# Weakly Supervised Semantic Segmentation: From Box to Tag and Back

Zongliang Ji jerryji007.github.io Olga Veksler https://cs.uwaterloo.ca/~oveksler/ School of Computer Science University of Waterloo Waterloo, Canada

#### Abstract

We propose an approach for semantic segmentation with weak supervision using bounding box annotations. Most previous work relies on segmenting bounding boxes into the object and the background. Each box is segmented independently from the other boxes. We argue that the collection of boxes for the same class naturally provides a dataset from which we can learn a model to segment that object class. Learned model, in turn, leads to a better segmentation of each individual box. Thus for each class, we propose to train a segmentation CNN from the dataset consisting of the bounding boxes for that class. This step transforms the bounding box weak supervision task is on a dataset with a single object class. After we train these single-class CNNs, we apply them back to the training bounding boxes to obtain object/background segmentations and merge them to construct pseudo-ground truth. The obtained pseudo-ground truth is used for training a standard segmentation CNN. We improve the state of the art on Pascal VOC 2012 benchmark in bounding box weak supervision setting. Our code is publicly available at https://github.com/Jerryji007/Box2TagBack-bmvc2021.

## **1** Introduction

There are various types of weak supervision. In image-tag annotation [1, 13, 21, 24, 29], only the classes contained in each image are given. In "scribble" supervision [24, 53, 59, 51, 11], a small set of pixels is annotated in each image. In this paper, we consider supervision with bounding boxes [1, 12, 21, 23, 24, 56, 56], 56], 56], 57], where annotation is in a form of bounding box with the corresponding class label placed around each object from the class of interest. Bounding boxes take approximately only 7 seconds per image to annotate [25], whereas pixel-precise annotations can take more than 4 minutes per image [1, 12].

It may be distributed unchanged freely in print or electronic forms.



Figure 1: Overview of our approach. First we construct single object class datasets by cutting out the bounding boxes from the annotated training images and separating them by class. Then we train CNN in image-tag weakly supervised setting on each dataset. The next step is to apply the trained CNN back to the training bounding boxes. The obtained segmentations are combined with conflict resolution to construct pseudo-ground truth. The final step (not illustrated) is to train a standard segmentation CNN on pseudo-ground truth.

is to use an object/background segmentation algorithm such as GrabCut [3] on each box independently, and then paste the segmented foreground back to form pseudo-ground truth.

Our approach is also based on constructing pseudo-ground truth from bounding boxes. However, we observe that segmenting each bounding box separately is sub-optimal, since each box has a rather limited information about object appearance. Instead we should use data from all boxes of the same class collectively, to construct a better appearance model for that class. The better appearance model, subsequently, can be used to segment each bounding box of that class more accurately, leading, in turn, to a more accurate pseudoground truth. In particular, for each object class, we propose to train a segmentation CNN using the bounding boxes from that class as training data. Note that this step transforms the collection of bounding boxes (for each class separately) into a dataset for weak supervision with image-tag annotations. Each such dataset contains only one object class of interest.

While there are many methods for image-tag weak supervision [1], 13, 21, 24, 29], we chose [11] as particularly suited for our task. Its advantage over previous work is that it does not require Class Activation Maps [14], 19], which have a limited object coverage. In addition, [11] is designed specifically for datasets which have a single object class of interest (as in our setting) and shows state of the art performance in this setting. Furthermore, the loss function in [11] can be redesigned to better suit bounding box datasets.

After training single-class CNNs, we apply them back to each bounding box in the training dataset to obtain pixel precise object masks. Then we develop and evaluate two strategies for constructing pseudo-ground truth from the obtained masks. Finally we train a standard semantic segmentation CNN on our pseudo-ground truth. In essence, we reduce bounding box supervision to image-tag supervision and apply the results back for the original bounding box supervision task. See Fig. 1 for an overview of our approach.

We are the first to explore the information across all bounding boxes of the same class to learn a better object appearance model for that class. The advantage of our approach over prior work is its simplicity. We use standard CNN architecture, our loss function is intuitive and easy to interpret, there is a natural pipeline with no complex stages. Our approach sets a new state-of-the art in weak segmentation with bounding boxes on Pascal VOC [1].

This paper is organized as follows. Sec. 2 reviews prior work, Sec. 3 explains the method [1] for image-tag weak supervision, Sec. 4 presents our approach to semantic segmentation from bounding boxes, and experiments are in Sec. 5.

## 2 Related Work

#### **Bounding Box Weak Supervision**

The bounding box approaches consist of two main stages. The first stage is to construct pseudo-ground truth. The second stage is to train a semantic segmentation CNN, such as  $[\square]$ , using pseudo ground truth instead of ground truth. When generating pseudo ground truth, a pixel that does not belong to any bounding box can be safely marked as the background. There are different approaches for dealing with pixels inside the boxes.

