning of a DCNN for semantic image segmentation. In: ICCV. (2015)
12. Dai, J., He, K., Sun, J.: Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In: ICCV. (2015)
13. Bearman, A., Russakovsky, O., Ferrari, V., Fei-Fei, L.: What's the point: Semantic segmentation with point supervision. In: ECCV. (2016)
14. Forsyth, D.A., Malik, J., Fleck, M.M., Greenspan, H., Leung, T., Belongie, S., Carson, C., Bregler, C.: Finding pictures of objects in large collections of images. Springer (1996)
15. Adelson, E.H.: On seeing stuff: the perception of materials by humans and machines. In: Human vision and electronic imaging VI. Volume 4299., International Society for Optics and Photonics (2001) 1–13
16. Arnab, A., Torr, P.H.S.: Pixelwise instance segmentation with a dynamically instantiated network. In: CVPR. (2017)
17. Arnab, A., Torr, P.H.S.: Bottom-up instance segmentation using deep higher-order crfs. In: BMVC. (2016)
18. Zhang, Z., Schwing, A.G., Fidler, S., Urtasun, R.: Monocular object instance segmentation and depth ordering with cnns. In: ICCV. (2015)
19. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. IJCV (2010)
20. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV. (2014)
21. Lin, D., Dai, J., Jia, J., He, K., Sun, J.: Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In: CVPR. (2016) 3159–3167
22. Kirillov, A., He, K., Girshick, R., Rother, C., Dollár, P.: Panoptic segmentation. In: arXiv preprint arXiv:1801.00868. (2018)

23. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: ICCV. (2017)
24. Dai, J., He, K., Sun, J.: Instance-aware semantic segmentation via multi-task network cascades. In: CVPR. (2016)
25. Li, Y., Qi, H., Dai, J., Ji, X., Wei, Y.: Fully convolutional instance-aware semantic segmentation. In: CVPR. (2017)
26. Liu, S., Qi, L., Qin, H., Shi, J., Jia, J.: Path aggregation network for instance segmentation. In: arXiv preprint arXiv:1803.01534. (2018)
27. Liu, S., Qi, X., Shi, J., Zhang, H., Jia, J.: Multi-scale patch aggregation (mpa) for simultaneous detection and segmentation. In: CVPR. (2016)
28. Bai, M., Urtasun, R.: Deep watershed transform for instance segmentation. In: CVPR, IEEE (2017) 2858–2866
29. De Brabandere, B., Neven, D., Van Gool, L.: Semantic instance segmentation with a discriminative loss function. In: CVPR Workshop. (2017)
30. Kirillov, A., Levinkov, E., Andres, B., Savchynskyy, B., Rother, C.: Instancecut: from edges to instances with multicut. In: CVPR. (2017)
31. Liu, S., Jia, J., Fidler, S., Urtasun, R.: Sgn: Sequential grouping networks for instance segmentation. In: ICCV. (2017)
32. Wei, Y., Liang, X., Chen, Y., Shen, X., Cheng, M.M., Feng, J., Zhao, Y., Yan, S.: Stc: A simple to complex framework for weakly-supervised semantic segmentation. PAMI **39**(11) (2017) 2314–2320
33. Oh, S.J., Benenson, R., Khoreva, A., Akata, Z., Fritz, M., Schiele, B.: Exploiting saliency for object segmentation from image level labels. In: CVPR. (2017)
34. Chaudhry, A., Dokania, P.K., Torr, P.H.: Discovering class-specific pixels for weakly-supervised semantic segmentation. In: BMVC. (2017)
35. Cheng, M.M., Mitra, N.J., Huang, X., Torr, P.H., Hu, S.M.: Global contrast based salient region detection. PAMI **37**(3) (2015) 569–582
36. Yang, C., Zhang, L., Lu, H., Ruan, X., Yang, M.H.: Saliency detection via graph-based manifold ranking. In: CVPR, IEEE (2013) 3166–3173
37. Shi, J., Yan, Q., Xu, L., Jia, J.: Hierarchical image saliency detection on extended cssd. PAMI **38**(4) (2016) 717–729
38. Hariharan, B., Arbeláez, P., Girshick, R., Malik, J.: Simultaneous detection and segmentation. In: ECCV. (2014)
39. Verbeek, J.J., Triggs, B.: Scene segmentation with crfs learned from partially labeled images. In: NIPS. (2008) 1553–1560
40. He, X., Zemel, R.S.: Learning hybrid models for image annotation with partially labeled data. In: NIPS. (2009) 625–632
41. Pinheiro, P.O., Collobert, R.: From image-level to pixel-level labeling with convolutional networks. In: CVPR. (2015)
42. Pathak, D., Krahenbuhl, P., Darrell, T.: Constrained convolutional neural networks for weakly supervised segmentation. In: ICCV. (2015)
43. Khoreva, A., Benenson, R., Hosang, J., Hein, M., Schiele, B.: Simple does it: Weakly supervised instance and semantic segmentation. In: CVPR. (2017)
44. Scudder, H.: Probability of error of some adaptive pattern-recognition machines. IEEE Transactions on Information Theory **11**(3) (1965) 363–371
45. Hu, R., Dollár, P., He, K., Darrell, T., Girshick, R.: Learning to segment every thing. In: arXiv preprint arXiv:1711.10370. (2017)
46. Bansal, A., Chen, X., Russell, B., Gupta, A., Ramanan, D.: Pixelnet: Representation of the pixels, by the pixels, and for the pixels. In: arXiv preprint arXiv:1702.06506. (2017)
