BoxSup: Exploiting Bounding Boxes to Supervise Convolutional Networks for Semantic Segmentation

Jifeng Dai   Kaiming He   Jian Sun

Microsoft Research

{jifdai,kahe,jiansun}@microsoft.com

Abstract

Recent leading approaches to semantic segmentation 以前的语义分割网络都依赖于大量的数据标注，我们提出一种方法实现高准确率但只需要很容易获得的bounding box 标注就可以。 *uman- pixel- rt and limits the performance of deep networks that usually benefit from more training data. In this paper, we propose a method that achieves competitive accuracy but only requires easily obtained bounding box annotations. The basic idea is to iterate between automatically generating region proposals and training convolutional networks. These two steps gradually recover segmentation masks for improving the networks, and vise versa. Our method, called "BoxSup", produces competitive results (e.g., 62.0% mAP for validation) supervised by boxes only, on par with strong baselines (e.g., 63.8% mAP) fully supervised by masks under the same setting. By leveraging a large amount of bounding boxes, BoxSup further unleashes the power of deep convolutional networks and yields state-of-the-art results on PASCAL VOC 2012 and PASCAL-CONTEXT [24].*

1. Introduction

In the past few months, tremendous progress has been made in the field of semantic segmentation [12, 22, 13, 6, 5, 23]. Deep convolutional neural networks (CNNs) [19, 18] that play as rich hierarchical feature extractors are a key to these methods. These networks are trained on large-scale datasets [7, 27] as classifiers, and transferred to the semantic segmentation tasks based on the annotated segmentation masks as supervision.

But pixel-level mask annotations are time-consuming, frustrating, and in the end commercially expensive to obtain. According to the annotation report of the large-scale Microsoft COCO dataset [21], the workload of labeling segmentation masks is more than 15 times heavier than that of spotting object locations. Further, the crowdsourcing annotators need to be specially trained for the tedious and diffi-

cult task of labeling per-pixel masks. These facts limit the amount of available segmentation mask annotations, and thus hinder the performance of CNNs that in general desire large-scale data for training. On the contrary, bounding box annotations are more economical than masks. There have already existed a large number of available box-level annotations in datasets like PASCAL VOC 2007[1] [8] and ImageNet [27]. Though these box-level annotations are less precise than pixel-level masks, their amount may help improve training deep networks for semantic segmentation.

In addition, current leading approaches have not fully utilized the detailed pixel-level annotations. For example, in the Convolutional Feature Masking (CFM) method [6], the fine-resolution masks are used to generate very low-resolution (*e.g.*, 6×6) masks on the feature maps. In the Fully Convolutional Network (FCN) method [22], the network predictions are regressed to the ground-truth masks using a large stride (*e.g.*, 8 pixels). These methods yield competitive results without explicitly harnessing the finer masks. If we consider the box-level annotations as very coarse masks, can we still retain comparably good results without using the segmentation masks?

In this work, we investigate bounding box annotations as an alternative or extra source of supervision to train convolutional networks for semantic segmentation[2]. We resort to unsupervised region proposal methods [31, 2] to generate candidate segmentation masks. The convolutional network is trained under the supervision of these approximate masks. The updated network in turn improves the estimated masks used for training. This process is iterated. Although the masks are coarse at the beginning, they are gradually improved and then provide useful information for network training. Fig. 1 illustrates our training algorithm.

We extensively evaluate our method, called "BoxSup", on the PASCAL segmentation benchmarks [8, 24]. Our

---

[1]The PASCAL VOC 2007 dataset only has bounding box annotations.

[2]The idea of using bounding box annotations for CNN-based semantic segmentation is developed concurrently and independently in [25]. We also compare with the results of [25].
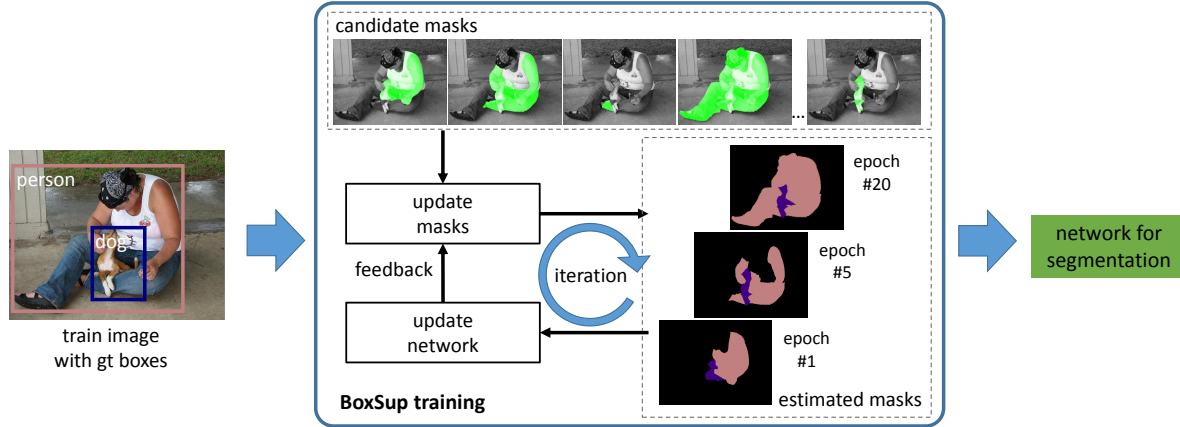
Figure 1: Overview of our training approach supervised by bounding boxes.