A simple approach is to consider each pixel inside the bounding box as a positive example for the corresponding object class [26]. Conflicts, i.e. pixels that fall inside two bounding boxes, are usually resolved by assuming the smaller box is in front of the larger box. One can also consider only a certain percentage of pixels centrally located in the box as positive examples, and to label the rest as a void class [127]. While simple, these approaches are less accurate, as they are based on a crude approximation of an object with a bounding box.

To get a more accurate estimate of the object in a box, one can apply an object/background segmentation algorithm. DenseCRF [13], GrabCut [3], MCG [3], or their combination have been used in prior work [2, 17, 23, 26]. All of these approaches segment a box into object/background separately from the other boxes, unlike our approach.

Instead of cross entropy, one can design a loss that is better suited for handling noisy pseudo-ground truth [21, 56]. In [56] they first use denseCRF [59], separately in each box, to segment the object from background. From these segmentations, they compute the class specific *filling rate*, which is the object size relative to its box, averaged over the corresponding class. The loss function for training the final CNN uses a fraction of the most confident pixels in the box, and this fraction is equal to the class filling rate. Filling rate computation does use information across different boxes of the same class, but in a rather limited way, i.e. for computing the average area of an object. We use information across boxes in a more substantial way, to model class appearance, and, even with a simpler loss, we outperform [56].

In [22], they generalize the loss from [32] and develop not only class specific, but also image specific filling rate loss and attention mechanism. The first stage is still based on segmentation masks obtained from each box using GrabCut [32], independently from the

other boxes. We could use the loss function from  $[22]^1$  to possibly improve our performance, but again, even with our simpler cross entropy training, we outperform [22].

#### **Regularized Loss for Weak Supervision**

Regularized loss has been used for weak supervision with scribbles [53, 59] and imagetags [51], but has not been used for weak supervision with bounding boxes. Scribbles provide a definitive set of pixels that belong to an object class, and can be used with cross entropy in an objective function. With bounding boxes, there are no pixels that definitely belong to the object class and thus our method is substantially different from [53, 59]. Our method uses a modified version of [50] for image-tag weak supervision. We review [50] in detail in Sec. 3.

## **3** Regularized Loss for Single Object Class

In [1] they develop an approach for training segmentation CNN for a single object class. They design a loss function that models prior knowledge about likely object shape properties and thus can be used to train CNN without pixel precise ground truth. Their loss function is called *regularized* because it is based on certain regularity properties of real world objects.

Let x be the output of segmentation CNN, and  $x_p$  be CNN output for pixel p. The last layer of CNN is softmax, and, therefore,  $x_p \in (0,1)$ . The object is 1, and the background is 0. The most important part of regularized loss is *Sparse-CRF*, defined as

$$L_{crf}(x) = \frac{1}{|\mathcal{P}|} \sum_{(p,q)\in\mathcal{N}} w_{pq} \cdot |x_p - x_q|.$$
(1)

Here  $\mathcal{P}$  is the set of image pixels,  $\mathcal{N}$  is the set of neighboring pixel pairs, and  $w_{pq} = exp(-\frac{||C_p - C_q||^2}{2\sigma^2})$ , where  $C_p$  is the color of pixel p, and  $\sigma$  controls edge scale. There is a loss of  $w_{pq}$  if neighboring pixels are in different classes, assuming that  $x_p, x_q$  are close to 0 or 1. Sparse-CRF is low when segment boundaries are shorter and align with image edges [**1**].

To prevent a trivial solution (everything segmented as object or background), regularized loss must have other parts. The next part is volumetric loss. It assigns a high value to trivial solutions. First let us define the normalized object size  $\bar{x} = \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} x_p$ . Let the batch outputs be  $x_1, \ldots, x_m$ . The volumetric loss is:

$$L_{\nu}(x_{1},...,x_{m}) = \lambda_{s} \sum_{i} \left[ \bar{x}_{i} < s_{min} \right] \cdot \left( \bar{x}_{i} - s_{min} \right)^{2} + \left( \frac{1}{m} \sum_{i} \bar{x}_{i} - 0.5 \right)^{2}$$
(2)

Here  $\lfloor \cdot \rfloor$  is 0 if the argument is false and 1 otherwise,  $s_{min}$  is the minimum object size, and  $\lambda_s$  is the relative weight. The first sum in Eq. (2) is non-zero for any  $x_i$  that has normalized object size less than  $s_{min}$ . Thus it penalizes any segmentation where an object is too small (i.e. less than  $s_{min}$ ). The second sum in Eq. (2) has a low penalty if the object size, averaged over a batch, is not too far from half of the image size. Averaging over a batch allows deviation in size for an individual image without a severe penalty.