47. Pohlen, T., Hermans, A., Mathias, M., Leibe, B.: Full-resolution residual networks for semantic segmentation in street scenes. In: CVPR. (2017)
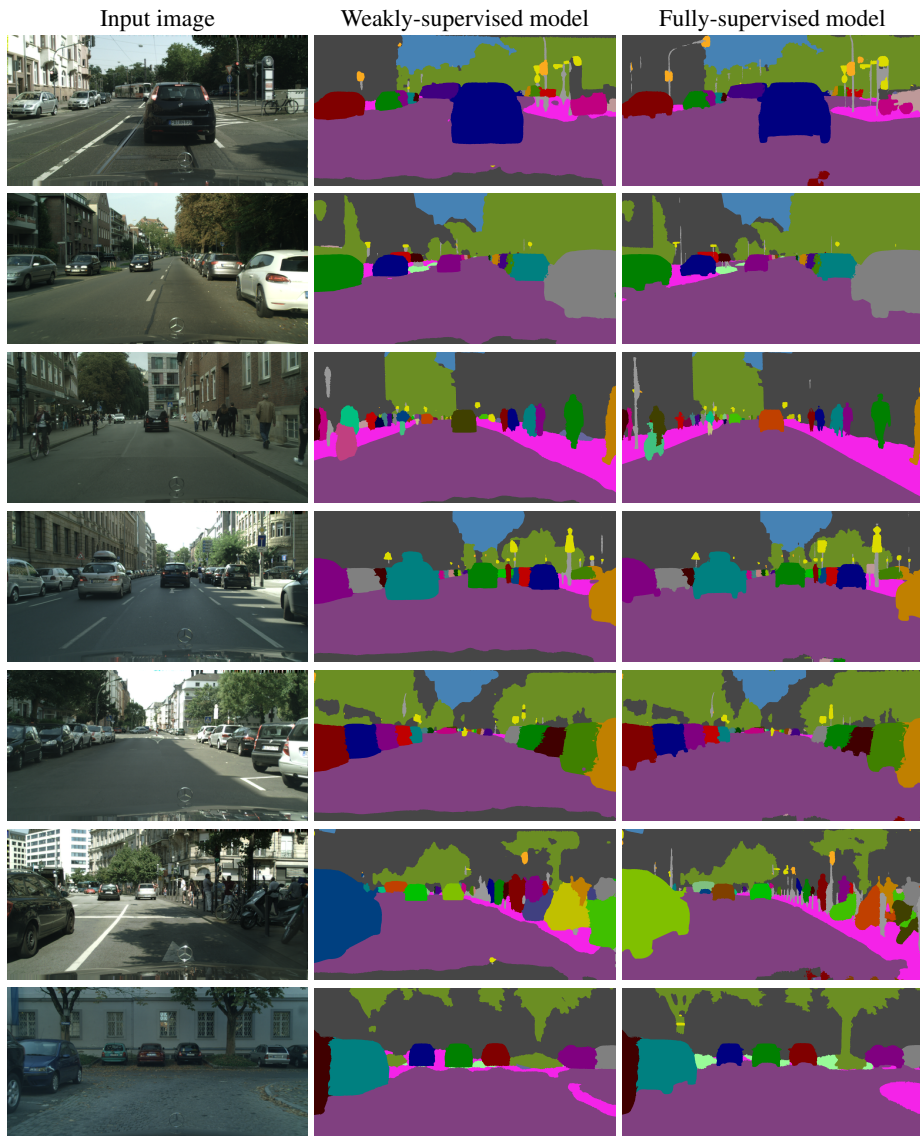
48. Li, Q., Arnab, A., Torr, P.H.: Holistic, instance-level human parsing. In: BMVC. (2017)
49. Rother, C., Kolmogorov, V., Blake, A.: Grabcut: Interactive foreground extraction using iterated graph cuts. ACM TOG (2004)
50. Arbelaez, P., Pont-Tuset, J., Barron, J., Marques, F., Malik, J.: Multiscale combinatorial grouping. In: CVPR. (2014)
51. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: CVPR, IEEE (2016) 2921–2929
52. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: ICCV. (2017)
53. Zhang, J., Lin, Z., Brandt, J., Shen, X., Sclaroff, S.: Top-down neural attention by excitation backprop. In: ECCV, Springer (2016) 543–559
54. Krähenbühl, P., Koltun, V.: Efficient inference in fully connected CRFs with Gaussian edge potentials. In: NIPS. (2011)
55. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic image segmentation with deep convolutional nets and fully connected crfs. ICLR (2015)
56. 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)
57. Arnab, A., Zheng, S., Jayasumana, S., Romera-Paredes, B., Larsson, M., Kirillov, A., Savchynskyy, B., Rother, C., Kahl, F., Torr, P.H.S.: Conditional random fields meet deep neural networks for semantic segmentation: Combining probabilistic graphical models with deep learning for structured prediction. IEEE Signal Processing Magazine **35**(1) (Jan 2018) 37–52
58. Hariharan, B., Arbeláez, P., Bourdev, L., Maji, S., Malik, J.: Semantic contours from inverse detectors. In: ICCV. (2011)
59. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: CVPR. (2017)
60. Chen, L.C., Papandreou, G., Schroff, F., Adam, H.: Rethinking atrous convolution for semantic image segmentation. In: arXiv preprint arXiv:1706.05587. (2017)
61. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. arXiv preprint arXiv:1606.00915v2 (2016)
62. Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., et al.: Speed/accuracy trade-offs for modern convolutional object detectors. In: CVPR. (2017)
63. Yang, Y., Hallman, S., Ramanan, D., Fowlkes, C.C.: Layered object models for image segmentation. PAMI (2012)
64. Chen, Y.T., Liu, X., Yang, M.H.: Multi-instance object segmentation with occlusion handling. In: CVPR. (2015)
65. Liang, X., Wei, Y., Shen, X., Yang, J., Lin, L., Yan, S.: Proposal-free network for instance-level object segmentation. arXiv preprint arXiv:1509.02636 (2015)
66. Arnab, A., Jayasumana, S., Zheng, S., Torr, P.H.S.: Higher order conditional random fields in deep neural networks. In: ECCV. (2016)
67. Everingham, M., Eslami, S.A., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes challenge: A retrospective. IJCV **111**(1) (2015)
68. Uhrig, J., Cordts, M., Franke, U., Brox, T.: Pixel-level encoding and depth layering for instance-level semantic labeling. In: GCPR. (2016)
69. Ren, M., Zemel, R.S.: End-to-end instance segmentation with recurrent attention. In: CVPR. (2017)
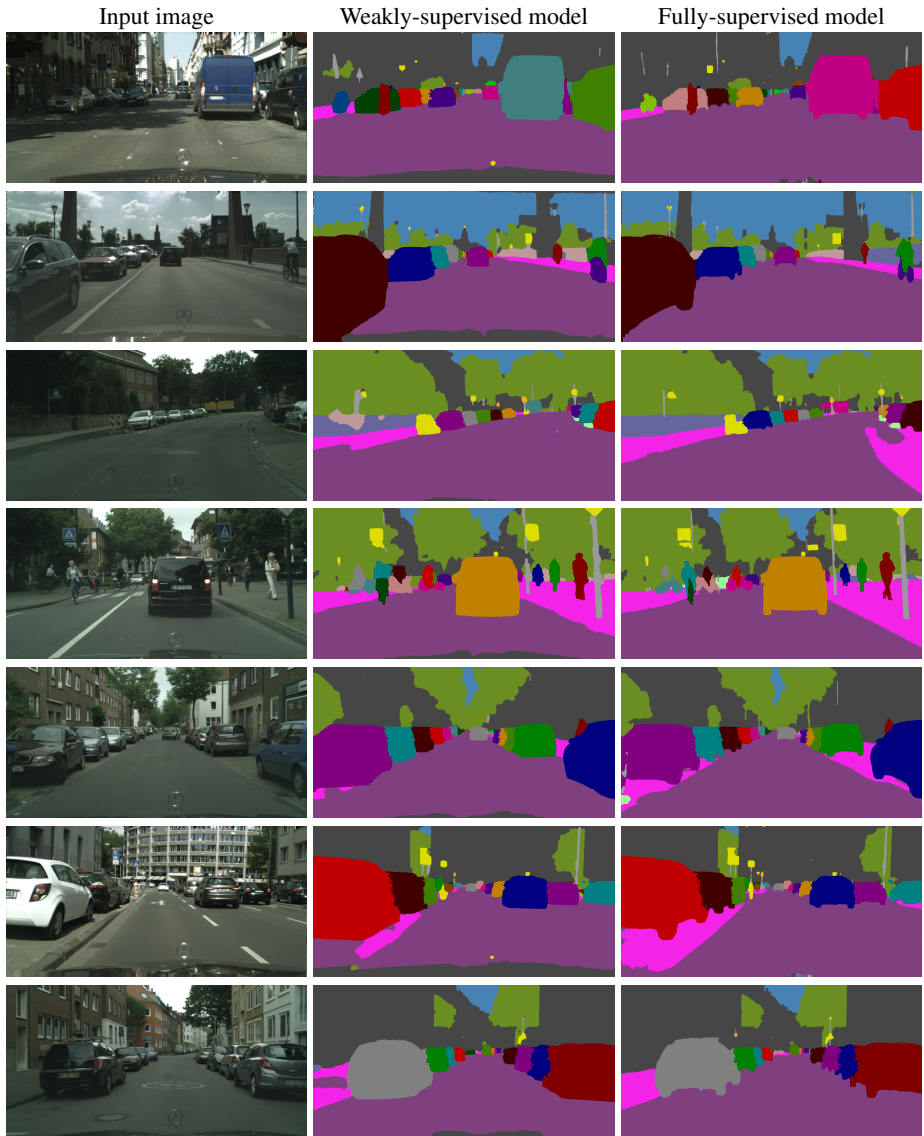
Appendix