bo... ...(i.e., using bounding box annotations) me... ...ed with its mask-supervised (*i.e.*, using mask annotations) counterpart. As such, our method waives the requirement of pixel-level masks for training. Further, our semi-supervised variant in which 9/10 mask annotations are replaced with bounding box annotations yields comparable accuracy with the fully mask-supervised counterpart. This suggests that we may save expensive labeling effort by using bounding box annotations dominantly. Moreover, our method makes it possible to harness the large number of available box annotations to improve the mask-supervised results. Using the limited provided mask annotations and extra large-scale bounding box annotations, our method achieves state-of-the-art results on the PASCAL VOC 2012 and PASCAL-CONTEXT benchmarks.

Why can a large amount of bounding boxes help improve convolutional networks? Our error analysis reveals that a BoxSup model trained with a large set of boxes effectively increases the object *recognition* accuracy (the accuracy in the middle of an object), and its improvement on object boundaries is secondary. Though a box is too coarse to contain detailed segmentation information, it provides an instance for learning to distinguish object categories. The large-scale object instances improve the feature quality of the learned convolutional networks, and thus impact the overall performance for semantic segmentation.

2. Related Work

Deep convolutional networks in general have better accuracy with the growing size of training data, as is evidenced in [18, 34]. The ImageNet classification dataset [27] is one of the largest datasets with quality labels, but the current available datasets for object detection, semantic segmentation, and many other vision tasks mostly have orders

of magnitudes fewer labeled samples. The milestone work of R-CNN [9] proposes to pre-train deep networks as classifiers on the large-scale ImageNet dataset and go on training (fine-tuning) them for other tasks that have limited number of training data. This transfer learning strategy is widely adopted for object detection [9, 14, 30], semantic segmentation [12, 22, 13, 6, 5, 23], visual tracking [32], and other visual recognition tasks. With the continuously improving deep convolutional models [34, 28, 4, 14, 29, 30, 15], the accuracy of these vision tasks also improves thanks to the more powerful generic features learned from large-scale datasets.

Although pre-training partially relieves the problem of limited data, the amount of the task-specific data for fine-tuning still matters. In [1], it has been found that augmenting the object detection training set by combining the VOC 2007 and VOC 2012 sets improves object detection accuracy compared with using VOC 2007 only. In [20], the training set for object detection is augmented by visual tracking results obtained from videos and improves detection accuracy. These experiments demonstrate the importance of dataset sizes for task-specific network training.

For semantic segmentation, there have been existing papers [33, 10] that investigate exploiting bounding box annotations instead of masks. But the box-level annotations have not been used to supervised deep convolutional networks in those works.

3. Baseline

Our BoxSup method is in general applicable for many existing CNN-based mask-supervised semantic segmentation methods, such as FCN [22], improvements on FCN [5, 35], and others [13, 6, 23]. In this paper, we adopt our implementation of the FCN method [22] refined by CRF [5] as the mask-supervised baseline, which we briefly introduce
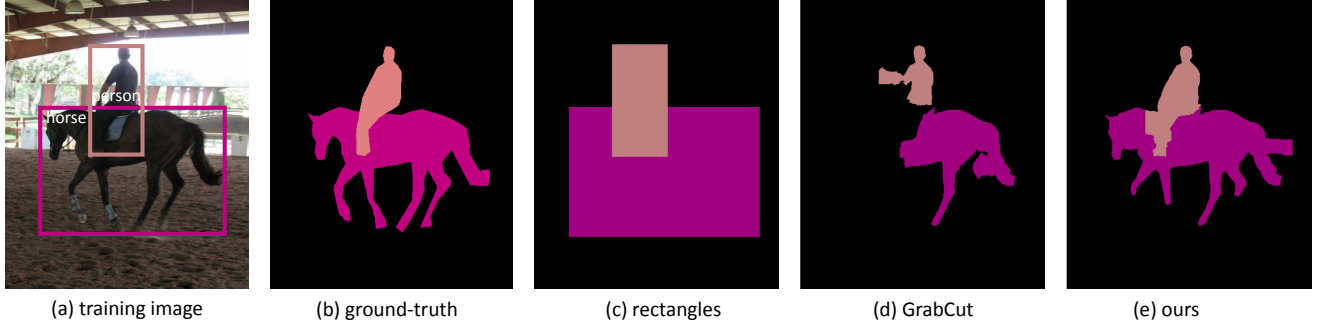
2

| (a) training image | (b) ground-truth | (c) rectangles | (d) GrabCut | (e) ours |

Figure 2: Segmentation masks used as supervision. (a) A training image. (b) Ground-truth. (c) Each box is naïvely considered as a rectangle mask. (d) A segmentation mask is generated by GrabCut [26]. (e) For our method, the supervision is estimated from region proposals (MCG [2]) by considering bounding box annotations and network feedbacks.

as follows.