The last regularized loss part is the positional loss that gives additional cues for the more likely spatial locations of the object/background. Since there is only one object class of interest, the center pixel is likely to be object, and the border pixels are likely to be background. Let  $\mathcal{B}$  be the border of the image of width w = 3. Let  $\mathcal{C}$  be the pixels in the central box of the image of side size c = 3. The positional loss  $L_p(x)$  is

$$L_p(x) = \left(\frac{1}{|\mathcal{B}|} \sum_{p \in \mathcal{B}} x_p\right)^2 + \left(\frac{1}{|\mathcal{C}|} \sum_{p \in \mathcal{C}} x_p - 1\right)^2.$$
(3)

<sup>&</sup>lt;sup>1</sup>They plan to release the code but it is not publicly available yet.

The complete regularized loss is a weighted combination of the losses above

$$L_{reg}(x) = \lambda_{crf} L_{crf}(x) + \lambda_{\nu} L_{\nu}(x) + \lambda_{p} L_{p}(x).$$
(4)

If negative images (i.e. images containing only background pixels) are available, then they can also be included in the training with *negative* loss:

$$L_{neg}(x) = \lambda_{neg} \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} x_p^2.$$
(5)

The parameter settings are:  $s_{min} = 0.15$ ,  $\lambda_{crf} = 100$ ,  $\lambda_v = \lambda_p = 1$ ,  $\lambda_s = 5$ ,  $\lambda_{neg} = 1$ ,  $\sigma = 0.15$ .

# 4 Our Approach

In this section we describe our approach. It has the following pipeline. First we take the training dataset with bounding box annotations, crop out the bounding boxes, separate them by class, thus constructing a single class dataset for each object class. Next we train a single class CNN for each class in image-tag supervision setting, thus creating a corresponding object segmentation CNN. Next we construct pseudo-ground truth by applying trained single class CNNs to the bounding boxes and combining the results with conflict resolution (three different conflict resolution approaches are evaluated). Finally, we train a standard multiclass CNN on pseudo-ground truth to obtain a semantic segmentation CNN.

We describe single object class dataset construction in Sec. 4.1, our approach for weakly supervised single class CNN training in Sec. 4.2, and the pseudo-ground truth construction in Sec. 4.3. The final step of training a standard segmentation CNN with our pseudo-ground is described in Sec. 5.

#### 4.1 Single Object Dataset Construction

For each semantic class, we crop out its boxes and put them into a separate single object class dataset. When cropping a box, we take a border of 3 pixels in width from the surrounding image outside the box. The border is used for the positional loss, which encourages border pixels to be the background. If pixels in the border region contain boxes of another object, we mark them as void. The void class is ignored during loss calculation. Border pixels provide examples of the background class and make it easier to learn to segment the object from the background. Strictly speaking, this makes training on our single class dataset a mixture of image-tag and scribble supervision, where scribbles are provided for the background class only, and only at the border of each sample. Still, we refer to this setting as image-tag weak supervision, as the main source of supervision is through image-tags<sup>2</sup>.

Suppose we are constructing a single object class dataset for, say, class *cat*. We could take all the boxes for the class *cat*. Some of the boxes for class *cat* intersect boxes of objects of other classes. In case of intersection with another class, the box of class *cat* may have pixels of that other object class. Learning appearance of the *cat* class is made harder by presence of another class, since we are learning only with image-tag annotations. Therefore, we only take the boxes for the class that do not intersect with boxes from the other classes<sup>3</sup>.

 $<sup>^{2}</sup>$ In fact we could still learn without the border pixel labels, similar to how it is done in [ $\square$ ]. But since the box annotation data does provide labels for the pixels outside any box, it is advantageous to make use of it.

<sup>&</sup>lt;sup>3</sup>Note that in Pascal VOC [ $\square$ ] dataset which we use for evaluation, there are objects from the classes of interest without a bounding box annotations around them. Thus any box which does not have an intersection with a box from another class in the annotation, still may have pixels from the other classes.

Intersection with its own class does not present difficulties, since no mixing between different classes occurs in such case. Thus, for example, if a box of class *cat* intersects with another box of class *cat*, we still include it in the single class dataset for object *cat*.

Lastly, we take boxes in the single class dataset only if their longer side is bigger than 50 pixels. Otherwise a box is too small in resolution and is less helpful for learning appearance. We rescale all cropped boxes to be of size 256 by 256 for training CNN.

#### 4.2 Single Object Class Training

We train a single class CNN with a regularized loss function, which we redesign from [1] to better suit a single object class dataset derived from bounding box annotations.