Section A presents further qualitative and quantitative results of our experiments on Cityscapes and Pascal VOC. Section B describes the training of the networks described in the main paper. Section 4.2 of our paper mentioned that the annotation quality of Pascal VOC [19] is better than COCO [20]. Some randomly drawn images from these datasets are presented to illustrate this point in Sec. C. Finally, Sec. D shows our calculation of how much the overall annotation time is reduced by using weak annotations, in comparison to full annotations, on the Cityscapes dataset.

A   Additional Qualitative and Quantitative Results

Figure 7 and Tab. 7 present additional qualitative and quantitative results on the Cityscapes dataset. Similarly, Fig. 8 and Tab. 8 show additional results on the Pascal VOC dataset.
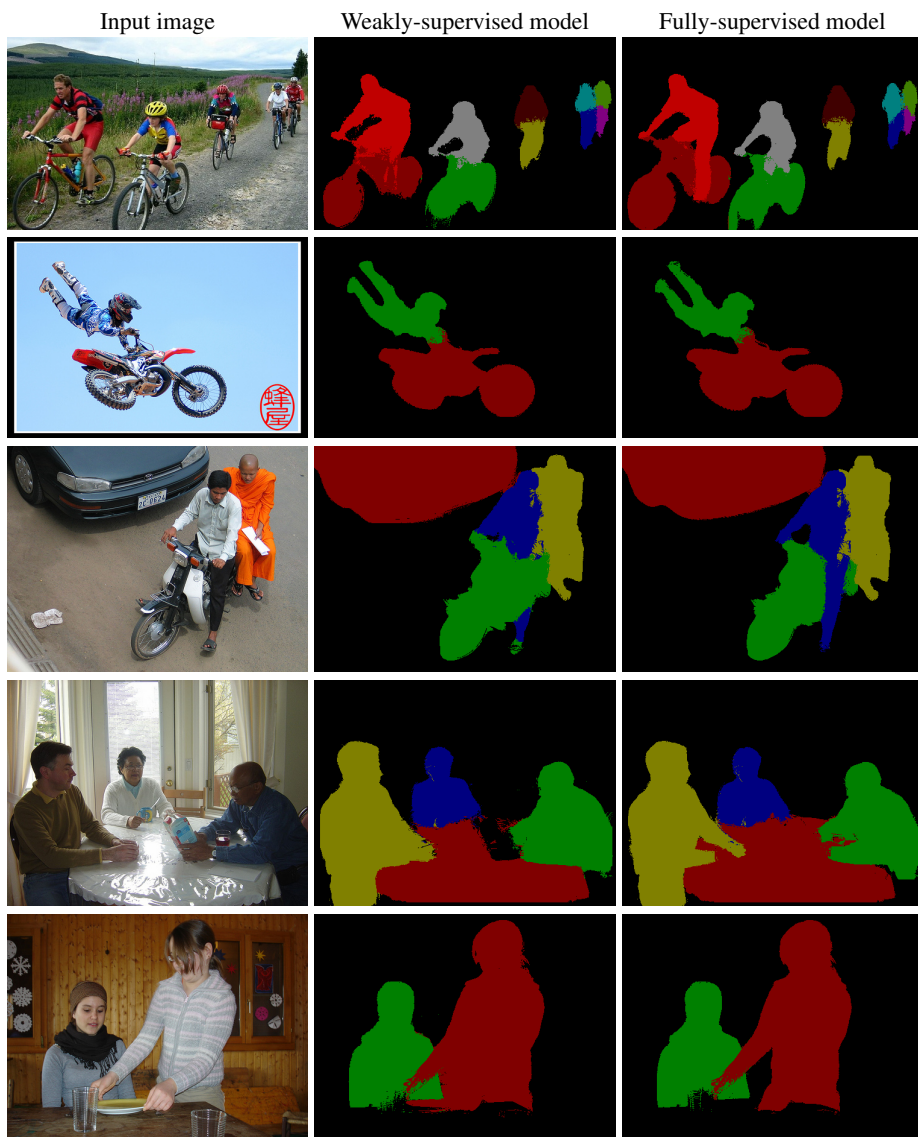
Fig. 7. Comparison of our weakly- and fully-supervised instance segmentation models on the Cityscapes dataset. The fully-supervised model produces more precise segmentations, as seen by its sharper boundaries. The last row also shows how the fully-supervised model segments "stuff" classes such as "vegetation" and "sidewalk" more accurately. Both of these were expected, as the weakly-supervised model is trained only with bounding box and image tag annotations. Rows 3 and 6 also show some instances with different colouring. Each colour represents an instance ID, and a discrepancy between the two indicates that a different number of instances were segmented.