The network training of FCN [22] is formulated as a per-pixel regression problem to the ground-truth segmentation masks. Formally, the objective function can be written as:

$$\mathcal{E}(\theta) = \sum_{p} e(X_\theta(p), l(p)), \qquad (1)$$

where p is a pixel index, $l(p)$ is the ground-truth semantic label at a pixel, and $X_\theta(p)$ is the per-pixel labeling produced by the fully convolutional network with parameters θ. $e(X_\theta(p), l(p))$ is the per-pixel loss function. The network parameters θ are updated by back-propagation and stochastic gradient descent (SGD). A CRF is used to post-process the FCN results [5].

The objective function in Eqn.(1) demands pixel-level segmentation masks $l(p)$ as supervision. It is not directly applicable if only bounding box annotations are given as supervision. Next we introduce our method for addressing this problem.

4. Approach

4.1. Unsupervised Segmentation for Supervised Training

To harness the bounding boxes annotations, it is desired to estimate segmentation masks from them. This is a widely studied supervised image segmentation problem, and can

1.
2.

insufficient for deep network training.

We propose to generate a set of candidate segments using *unsupervised* region proposal methods (*e.g.*, Selective Search [31]) due to their nice properties. First, region proposal methods have high recall rates [2] of having a good candidate in the proposal pool. Second, region proposal methods generate candidates of greater variance, which provide a kind of data augmentation [18] for network training. We will show by experiments the improvements of these properties.

The candidate segments are used to update the deep convolutional network. The semantic features learned by the network are then used to pick better candidates. This procedure is iterated. We formulate this procedure as an objective function as we will describe below.

It is worth noticing that the region proposal is only used for networking training. For inference, the trained FCN is directly applied on the image and produces pixel-wise predictions. So our usage of region proposals does not impact the test-time efficiency.

4.2. Formulation

Multiscale Combinatorial Grouping (MCG)

As a pre-processing generate segmentation masks. We adopt Multiscale Combinatorial Grouping (MCG) [2] by default, while other methods [31, 17] are also evaluated. The proposal candidate masks are fixed throughout the training procedure. But during training, each candidate mask will be assigned a label which can be a semantic category or background. The labels assigned to the masks will be updated.

With a ground-truth bounding box annotation, we expect it to pick out a candidate mask that overlaps the box as much as possible. Formally, we define an overlapping objective function $\mathcal{E}_o$ as:

$$\mathcal{E}_o = \frac{1}{N} \sum_{S} (1 - \text{IoU}(B, S))\delta(l_B, l_S). \qquad (2)$$

Here S represents a candidate segment mask, and B represents a ground-truth bounding box annotation. $\text{IoU}(B, S) \in [0, 1]$ is the intersection-over-union ratio computed from the ground-truth box B and the tight bounding box of the segment S. The function δ is equal to one if the semantic label l_S assigned to segment S is the same as the ground-truth label l_B of the bounding box B, and zero otherwise. Minimizing $\mathcal{E}_o$ favors higher IoU scores when the semantic labels are consistent. This objective function is normalized by the number of candidate segments N.
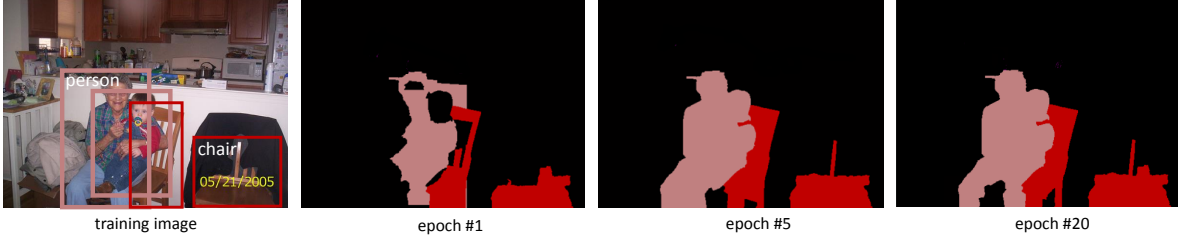
3

Figure 3: Update of segmentation masks during training. Here we show the masks in epoch #1, epoch #5, and epoch #20. Each segmentation mask will be used as the supervision for the next epoch.

With the candidate masks and their estimated semantic labels, we can supervise the deep convolutional network as in Eqn.(1). Formally, we consider the following regression objective function $\mathcal{E}_r$:

$$\mathcal{E}_r = \sum_p e(X_\theta(p), l_S(p)). \qquad (3)$$

Here l_S is the estimated semantic label used as supervision for the network training. This objective function is the same as Eqn.(1) except that its regression target is the estimated candidate segment.

We minimize an objective function that combines the above two terms:

$$\min_{\theta, \{l_S\}} \sum_i (\mathcal{E}_o + \lambda \mathcal{E}_r) \qquad (4)$$

Here the summation $\sum_i$ runs over the training images, and $\lambda = 3$ is a fixed weighting parameter. The variables to be optimized are the network parameters θ and the labeling $\{l_S\}$ of all candidate segments $\{S\}$. If only the term $\mathcal{E}_o$ exists, the optimization problem in Eqn.(4) trivially finds a candidate segment that has the largest IoU score with the box; if only the term $\mathcal{E}_r$ exists, the optimization problem in Eqn.(4) is equivalent to FCN. Our formulation simultaneously considers both cases.