In [1], they observe that in a single class dataset, the central pixel is likely to be the object. Since they deal with general images, the object size can be rather small. Therefore, they use a rather modest prior, only the small central patch is encouraged to be the object. In contrast, for bounding boxes, we observe that the bounding box is likely to cover the object rather tightly, and can use a stricter, and, therefore, more useful prior on the object position.

We replace the central pixel prior by a tight box prior, inspired by  $[\Box ]$ . Our tight box prior encourages either every row or every column in the box to contain at east one object pixel. This means that the object fills out the full width or the full height of the box. We do not expect the object to fill out both width and height, since often bounding boxes are placed around the object more loosely than that, for example, see the box for the plant in Fig. 1, bottom left. However, usually a box is tight in at least horizontal or vertical dimension.

An additional modification is also to the positional prior. In [11], the border of the image is not guaranteed to be the background since they consider general images. Therefore, they used mean squared loss, which penalizes mistakes less. Since we cropped the border of each sample from the outside of a bounding box, we know the border region belongs to the background, so we use cross entropy instead of a mean squared loss.

For a training image, let  $\mathcal{B}$  be the border region (cropped from the outside of the corresponding bounding box), and  $\mathcal{R}$  be the inner region (corresponding to the actual bounding box). Our positional prior, replacing the one in Eq. (3) is

$$L_{p}(x) = -\frac{1}{|\mathcal{B}|} \sum_{p \in \mathcal{B}} \log(1 - x_{p}) + (\max\{\frac{1}{r} \sum_{row \in \mathcal{R}} \max_{x_{p} \in row} x_{p}, \frac{1}{c} \sum_{col \in \mathcal{R}} \max_{x_{p} \in col} x_{p}\} - 1)^{2}, \quad (6)$$

where r and c are the row and column lengths.

The last change to the regularized loss is to raise the minimum size requirement for the object to  $s_{min} = 0.3$  in Eq. (2), since the size of an object relative to its bounding box is large.

Finally, we add the negative loss in Eq. (5) to the regularized loss in Eq. (4). Although we collect non-overlapping boxes in our single class datasets, still, for a box of some object class, there can be pixels from other object classes due to the limited precision of box annotations. With negative loss, when we train for a class, the datasets for all the other classes are used in Eq. (5), thus reducing sensitivity to objects of other classes.

Except for  $s_{min}$ , other parameter settings are the same as in Sec. 3 for all single class datasets. We train CNN for each object class separately. CNN architecture is the same as in  $[\fbox]^4$ . They use Unet  $[\boxdot]$  with ResNeXt  $[\boxdot]$  fixed features in the encoder. The features are pretrained on Imagenet  $[\fbox]$ . After single CNN classifiers are trained for each

<sup>&</sup>lt;sup>4</sup>https://github.com/morduspordus/SingleClassRL

class	#boxes	MCG	denseCRF	grabcut	salient	ours	ours(T)	ours(T+N)
aero	608	63.72	70.97	67.12	88.43	87.13	88.88	89.12
bike	243	58.41	69.56	69.60	80.00	84.37	85.59	85.74
bird	827	67.32	76.54	70.69	90.07	91.16	91.57	91.70
boat	444	65.73	76.68	76.58	77.57	84.91	87.26	87.14
bottle	376	78.61	85.19	84.97	82.50	89.28	92.45	92.68
bus	310	76.75	89.32	87.04	92.71	91.44	94.73	94.60
car	909	72.07	86.27	81.99	88.00	89.23	92.54	92.90
cat	901	73.86	86.49	80.44	84.62	92.69	94.71	94.73
chair	1003	58.73	69.41	63.48	68.28	74.71	75.99	78.64
cow	428	72.54	84.99	75.01	87.17	91.42	92.56	92.53
table	65	62.45	83.63	82.22	63.32	75.26	79.59	79.69
dog	953	72.58	85.95	79.40	89.25	93.98	94.36	94.42
horse	359	66.57	79.22	71.00	88.22	90.26	91.35	91.53
mbike	248	59.30	80.60	75.92	85.48	87.95	89.61	89.61
person	4029	70.02	80.01	78.29	84.60	85.83	88.02	89.05
plant	437	63.13	81.24	76.10	74.62	82.59	87.51	87.69
sheep	548	73.95	84.66	76.78	85.22	88.67	91.42	91.68
sofa	256	70.07	78.13	73.04	66.04	78.79	81.59	81.99
train	445	68.64	83.94	78.91	85.89	87.86	92.05	92.08
tv	455	82.09	86.96	85.00	83.52	88.18	92.20	92.36
mean $F_{\beta}$		68.83	81.00	76.68	82.28	86.79	89.20	89.50

Z. JI, O. VEKSLER: FROM BOX TO TAG AND BACK

Table 1: Comparison of segmentation accuracy on training boxes from Pascal VOC 2012 using MCG [26], denseCRF [19] GrabCut [63], salient object detection [63] and three versions of our method, see text for explanation. Performance metric is  $F_{\beta}$  score (higher is better).

object class, they are applied back to the bounding boxes and a segmentation into object/background is obtained, which is used for constructing pseudo-ground truth (Sec. 4.3).