Input image	Weakly-supervised model	Fully-supervised model

Fig. 7 cont. Comparison of our weakly- and fully-supervised instance segmentation models on the Cityscapes dataset. The last three rows show how the fully-supervised model is also able to segment "stuff" classes such as "sidewalk" more accurately. This was expected since the weakly-supervised model is only trained with image-level tags for "stuff" classes, which provides very little localisation information.

Fig. 8. Comparison of our weakly- and fully-supervised instance segmentation models on the Pascal VOC validation set. The weakly-supervised model typically obtains results similar to its state-of-the-art, fully-supervised counterpart. However, the fully-supervised model produces more accurate and precise segmentations, as seen in the last two rows.

Input image	Weakly-supervised model	Fully-supervised model

Fig. 8 cont. The first and second rows show examples where the results of the two models are similar. In the third and fourth rows, the weakly-supervised model does not segment the "green person" as well as the fully-supervised model. In the last row, both weakly- and fully-supervised models have made an error in not completely segmenting each of the bottles.

Table 7. Per-class results of our weakly- and fully-supervised models for both semantic and instance segmentation on the Cityscapes validation set. The IoU measures semantic segmentation performance, whilst the AP^r_{vol} and PQ measure instance segmentation performance.

Metric	Mean	road	side-walk	build-ing	wall	fence	pole	traffic light	traffic sign	vege-tation	terrain	sky	person	rider	car	truck	bus	train	motor-cycle	bi-cycle
Weakly supervised model																				
IoU	63.6	93.3	59.3	86.6	38.7	29.6	32.0	44.0	59.2	88.7	39.1	91.7	69.4	48.4	87.4	68.0	80.7	68.0	56.0	67.5
AP^r_{vol}	26.3	82.7	27.6	68.1	5.9	5.2	0.6	3.0	16.6	74.1	4.7	76.1	11.7	5.0	27.7	17.4	36.3	23.0	9.0	5.9
PQ	40.5	91.2	47.0	79.6	14.8	12.7	5.5	13.2	37.3	83.3	16.2	82.3	30.6	25.7	46.9	33.7	55.5	37.0	31.8	24.9
Fully supervised model																				
IoU	71.6	97.6	81.9	90.4	42.2	52.3	54.5	61.1	71.8	90.5	61.1	93.5	76.6	53.2	93.4	68.3	77.8	70.6	50.7	72.3
AP^r_{vol}	34.9	94.8	56.2	73.6	10.5	7.4	11.9	10.7	31.9	77.3	16.2	78.2	21.2	15.0	32.6	25.5	41.4	30.5	15.3	12.6
PQ	47.3	95.5	67.9	83.4	17.2	15.5	38.0	22.2	54.7	84.7	21.7	80.4	40.4	37.1	49.8	31.8	54.1	36.4	34.3	32.5

Table 8. Per-class results of our weakly- and fully-supervised models for both semantic and instance segmentation on the Pascal VOC validation set. The IoU measures semantic segmentation performance, whilst the AP^r_{vol} and PQ measure instance segmentation performance.

Metric	Mean	aero-plane	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	motor-bike	per-son	plant	sheep	sofa	train	tv
Weakly supervised model																					
IoU	75.7	85.0	35.9	88.6	70.3	77.9	91.9	83.6	90.5	39.2	84.5	59.4	86.5	82.4	81.5	84.3	57.0	85.9	55.8	85.8	70.4
AP^r_{vol}	55.5	68.8	26.4	74.4	50.4	37.9	70.0	49.4	78.6	22.0	57.1	37.4	78.7	61.6	61.7	50.8	42.2	54.6	46.9	74.9	66.5
PQ	59.5	69.7	18.0	76.8	55.1	48.2	75.4	54.9	77.8	26.4	65.8	43.6	73.8	62.9	68.9	60.8	48.7	62.9	53.7	75.9	71.4
Fully supervised model																					
IoU	79.0	92.0	42.2	90.6	71.1	80.7	95.0	88.5	91.9	41.5	90.6	60.3	86.5	88.3	85.4	86.9	61.7	91.6	53.3	89.2	76.8
AP^r_{vol}	59.5	77.1	31.7	78.1	50.9	40.2	72.4	52.6	82.9	27.0	60.3	35.4	83.1	65.4	72.3	57.3	45.6	56.4	49.7	80.1	71.3
PQ	63.1	77.8	29.1	79.0	57.2	48.9	75.5	59.8	81.7	31.8	67.3	46.2	77.3	69.0	75.3	64.8	52.2	62.0	54.6	79.8	73.7