4.3. Training Algorithm

The objective function in Eqn.(4) involves a problem of assigning labels to the candidate segments. Next we propose a greedy iterative solution to find a local optimum.

With the network parameters θ fixed, we update the semantic labeling $\{l_S\}$ for all candidate segments. In our implementation, we only consider the case in which one ground-truth bounding box can "activate" (*i.e.*, assign a non-background label to) one and only one candidate. As such, we can simply update the semantic labeling by selecting a single candidate segment for each ground-truth bounding box, such that its cost $\mathcal{E}_o + \lambda \mathcal{E}_r$ is the smallest among all candidates. The selected segment is assigned the ground-truth semantic label associated with that bounding box. All other pixels are assigned the background label.

The above winner-takes-all selection tends to repeatedly use the same or very similar candidate segments, and the optimization procedure may be trapped in poor local optima. To increase the sample variance for better stochastic training, we further adopt a random sampling method to select the candidate segment for each ground-truth bounding box. Instead of selecting the single segment with the largest cost $\mathcal{E}_o + \lambda \mathcal{E}_r$, we randomly sample a segment from the first k segments with the largest costs. In this paper we use $k = 5$. This random sampling strategy improves the accuracy by about 2% on the validation set.

With the semantic labeling $\{l_S\}$ of all candidate segments fixed, we update the network parameters θ. In this case, the problem becomes the FCN problem [22] as in Eqn.(1). This problem is minimized by SGD.

We iteratively perform the above two steps, fixing one set of variables and solving for the other set. For each iteration, we update the network parameters using one training epoch (*i.e.*, all training images are visited once), and after that we update the segment labeling of all images. Fig.3 shows the gradually updated segmentation masks during training. The network is initialized by the model pre-trained in the ImageNet classification dataset, and our algorithm starts from the step of updating segment labels.

Our method is applicable for the semi-supervised case (the ground-truth annotations are mixtures of segmentation masks and bounding boxes). The labeling $l(p)$ is given by candidate proposals as above if a sample only has ground-truth boxes, and is simply assigned as the true label if a sample has ground-truth masks.

In the SGD training of updating the network, we use a mini-batch size of 20, following [22]. The learning rate is initialized to be 0.001 and divided by 10 after every 15 epochs. The training is terminated after 45 epochs.

5. Experiments

In all our experiments, we use the publicly released VGG-16 model[3] [29] that is pre-trained on ImageNet [27]. The VGG model is also used by all competitors [22, 13, 6,

---

[3] www.robots.ox.ac.uk/~vgg/research/very_deep/

4

data	VOC train			VOC train + COCO	
total #	10,582			133,869	
supervision	mask	box	semi	mask	semi
mask #	10,582	-	1,464	133,869	10,582
box #	-	10,582	9,118	-	123,287
mean IoU	63.8	62.0	63.5	68.1	68.2

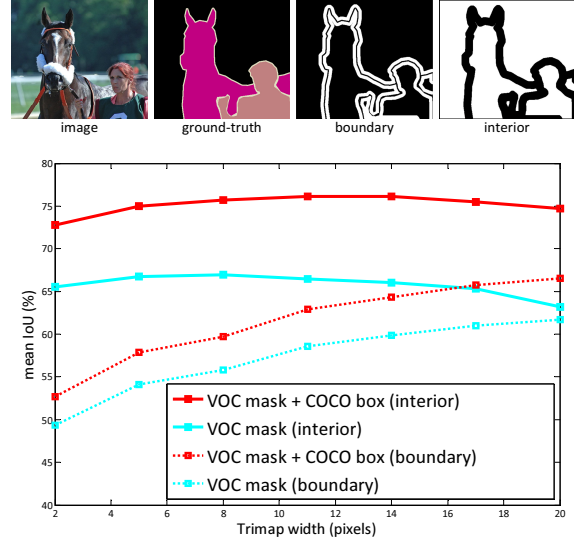Table 1: Comparisons of supervision in PASCAL VOC 2012 validation.

Figure 4: Error analysis on the validation set. Top: (from left to right) image, ground-truth, *boundary* regions marked as white, *interior* regions marked as white). Bottom: *boundary* and *interior* mean IoU, using VOC masks only (blue) and using extra COCO boxes (red).

5, 23] compared in this paper.

5.1. Experiments on PASCAL VOC 2012

We first evaluate our method on the PASCAL VOC 2012 semantic segmentation benchmark [8]. This dataset involves 20 semantic categories of objects. We use the "comp6" evaluation protocol. The accuracy is evaluated by mean IoU scores. The original training data has 1,464 images. Following [11], the training data with ground-truth segmentation masks are augmented to 10,582 images. The validation and test sets have 1,449 and 1,456 images respectively. When evaluating the validation set or the test set, we only use the training set for training. A held-out 100 random validation images are used for cross-validation to set hyper-parameters.