We now compare the accuracy of our box segmentations with those obtained by MCG [[]], denseCRF [[]], and GrabCut [[]] on Pascal VOC 2012 [[]] dataset, as all of these have been used for pseudo-ground truth construction by previous work. Please see supplementary materials for implementation details of MCG, denseCRF, and GrabCut. In addition, we also evaluate salient object detection for segmenting a box. We use BasNet [[]]<sup>5</sup>. Note that BasNet was trained on pixel precise ground truth from salient object benchmarks. We use  $F_{\beta}$  performance metric, defined as  $F_{\beta} = \frac{(1+\beta^2)precision \times recall}{\beta^2 \times precision + recall}$ , with  $\beta^2 = 0.3$ .

The comparison is in Table 1. There are three versions of our method: **ours** uses the original positional loss in Eq. (3) from [ $\square$ ], **ours**(T) uses our new positional loss based on the tight box assumption (Eq. (6)), and **ours**(T+N) uses our new positional loss and the negative loss (Eq. (5)). Our new tight box positional loss significantly outperforms the positional loss from [ $\square$ ]. Performance is improved for all classes. The largest improvement, by over 4 points, is for the table and plant classes. The appearance of these classes is harder to model in weakly supervised setting, so having a stronger prior helps them the most. Adding negative loss improves the performance for almost all classes, but only slightly.

All three versions of our method outperform MCG, denseCRF, GrabCut and salient object detection by a large margin in terms of the mean  $F_{\beta}$  score. Saliency, although not previously used for bounding box segmentation, performs best out of methods other than ours. Some classes are naturally more salient than others, for example, the bird class. Here saliency achieves almost the same performance as we do. However, some classes, such as

<sup>&</sup>lt;sup>5</sup>https://github.com/xuebinqin/BASNet



Figure 2: Illustrates pseudo-ground truth construction, see text for a description.

sofa and table, are far from salient, and the difference in  $F_{\beta}$  score is almost 15.

The total training time for our method for all classes is 33.7 hours, which is less than training the final CNN on pseudo-ground truth, see Sec. 5. This is because single class CNNs have a lighter architecture compared to the final CNN (deepLab [2]). Also note that with our approach if one wishes to add new classes, previous classes do not have to be retrained.

## 4.3 Pseudo Ground Truth Construction

We now describe how we construct pseudo-ground truth after training a single-class CNN for each class. First, to each bounding box in the training data, we apply the single-class trained CNN corresponding to the class of the box and obtain the segmentation of that box into the object and the background. Note that at this stage, we apply trained CNN to all bounding boxes in the training data, whether they overlap the other boxes or not, in order to obtain a denser pseudo ground truth. Each box is scaled to  $256 \times 256$ .

Next, we combine box segmentations for pseudo-ground truth with conflict resolution. Pixels that do not belong to any box are marked as background. A pixel that gets segmented as an object in one box, or in several boxes of the same class, is assigned the box class.

Next we need to resolve conflicts for pixels that get labeled as an object in two or more boxes (i.e. *overlap* pixels) that belong to different object classes. We test three strategies, see Fig. 2(c). The first strategy is to assign overlap pixels to the class of the smaller object in the overlapping boxes [II], we call this approach *overlap-small*, Fig. 2(c), left. This heuristic assumes that an object in front of another one tends to be smaller. It does result in errors when the object in front is larger. To reduce errors, at the cost of a less dense pseudo-ground truth, our second approach is to label the overlap pixels as void, to be ignored during training. We call this approach *overlap-void*, Fig. 2(c), middle. Void pixels are in white.

Our last approach is to label as void not only the overlap pixels, but also pixels that are labeled as background in any box, Fig. 2(c), right. We call this approach *overlap-back-void*. The motivation is as follows. Some pixels inside boxes are assigned to background erroneously, i.e. they are false negative. Labeling the background inside a box as void removes all false negative errors, at the cost of having less background samples. However, we already do have many background samples from pixels that do not fall into any box.

In Table 2 we show the accuracy, in terms of miou metric of our pseudo-ground truth for all three cases, as well as the percentage of new void pixels we introduce during pseudo-ground truth construction. Thus we do not include the void pixels already present in Pascal VOC dataset in the count. As expected, *overlap-small* is the least accurate but has no void pixels, *overlap-back-void* is highly accurate at the cost of having almost a quarter of void pixels, and *overlap-void* has middle accuracy with only a small number of void pixels.