B Experimental Details

B.1 Network architecture and training

The underlying semantic segmentation network is a reimplementation of PSPNet [59] as described in Sec. 3.5 of the main paper, using a ResNet-101 backbone. This network has an output stride of 8, meaning that the result of the network has to be upsampled by a factor of 8 to obtain the final prediction at the original resolution.

We used most of the same training hyperparameters for training both our fully- and weakly-supervised networks. A batch size of a single 521×521 image crop, momentum of 0.9, and a weight decay of 5×10^{-4} were used in all our experiments.

We trained the semantic segmentation module first, and finetuned the entire instance segmentation network afterwards. For training the semantic segmentation module, the fully supervised models were trained with an initial learning rate of 1×10^{-4}, which was then reduced to 1×10^{-5} when the training loss converged. We used the same learning rate schedule for our weakly-supervised model on Pascal VOC where we did not do any iterative training. In total, about 400k iterations of training were performed. When training our weakly-supervised model iteratively on Cityscapes, we used an initial learning rate of 1×10^{-4} which was then halved for each subsequent stage of iterative training. Each of these iterative training stages were 150k iterations long. Both of the weakly- and fully-supervised models were initialised with ImageNet-pretrained weights and batch normalisation statistics.

In the instance training stage, we fixed the learning rate to 1×10^{-5} for both weakly- and fully-supervised experiments on the VOC and Cityscapes datasets. We observed that a total of 400k iterations were required for the models' training losses to converge.

When training the Faster-RCNN object detector [3], we used all the default training hyperparameters in the publicly available code.

B.2 Multi-label classification network

We obtained weak localisation cues, as described in Sec. 3.3 of the main paper, by first training a network to perform multi-label classification on the Cityscapes dataset.

We adapted the same PSPNet [59] architecture for segmentation for the classification task: The output of the last convolutional layer (`conv5_4`) is followed by a global average pooling layer to aggregate all the spatial information. Thereafter, a fully-connected layer with 19 outputs (the number of classes in the Cityscapes dataset) is appended. This network was then trained with a binary cross entropy loss for each of the 19 labels in the dataset. The loss for a single image is

$$L = \frac{1}{N} \sum_{i=1}^{N} -y_i \log(\mathrm{sigmoid}(z_i)) - (1 - y_i) \log(1 - \mathrm{sigmoid}(z_i)), \qquad (6)$$

where $\mathbf{y}$ is the ground truth image-level label vector and $y_i = 1$ if the ith class is present in the image and 0 otherwise. z_i is the logit for the ith class output by the final fully-connected layer in the network.

It is not possible to fit an entire 2048×1024 Cityscapes image in memory to perform multi-label classification. Using the PSPNet architecture described above (with an output stride of 8), it would take 48.8 GB of memory to train a network with a batch size of 1. Even the standard ResNet-101 architecture [1] (which has a higher output stride of 32, and thus sixteen times less spatial resolution) would take 21.7 GB of memory, which is still almost double the 12GB available in our Titan X GPU. Consequently, we took 15 fixed crops of size 500×400 from the original 2048×1024 image and trained with these crops instead. We were careful not to take random crops during training, as this could be a form of extra supervision. Instead, as we took 15 fixed crops which tile the image and derived image-level labels from them, it effectively means that in a real-world scenario annotators would be asked to annotate image-level labels for fifteen 500×400 images rather than a single 2048×1024 image.

This multi-label classification network was trained with a batch size of 1 and a fixed learning rate of 1×10^{-4} until the training loss converged. We found that this occurred after 50k iterations of training. At this point, the mean Average Precision (mAP) on the validation set was 78.8. The mAP is also used by the Pascal VOC dataset to benchmark multi-label classification [19].