Comparisons of Supervision Strategies

Table 1 compares the results of using different strategies of supervision on the validation set. When all ground-truth masks are used as supervision, the result is our implementation of the baseline DeepLab-CRF [5]. Our reproduction has a score of 63.8 (Table 1, "mask only"), which is very close to 63.74 reported in [5] under the same setting. So we believe that our reproduced baseline is convincing.

When all 10,582 training samples are replaced with bounding box annotations, our method yields a score of 62.0 (Table 1, "box only"). Though the supervision information is substantially weakened, our method shows a graceful degradation (1.8%) compared with the strongly supervised baseline of 63.8. This indicates that in practice we can avoid the expensive mask labeling effort by using only bounding boxes, with small accuracy loss.

Table 1 also shows the semi-supervised result of our method. This result uses the ground-truth masks of the original 1,464 training images and the bounding box annotations of the rest 9k images. The score is 63.5 (Table 1, "semi"), on par with the strongly supervised baseline. Such semi-supervision replaces 9/10 of the segmentation mask annotations with bounding box annotations. This means that we can greatly reduce the labeling effort by dominantly using bounding box annotations.

As a proof of concept, we further evaluate using a sub-

stantially larger set of boxes. We use the Microsoft COCO dataset [21] that has 123,287 images with available ground-truth segmentation masks. This dataset has 80 semantic categories, and we only use the 20 categories that also present in PASCAL VOC. For our mask-supervised baseline, the result is a score of 68.1 (Table 1). Then we replace the ground-truth segmentation masks in COCO with their tight bounding boxes. Our semi-supervised result is 68.2 (Table 1), on par with the strongly supervised baseline. Fig. 5 shows some visual results in the validation set.

The semi-supervised result (68.2) that uses VOC+COCO is considerably better than the strongly supervised result (63.8) that uses VOC only. The 4.4% gain is contributed by the extra large-scale bounding boxes in the 123k COCO images. This comparison suggests a promising strategy - we may make use of the larger amount of existing bounding boxes annotations to improve the overall semantic segmentation results, as further analyzed below.

Error Analysis

Why can a large set of bounding boxes help improve convolutional networks? The error in semantic segmentation can be roughly thought of as two types: (i) *recognition* error that is due to confusions of recognizing object categories, and (ii) *boundary* error that is due to misalignments of pixel-level labels on object boundaries. Although the bounding box annotations have no information about the object boundaries, they provide extra object instances for recognizing them. We may expect that the large amount of

masks	mean IoU
rectangles	52.3
GrabCut	55.2
WSSL [25]	58.5
ours w/o sampling	59.7
ours	<u>62.0</u>

Table 2: Comparisons of estimated masks for supervision in PASCAL VOC 2012 validation. All methods only use 10,582 bounding boxes as annotations, with no ground-truth segmentation mask used.

	SS	GOP	MCG
mean IoU	59.5	60.4	<u>62.0</u>

Table 3: Comparisons of the effects of region proposal methods on our method in PASCAL VOC 2012 validation. All methods only use 10,582 bounding boxes as annotations, with no ground-truth segmentation mask used.

boxes mainly improve the recognition accuracy.

To analyze the error, we separately evaluate the performance on the *boundary* regions and *interior* regions. Following [16, 5], we generate a "trimap" near the ground-truth boundaries (Fig. 4, top). We evaluate mean IoU scores inside/outside the bands, referred to as boundary/interior regions. Fig. 4 (bottom) shows the results of using different band widths for the trimaps.

For the interior region, the accuracy of using the extra COCO boxes (red solid line, Fig. 4) is considerably higher than that of using VOC masks only (blue solid line). On the contrary, the improvement on the boundary regions is relatively smaller (red dash line *vs*. blue dash line). Note that correctly recognizing the interior may also help improve the boundaries (*e.g*., due to the CRF post-processing). So the improvement of the extra boxes on the boundary regions is secondary.

Because the accuracy in the interior region is mainly determined by correctly recognizing objects, this analysis suggests that the large amount of boxes improve the feature quality of a learned BoxSup model for better recognition.

Comparisons of Estimated Masks for Supervision

In Table 2 we evaluate different methods of estimating masks from bounding boxes for supervision. As a naïve baseline, we fill each bounding box with its semantic label, and consider it as a rectangular mask (Fig. 2(c)). Using these rectangular masks as the supervision throughout training, the score is 52.3 on the validation set. We also use GrabCut [26] to generate segmentation masks from boxes (Fig. 2(d)). With the GrabCut masks as the supervision

method	sup.	mask #	box #	mIoU
FCN [22]	mask	V 10k	-	62.2
DeepLabCRF [5]	mask	V 10k	-	66.4
WSSL [25]	box	-	V 10k	60.4
BoxSup	box	-	V 10k	64.6
BoxSup	semi	V 1.4k	V 9k	66.2
WSSL [25]	mask	V+C 133k	-	70.4
BoxSup	semi	V 10k	C 123k	71.0
BoxSup	semi	V 10k	V_{07}+C 133k	73.1
BoxSup+	semi	V 10k	V_{07}+C 133k	**75.2**

Table 4: Results on **PASCAL VOC 2012 test** set. In the supervision ("sup") column, "mask" means all training samples are with segmentation mask annotations, "box" means all training samples are with bounding box annotations, and "semi" means mixtures. "V" denotes the VOC data, "C" denotes the COCO data, and "V_{07}" denotes the VOC 2007 data which only has bounding boxes available.

throughout training, the score is 55.2. In both cases, the masks are not updated by the network feedbacks.