#### Z. JI, O. VEKSLER: FROM BOX TO TAG AND BACK

method	mIoU-Train	% void
Ours (overlap-small)	77.3	0
Ours (overlap-back-void)	90.2	23.5
Ours (overlap-void)	79.9	3.1

Table 2: Accuracy of pseudo ground truth for our method, in terms of miou metric, and the percentage of void pixels. These results are computed on Pascal VOC 2012, training set.

method	backbone	mIoU-Val	mIoU-Test	mIoU-Val FullS
WSSL (CRF)[26]	VGG-16	60.6	62.2	67.6
BoxSup (CRF) []	VGG-16	62.0	64.6	63.8
SDI (CRF) [	Resnet-101	65.7	-	69.1
BCM (CRF) [	Resnet-101	70.2	-	74.5
GCMCG [🛂]	Resnet-101	74.3	75.5	77.3
Box2Seg [21]	Resnet-101	74.9	-	-
Box2Seg (CRF)[	Resnet-101	76.4	-	-
Ours (overlap-small)	Resnet-101	76.1	76.0	77.8
Ours (overlap-back-void)	Resnet-101	75.7	75.5	77.8
Ours (overlap-void)	Resnet-101	77.1	76.1	77.8

Table 3: Comparison of our approach to previous bounding box supervised methods on PAS-CAL VOC 2012 validation and test sets. Here (CRF) means the method uses denseCRF [1] post-processing. Performance metric is *mIoU*. The last column is performance of the corresponding method using full supervision on Pascal VOC 2012 validation set.

# **5** Experimental Results

We evaluate our approach on Pascal VOC 2012 dataset [ $\Box$ ] with augmented annotations from [ $\Box$ ] for a total of 10,582 training images. The number of validation images is 1,449. For pseudo-ground truth training, we use DeepLab-ResNet101 [ $\Box$ ] pretrained on ImageNet [ $\Box$ ]. We train on 513x513 images and use random horizontal flip, rescale, and Gaussian blur for data augmentation. We use stochastic gradient descent with the same parameters as in [ $\Box$ ], except our initial learning rate is 1e - 4. Training time is 46.6 hours. Our implementation is in pytorch [ $\Box$ ] and we use NVIDIA GeForce RTX 2080 Ti graphics card.

When we train on pseudo-ground truth which is overlap-small or overlap-void, we train



Figure 3: Example results (for *overlap-void*). In each image pair, the left is the ground truth, the right is our result.

with standard cross entropy for 200 epochs. When we train with *overlap-back-void*, because the number of void pixels is large, we get slightly better results<sup>6</sup> if we add denseCRF loss from [53]<sup>7</sup> to the cross-entropy loss. We use the same setting of denseCRF and its relative weight as in [53]. When training with denseCRF loss, we use the same strategy as in [53]: first we train only with cross-entropy (for 100 epochs), and then with both cross entropy and denseCRF (for another 100 epochs). For comparison, we also train DeepLab-ResNet101 with pixel precise ground truth, for 200 epochs.

We compare the accuracy of our algorithm to the recent bounding box weakly supervised methods in Table 3. The evaluation metric is mIoU. Methods using Resnet-101 backbone perform better. Most methods use denseCRF [I] as post-processing (marked with CRF in parenthesis in the table), and do not report results without post-processing. Typically performance with denseCRF post-processing improves the results by about 2 points of mIoU.

We do not use denseCRF post-processing. Previously best method on validation set is Box2Seg [21], their mIoU = 74.9 without post-processing. Note that [21] uses UPerNet [22], which significantly outperforms DeepLab. All our results are better than Box2Seg [21] without post-processing. Our result with *overlap-void* is slightly better than Box2Seg [21] with post-processing. We are using a standard cross-entropy loss. They use a loss function that better handles noisy pseudo-ground truth. It is likely that if our masks learned with image-tag weak supervision are used in conjunction with the loss in [21]<sup>8</sup>, the performance would improve. Our results are also the best of previously reported on the Pascal test set.

The last column in Table 3 is the performance of the same CNN architecture for each method but trained with pixel precise ground truth. Our method is only 0.7 points of mIoU measure behind training with full supervision.

We now compare the performance on the final task, semantic segmentation, if we use the loss function in [11] instead of our loss. In particular, we test the performance we get when constructing pseudo ground truth from boxes in column 7 of Table 1 instead of column 9. We use the *overlap-void* method for constructing the ground truth and train deepLab with the same parameters as before. The resulting *mIoU* is only 71.2, significantly worse than 77.1 using our modified regularized loss. Such a large gap in performance is due to the accuracy of the pseudo-ground truth. We computed *mIoU* of pseudo ground truth using the results of the loss in [11], and it is only 73.4, vs. *mIoU* = 79.9 using our loss. The percentage of void pixels is around 3 in both cases. Some examples of our results are in Fig. 3.