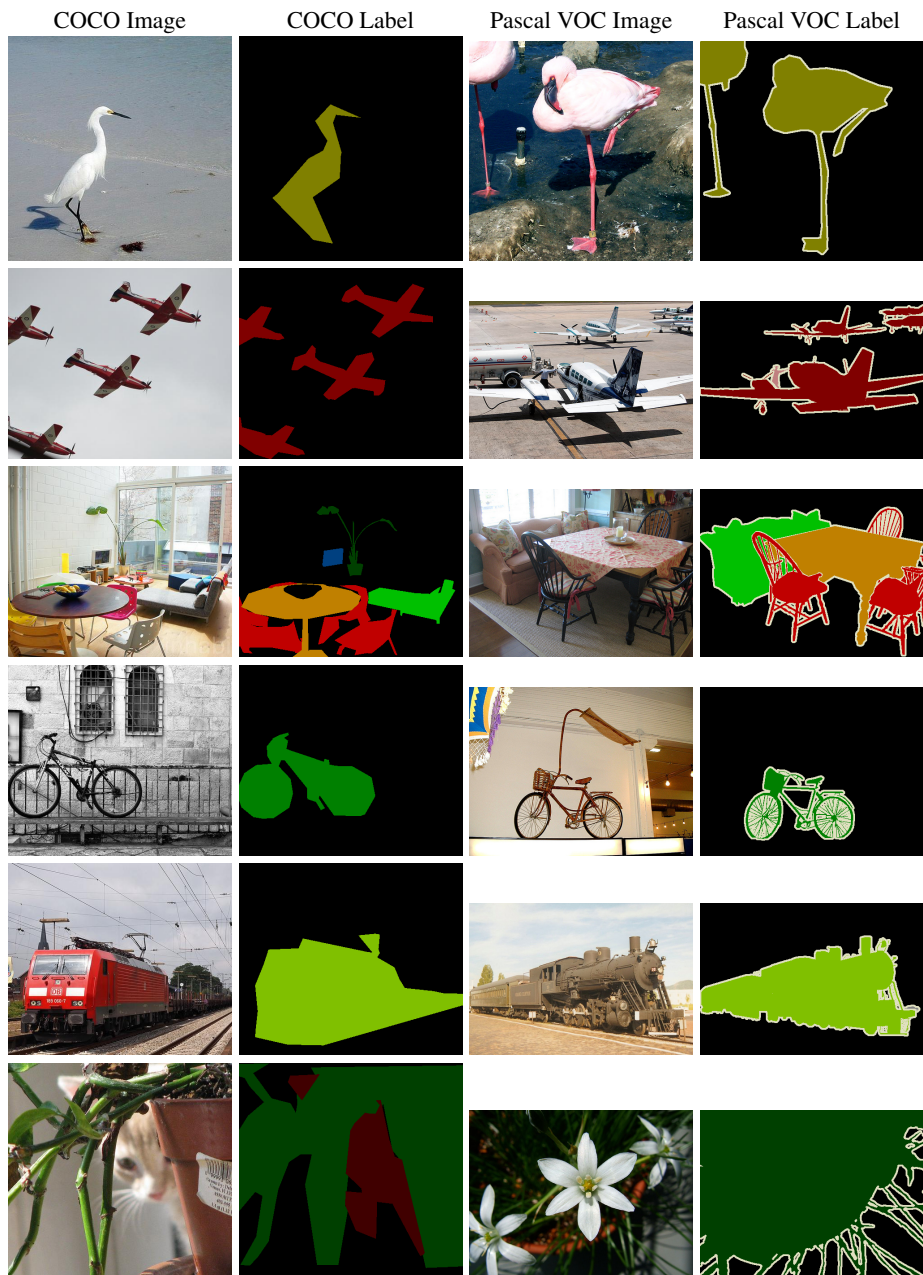## C   Comparison of Pascal VOC and Microsoft COCO annotation quality

Section 4.2 of the main paper mentioned that images in Pascal VOC [19] are annotated at a higher quality than those in Microsoft COCO [20]. Figure 9 illustrates this observation. Images were randomly drawn from Microsoft COCO, and then images from Pascal VOC with the same semantic classes present are shown alongside for comparison. The polygons used to annotate the objects in COCO are evident, and the annotations at the boundaries of objects are often incorrect.

D   Calculation of reduction factor in annotation time if only weak labels are used