Our method has a score 62.0 (Table 2) using the same set of bounding box annotations. This is a considerable gain over the baseline using fixed GrabCut masks. This indicates the importance of the mask quality for supervision. Fig. 3 shows that our method iteratively updates the masks by the network, which in turn improves the network training.

We also evaluate a variant of our method where each time the updated mask is the candidate with the largest cost, instead of randomly sampled from the first k candidates (see Sec. 4.3). This variant has a lower score of 59.7 (Table 2). The random sampling strategy, which is data augmentation and increases sample variances, is beneficial for training.

Table 2 also shows the result of the concurrent method WSSL [5] under the same evaluation setting. Its results is 58.5. This result suggests that our method estimates more accurate masks than [5] for supervision.

Comparisons of Region Proposals

Our method resorts to unsupervised region proposals for training. In Table 3, we compare the effects of various region proposals on our method: Selective Search (SS) [31], Geodesic Object Proposals (GOP) [17], and MCG [2]. Table 3 shows that MCG [2] has the best accuracy, which is consistent with its segmentation quality evaluated by other metrics in [2]. Note that at test-time our method does not need region proposals. So the better accuracy of using MCG implies that our method effectively makes use of the higher quality segmentation masks to train a better network.

Comparisons on the Test Set

Next we compare with the state-of-the-art methods on

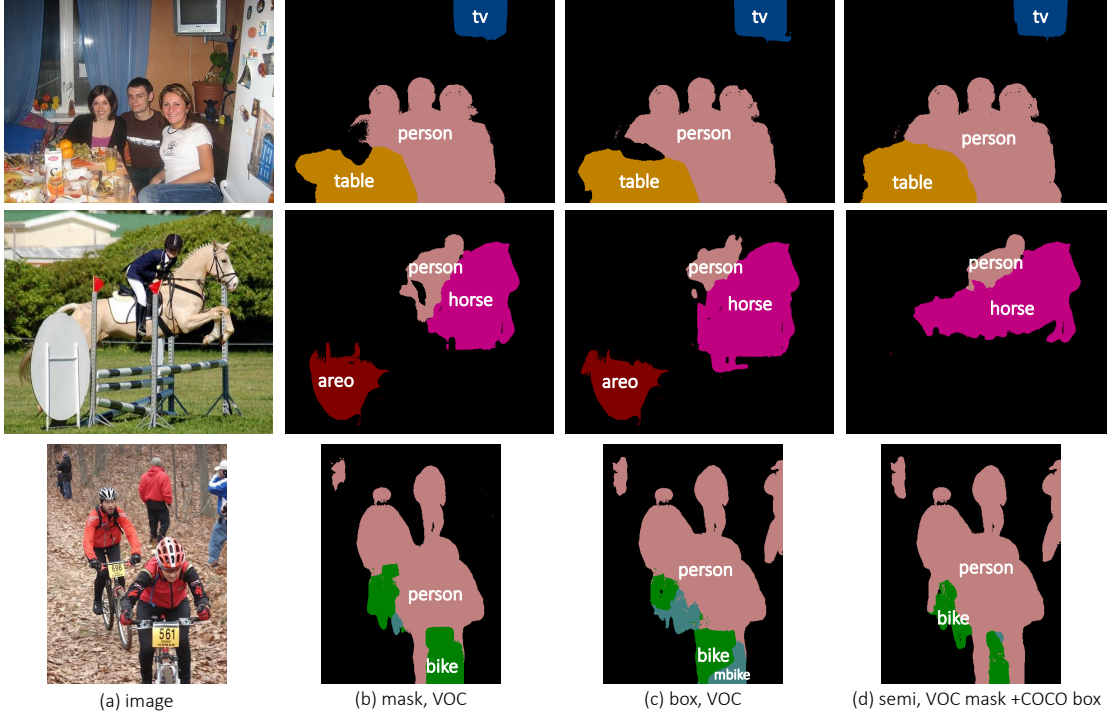| (a) image | (b) mask, VOC | (c) box, VOC | (d) semi, VOC mask +COCO box |

Figure 5: Example semantic segmentation results on **PASCAL VOC 2012** validation using our method. (a) Images. (b) Supervised by masks in VOC. (c) Supervised by boxes in VOC. (d) Supervised by masks in VOC and boxes in COCO.

the PASCAL VOC 2012 *test* set. In Table 4, the methods are based on the same FCN baseline and thus fair comparisons are made to evaluate the impact of mask/box/semi-supervision.