## Conclusions

We proposed an approach for semantic segmentation with bounding box supervision that takes advantage of the collective information in the boxes of the same class to learn the appearance of the corresponding class. This, in turn, is used to segment each bounding box more accurately. All prior bounding box supervision methods make use of some segmentation method in each bounding box, independently of other boxes. Thus all prior work could benefit from our approach for more accurate box segmentation. The advantage of our method is that it is simple and based on an intuitive loss function. A disadvantage is training a separate CNN for each object class. However, when adding a new class, the previous classes do not need to be retrained. One promising direction for a further improvement is to add more image-tag data for supervision, for example, by web image search on class names. Another direction is to incorporate more complex losses for training, such as those in [20].

<sup>&</sup>lt;sup>6</sup>See supplementary materials.

<sup>&</sup>lt;sup>7</sup>https://github.com/meng-tang/rloss

<sup>&</sup>lt;sup>8</sup>Their code is not available online yet.

## References

- Jiwoon Ahn and Suha Kwak. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In *Conference on Computer Vision and Pattern Recognition*, pages 4981–4990, 2018.
- [2] Nikita Araslanov and Stefan Roth. Single-stage semantic segmentation from image labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4253–4262, 2020.
- [3] Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. What's the point: Semantic segmentation with point supervision. In *European conference on computer vision*, pages 549–565. Springer, 2016.
- [4] Yuri Boykov and Vladimir Kolmogorov. Computing geodesics and minimal surfaces via graph cuts. In 9th IEEE International Conference on Computer Vision (ICCV 2003), 14-17 October 2003, Nice, France, pages 26–33, 2003.
- [5] Yu-Ting Chang, Qiaosong Wang, Wei-Chih Hung, Robinson Piramuthu, Yi-Hsuan Tsai, and Ming-Hsuan Yang. Weakly-supervised semantic segmentation via subcategory exploration. In *Proceedings of the IEEE Conference on Computer Vision* and Pattern Recognition, pages 8991–9000, 2020.
- [6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *International Conference on Learning Research*, 2015.
- [7] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2018.
- [8] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *European Conference on Computer Vision*, pages 833–851, 2018.
- [9] Jifeng Dai, Kaiming He, and Jian Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *International Conference on Computer Vision*, pages 1635–1643, 2015.
- [10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In CVPR09, 2009.
- [11] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2): 303–338, June 2010.
- [12] Junsong Fan, Zhaoxiang Zhang, Chunfeng Song, and Tieniu Tan. Learning integral objects with intra-class discriminator for weakly-supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4283–4292, 2020.

- [13] Ruochen Fan, Qibin Hou, Ming-Ming Cheng, Gang Yu, Ralph R. Martin, and Shi-Min Hu. Associating inter-image salient instances for weakly supervised semantic segmentation. In *European Conference on Computer Vision*, September 2018.
- [14] Clement Farabet, Camille Couprie, Laurent Najman, and Yann LeCun. Learning hierarchical features for scene labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013. in press.
- [15] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *International Conference on Computer Vision*, pages 991–998. IEEE, 2011.
- [16] Zilong Huang, Xinggang Wang, Jiasi Wang, Wenyu Liu, and Jingdong Wang. Weaklysupervised semantic segmentation network with deep seeded region growing. In *Conference on Computer Vision and Pattern Recognition*, pages 7014–7023, 2018.
- [17] Anna Khoreva, Rodrigo Benenson, Jan Hendrik Hosang, Matthias Hein, and Bernt Schiele. Simple does it: Weakly supervised instance and semantic segmentation. In *Conference on Computer Vision and Pattern Recognition*, pages 1665–1674, 2017.
- [18] Alexander Kolesnikov and Christoph H. Lampert. Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In *European Conference on Computer Vision*, pages 695–711, 2016.
- [19] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Neural Information Processing Systems*, pages 109–117, 2011.
- [20] Viveka Kulharia, Siddhartha Chandra, Amit Agrawal, Philip H.S. Torr, and Ambrish Tyagi. Box2seg: Attention weighted loss and discriminative feature learning for weakly supervised segmentation. *European Conference on Computer Vision*, 2020.
- [21] Jungbeom Lee, Eunji Kim, Sungmin Lee, Jangho Lee, and Sungroh Yoon. Ficklenet: Weakly and semi-supervised semantic image segmentation using stochastic inference. In *Conference on Computer Vision and Pattern Recognition*, pages 5267–5276, 2019.
- [22] Victor Lempitsky, Pushmeet Kohli, Carsten Rother, and Toby Sharp. Image segmentation with a bounding box prior. In 2009 IEEE 12th international conference on computer vision, pages 277–284. IEEE, 2009.
- [23] Qizhu Li, Anurag Arnab, and Philip HS Torr. Weakly-and semi-supervised panoptic segmentation. In *Proceedings of the European Conference on Computer Vision*, pages 102–118, 2018.
- [24] Di Lin, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun. Scribblesup: Scribblesupervised convolutional networks for semantic segmentation. In *Conference on Computer Vision and Pattern Recognition*, pages 3159–3167, 2016.
- [25] Dim P Papadopoulos, Jasper RR Uijlings, Frank Keller, and Vittorio Ferrari. Extreme clicking for efficient object annotation. In *International Conference on Computer Vision*, pages 4930–4939, 2017.