The Cityscapes dataset has 11 "stuff" classes, and 8 "thing" classes annotated. Over the training and validation sets, there are an average of 17.9 instances of "thing" classes per full-resolution, 2048×1024 image.
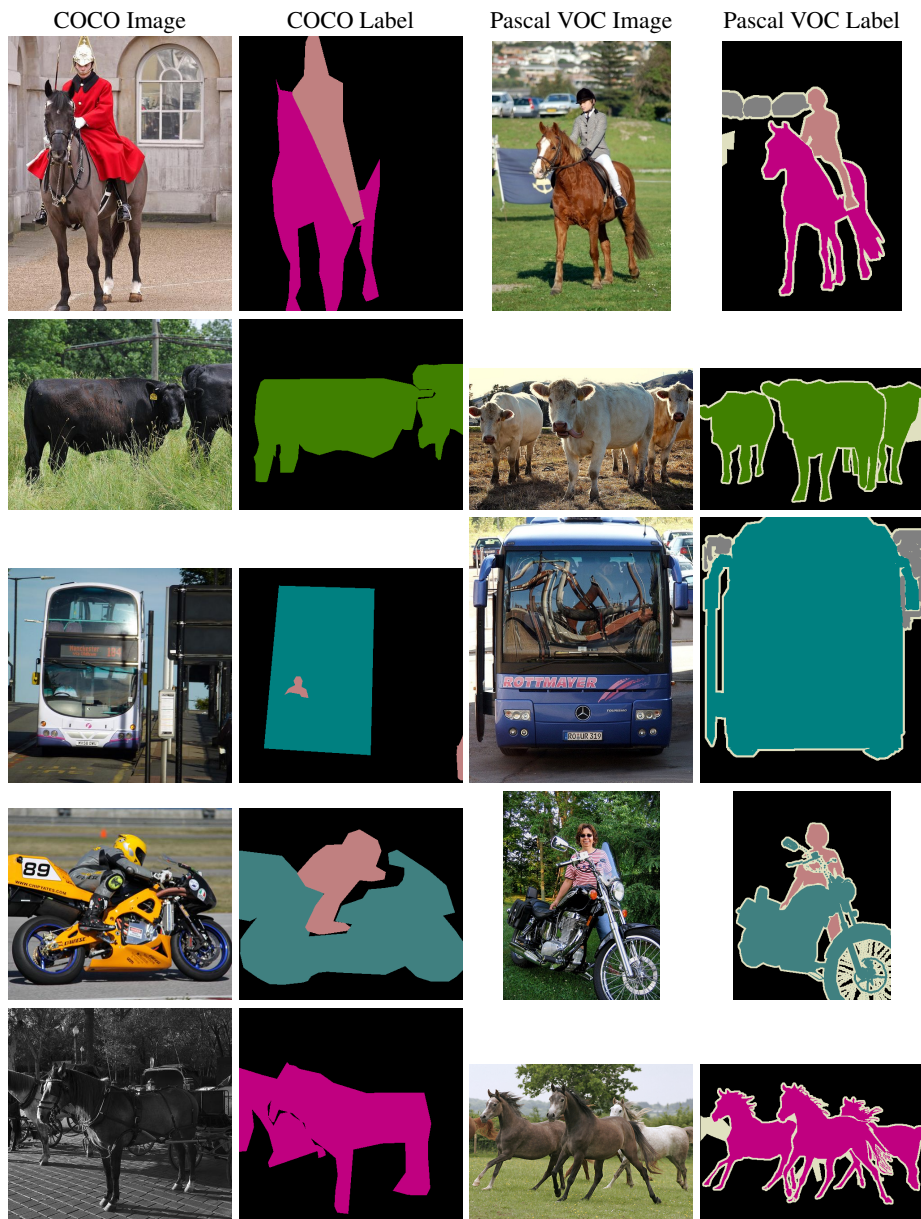
For the calculation in Sec. 1 of the paper, we assumed that each instance of a "thing" class is labelled with a bounding box, and that image-level tags are annotated for all present "stuff" classes. We assumed that a bounding box takes 7 seconds per instance to draw [7] and that an image-level tag takes 1 second to label [8].

Therefore the average time to annotate "thing" classes with a bounding-box is $17.9 \times 7 = 125.3$ seconds. As we took 15 fixed crops per image (as described in Sec. B.2) and there are an average of 3.8 "stuff" tags per crop, the average time to annotate stuff classes is $15 \times 3.8 = 57$ seconds. This totals 182.3 seconds = 3.0 minutes per image. Thus the annotation time is reduced by a factor of 29.6 (since the images originally required 90 minutes to label at a pixel-level by hand [6]) if weak annotations in the form of bounding boxes and image-level tags are used.

COCO Image	COCO Label	Pascal VOC Image	Pascal VOC Label

Fig. 9. Comparison of the annotation quality of images in the Microsoft COCO and Pascal VOC datasets. An image was randomly drawn from COCO, and an image from Pascal VOC with similar content is shown alongside it. The polygons used to annotate the objects in COCO are evident, and the annotations at the boundaries of objects are often incorrect. Grey regions in the Pascal images indicate "void" regions where the annotator was unsure of the correct label.

COCO Image	COCO Label	Pascal VOC Image	Pascal VOC Label

Fig. 9 cont. Comparison of the annotation quality of images in the Microsoft COCO and Pascal VOC datasets. An image was randomly drawn from COCO, and an image from Pascal VOC with similar content is shown alongside it. The polygons used to annotate the objects in COCO are evident, and the annotations at the boundaries of objects are often incorrect. Grey regions in the Pascal images indicate "void" regions where the annotator was unsure of the correct label.