As shown in Table 4, our *box-supervised* result that only uses VOC bounding boxes is 64.6. This compares favorably with the WSSL [25] counterpart (60.4) under the same setting. On the other hand, our *box-supervised* result has a graceful degradation (1.8%) compared with the *mask-supervised* DeepLab-CRF (66.4 [5]) using the VOC training data. Moreover, our semi-supervised variant which replaces 9/10 segmentation mask annotations with bounding boxes has a score of 66.2. This is on par with the mask-supervised counterpart of DeepLab-CRF, but the supervision information used by our method is much weaker.

In the WSSL paper [25], by using all segmentation mask annotations in VOC and COCO, the strongly mask-supervised result is 70.4. Our semi-supervised method shows a higher score of **71.0**. Remarkably, our result uses the bounding box annotations from the 123k COCO images. So our method has a more accurate result but uses much weaker annotations than [25].

On the other hand, compared with the DeepLab-CRF result (66.4), our method has a 4.6% gain enjoyed from exploiting the *bounding box* annotations of the COCO dataset. This comparison demonstrates the power of our method that

exploits large-scale bounding box annotations to improve accuracy.

Exploiting Boxes in PASCAL VOC 2007

To further demonstrate the effect of BoxSup, we exploit the bounding boxes in the PASCAL VOC 2007 dataset [8]. This dataset has no mask annotations. It is a de facto dataset which mask-supervised methods are *not* able to use.

We exploit all 10k images in the VOC 2007 trainval and test sets. We train a BoxSup model using the union set of VOC 2007 boxes, COCO boxes, and the augmented VOC 2012 training set. The score improves from 71.0 to **73.1** (Table 4) because of the extra box training data. It is reasonable for us to expect further improvement if more bounding box annotations are available.

Baseline Improvement

Although our focus is mainly on exploiting boxes as supervision, it is worth noticing that our method may also benefit from other improvements on the mask-sup baseline (FCN in our case). Concurrent with our work, there are a series of improvements [35, 5] made on FCN, which achieve excellent results using strong mask-supervision from VOC and COCO data.

To show the potential of our BoxSup method in parallel with improvements on the baseline, we use a simple test-time augmentation to boost our results. Instead of comput-

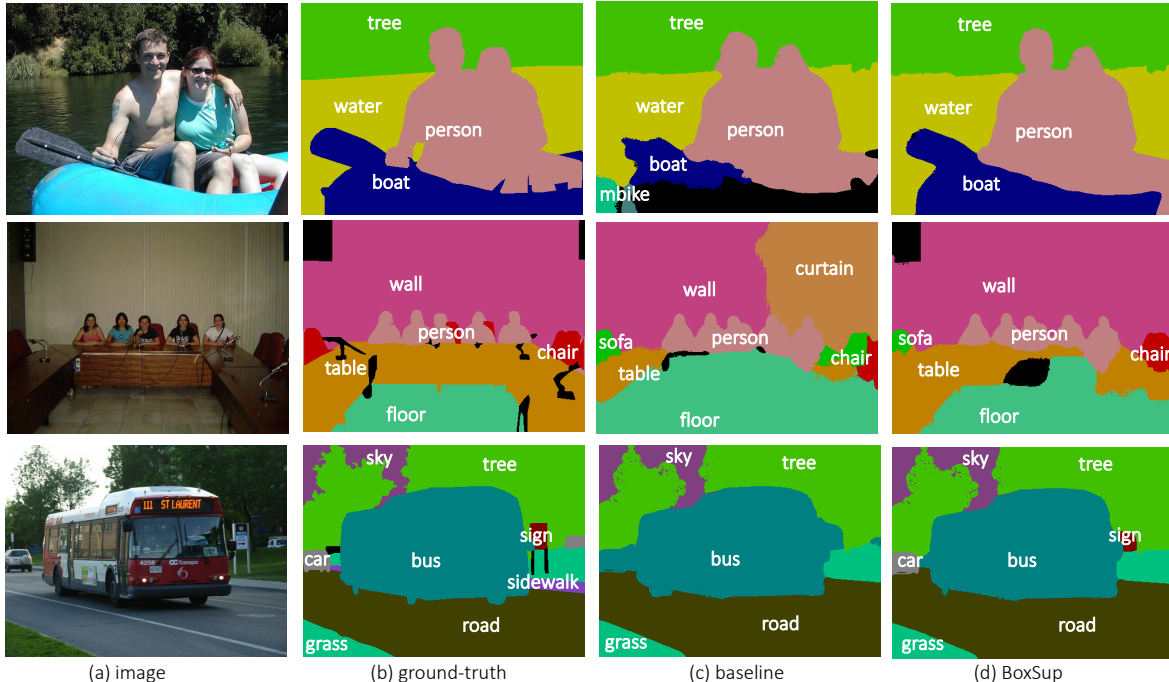|	tree	tree	tree
(a) image	(b) ground-truth	(c) baseline	(d) BoxSup

Figure 6: Example results on **PASCAL-CONTEXT** validation. (a) Images. (b) Results of our baseline (35.7 mean IoU), trained using VOC masks. (c) Results of BoxSup (40.5 mean IoU), trained using VOC masks and COCO boxes.