- [26] George Papandreou, Liang Chieh Chen, Kevin P. Murphy, and Alan L. Yuille. Weakly and semi supervised learning of a deep convolutional network for semantic image segmentation. In *International Conference on Computer Vision*, pages 1742–1750, 2015.
- [27] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.
- [28] Deepak Pathak, Philipp Krahenbuhl, and Trevor Darrell. Constrained convolutional neural networks for weakly supervised segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1796–1804, 2015.
- [29] Pedro H. O. Pinheiro and Ronan Collobert. From image-level to pixel-level labeling with convolutional networks. In *Conference on Computer Vision and Pattern Recognition*, pages 1713–1721, 2015.
- [30] Jordi Pont-Tuset, Pablo Arbelaez, Jonathan T Barron, Ferran Marques, and Jitendra Malik. Multiscale combinatorial grouping for image segmentation and object proposal generation. *IEEE transactions on pattern analysis and machine intelligence*, 39(1): 128–140, 2016.
- [31] Xuebin Qin, Zichen Zhang, Chenyang Huang, Chao Gao, Masood Dehghan, and Martin Jagersand. Basnet: Boundary-aware salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7479–7489, 2019.
- [32] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention*, pages 234–241, 2015.
- [33] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. ACM TOG, 23:309–314, 2004.
- [34] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *International Conference on Computer Vision*, pages 618–626, 2017.
- [35] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *Transactions on Pattern Analysis and Machine Intelligence*, 39 (4):640–651, 2017.
- [36] Chunfeng Song, Yan Huang, Wanli Ouyang, and Liang Wang. Box-driven class-wise region masking and filling rate guided loss for weakly supervised semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3136– 3145, 2019.
- [37] Guolei Sun, Wenguan Wang, Jifeng Dai, and Luc Van Gool. Mining cross-image semantics for weakly supervised semantic segmentation. *European Conference on Computer Vision*, 2020.

- [38] Meng Tang, Abdelaziz Djelouah, Federico Perazzi, Yuri Boykov, and Christopher Schroers. Normalized cut loss for weakly-supervised CNN segmentation. In *Conference on Computer Vision and Pattern Recognition*, pages 1818–1827, 2018.
- [39] Meng Tang, Federico Perazzi, Abdelaziz Djelouah, Ismail Ben Ayed, Christopher Schroers, and Yuri Boykov. On regularized losses for weakly-supervised CNN segmentation. In *European Conference on Computer Vision*, pages 524–540, 2018.
- [40] Olga Veksler. Regularized loss for weakly supervised single class semantic segmentation. In European Conference on Computer Vision, pages 348–365. Springer, 2020.
- [41] Paul Vernaza and Manmohan Chandraker. Learning random-walk label propagation for weakly-supervised semantic segmentation. In *Conference on Computer Vision and Pattern Recognition*, pages 2953–2961, 2017.
- [42] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *European Conference on Computer Vision*, pages 418–434, 2018.
- [43] Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Conference on Computer Vision* and Pattern Recognition, pages 5987–5995, 2017.
- [44] Jia Xu, Alexander G. Schwing, and Raquel Urtasun. Learning to segment under various forms of weak supervision. In *Conference on Computer Vision and Pattern Recognition*, pages 3781–3790, 2015.
- [45] Qi Yao and Xiaojin Gong. Saliency guided self-attention network for weakly and semisupervised semantic segmentation. *IEEE Access*, 8:14413–14423, 2020.
- [46] Yu Zeng, Yunzhi Zhuge, Huchuan Lu, and Lihe Zhang. Joint learning of saliency detection and weakly supervised semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7223–7233, 2019.
- [47] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Conference on Computer Vision and Pattern Recognition*, July 2017.
- [48] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and P. Torr. Conditional random fields as recurrent neural networks. In *International Conference on Computer Vision*, pages 1529–1537, 2015.
- [49] Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Conference on Computer Vision and Pattern Recognition*, pages 2921–2929, 2016.