method	sup.	mask #	box #	mean IoU
O$_2$P [3]	mask	V 5k	-	18.1
CFM [6]	mask	V 5k	-	34.4
FCN [22]	mask	V 5k	-	35.1
baseline	mask	V 5k	-	35.7
BoxSup	semi	V 5k	C 123k	**40.5**

Table 5: Results on **PASCAL-CONTEXT** [24] validation. Our baseline is our implementation of FCN+CRF. "V" denotes the VOC data, and "C" denotes the COCO data.

ing pixel-wise predictions on a single scale, we compute the score maps from two extra scales ($\pm 20\%$ of the original image size) and bilinearly re-scale the score maps to the original size. The scores from three scales are averaged. This simple modification boosts our result from 73.1 to **75.2** (BoxSup+, Table 4) in the VOC 2012 test set. This result is on par with the latest results using strong mask-supervision from both VOC and COCO, but in our case the COCO dataset only provides bounding boxes.

5.2. Experiments on PASCAL-CONTEXT

We further perform experiments on the recently labeled PASCAL-CONTEXT dataset [24]. This dataset provides ground-truth semantic labels for the whole scene, including object and stuff (*e.g.*, grass, sky, water). Following the protocol in [24, 6, 22], the semantic segmentation is performed on the most frequent 59 categories (identified by [24]) plus a background category. The accuracy is measured by mean IoU scores. The training and evaluation are performed on the training and validation sets that have 4,998 and 5,105 images respectively.

To train a BoxSup model for this dataset, we first use the *box annotations* from all 80 object categories in the COCO dataset to train the FCN (using VGG-16). This network ends with an 81-way (with an extra one for background) layer. Then we remove this last layer and add a new 60-way layer for the 59 categories of PASCAL-CONTEXT. We fine-tune this model in the 5k training images of PASCAL-CONTEXT. A CRF for post-processing is also used. We do no use the test-time scale augmentation.

Table 5 shows the results in PASCAL-CONTEXT. The methods of CFM [6] and FCN [22] are both based on the VGG-16 model. Our baseline method, which is our implementation of FCN+CRF, has a score of 35.7 using masks of the 5k training images. Using our BoxSup model pre-trained using the COCO boxes, the result is improved to **40.5**. The 4.8% gain is solely because of the bounding box annotations in COCO that improve our network training. Fig. 6 shows some examples of our results for joint object and stuff segmentation.

8

6. Conclusion

The proposed BoxSup method can effectively harness bounding box annotations to train deep networks for semantic segmentation. Our BoxSup method that uses 133k bounding boxes and 10k masks achieves state-of-the-art results. Our error analysis suggests that semantic segmentation accuracy is hampered by the failure of recognizing objects, which large-scale data may help with.

References

[1] P. Agrawal, R. Girshick, and J. Malik. Analyzing the performance of multilayer neural networks for object recognition. In *ECCV*, 2014.

[2] P. Arbeláez, J. Pont-Tuset, J. T. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. In *CVPR*, 2014.

[3] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu. Semantic segmentation with second-order pooling. In *ECCV*. 2012.

[4] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *BMVC*, 2014.

[5] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*, 2015.

[6] J. Dai, K. He, and J. Sun. Convolutional feature masking for joint object and stuff segmentation. In *CVPR*, 2015.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[8] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) Challenge. *IJCV*, 2010.

[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. 2014.

[10] M. Guillaumin, D. Küttel, and V. Ferrari. Imagenet auto-annotation with segmentation propagation. *IJCV*, 2014.

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

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

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

[14] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*, 2014.

[15] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *arXiv:1502.01852*, 2015.

[16] P. Kohli, P. H. Torr, et al. Robust higher order potentials for enforcing label consistency. *IJCV*, pages 302–324, 2009.

[17] P. Krähenbühl and V. Koltun. Geodesic object proposals. In *ECCV*, 2014.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[19] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1989.

[20] X. Liang, S. Liu, Y. Wei, L. Liu, L. Lin, and S. Yan. Computational baby learning. *arXiv:1411.2861*, 2014.

[21] T.-Y. 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.

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

[23] M. Mostajabi, P. Yadollahpour, and G. Shakhnarovich. Feedforward semantic segmentation with zoom-out features. *arXiv preprint arXiv:1412.0774*, 2014.

[24] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille. The role of context for object detection and semantic segmentation in the wild. In *CVPR*. 2014.

[25] G. Papandreou, L.-C. Chen, K. Murphy, and A. L. Yuille. Weakly- and semi-supervised learning of a dcnn for semantic image segmentation. *arXiv preprint arXiv:1502.02734*, 2015.

[26] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 2004.

[27] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *arXiv:1409.0575*, 2014.

[28] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In *ICLR*, 2014.

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

[30] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

[31] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *IJCV*, 2013.

[32] N. Wang, S. Li, A. Gupta, and D.-Y. Yeung. Transferring rich feature hierarchies for robust visual tracking. *arXiv:1501.04587*, 2015.

[33] W. Xia, C. Domokos, J. Dong, L.-F. Cheong, and S. Yan. Semantic segmentation without annotating segments. In *ICCV*, 2013.

[34] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional neural networks. In *ECCV*, 2014.

[35] 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. *arXiv:1502.03240*, 2015.