Low Shot Box Correction for Weakly Supervised Object Detection

Tianxiang Pan[1,2], **Bin Wang**[1,2*], **Guiguang Ding**[1,2], **Jungong Han**[3] and **Junhai Yong**[1,2]
[1]School of Software, Tsinghua University, China
[2]Beijing National Research Center for Information Science and Technology (BNRist), China
[3]WMG Data Science, University of Warwick, CV4 7AL Coventry, United Kingdom
ptx9363@gmail.com, {wangbins, dinggg, yongjh}@tsinghua.edu.cn, jungonghan77@gmail.com

Abstract

Weakly supervised object detection (WSOD) has been widely studied but the accuracy of state-of-art methods remains far lower than strongly supervised methods. One major reason for this huge gap is the incomplete box detection problem which arises because most previous WSOD models are structured on classification networks and therefore tend to recognize the most discriminative parts instead of complete bounding boxes. To solve this problem, we define a low-shot weakly supervised object detection task and propose a novel low-shot box correction network to address it. The proposed task enables to train object detectors on a large data set all of which have image-level annotations, but only a small portion or few shots have box annotations. Given the low-shot box annotations, we use a novel box correction network to transfer the incomplete boxes into complete ones. Extensive empirical evidence shows that our proposed method yields state-of-art detection accuracy under various settings on the PASCAL VOC benchmark.

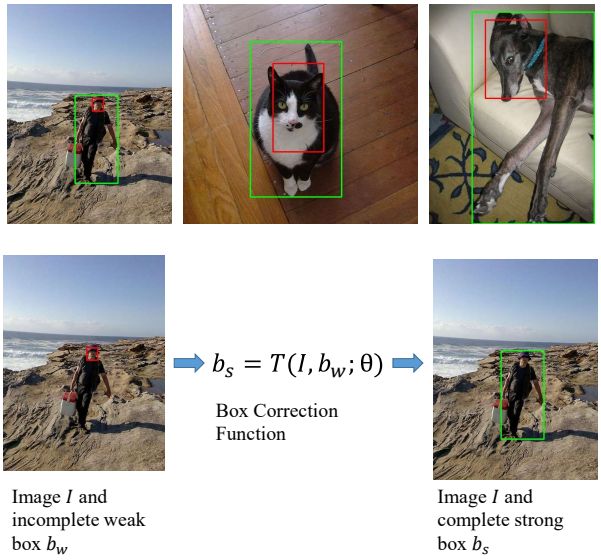

Figure 1: Images in first row show the incomplete box detections (red, small) predicted by state-of-art WSOD method and manually annotated complete bounding boxes (green, big). In general, our method is trying to learn a transfer function T that can predict strong boxes from weak boxes. θ is the model parameter.

1 Introduction

Weakly Supervised Object Detection (WSOD) has been widely studied because annotating a large-scale dataset with bounding boxes is expensive and time-consuming. For WSOD methods, datasets with only image-level annotations are required to do the task of detecting objects. Although a lot of great works have been made in this area, the accuracy of state-of-art methods remains far lower than their strongly supervised baselines (0.47 vs 0.67 mAP by Fast-RCNN on PASCAL VOC 2007).

We think one of the major reasons for this huge gap might be the incomplete box detection. Here, we take a simple example to illustrate this problem. Given an image with label *person* like the first image in Figure 1, the trained detector only knows that there's a person in this image but has no idea of where she/he is. Most of previous methods are structured on deep classification networks and tend to recognize the most discriminative parts (red boxes) instead of complete object bounding boxes (green boxes). In the above example, these models are tended to recognize the person's head instead of the full body because heads are similar and easy to distinguish while the full bodies are much more diverse and harder to distinguish due to different clothes or postures. For some object labels like person, their most discriminative parts are much smaller than complete bounding boxes and thus will result in a very low detection accuracy. Through the observation, we think incomplete box detection is the main reason why previous methods perform badly on these object classes.

To address the incomplete box detection problem, it is necessary to introduce auxiliary information about the true complete bounding boxes. Inspired by few-shot and semi-supervised learning methods, we introduce a low-shot weakly supervised object detection task in this paper and propose a novel box correction method to address it. The definition of low-shot weakly supervised object detection task is: given a large dataset with image-level classification labels and only a

*Corresponding Author

small subset of it has bounding box annotations, the model is expected to detect the complete objects, rather than a part of them. The term *low-shot* here includes both semi-supervised and few-shot settings so that we use this term for brevity.

For this low-shot weakly supervised object detection task, we propose a novel box transfer method to correct the incomplete box predictions. Our method starts with generating *weak boxes* and *strong boxes* pairs. Here, weak boxes are those incomplete boxes predicted by common weakly supervised methods which are trained only by image-level annotations, while strong boxes are the ground truth complete bounding boxes annotated manually. In the low-shot task, all images have weak boxes while only a small subset (e.g. 10%) of images have both weak and strong boxes. For images with ground truth boxes, we use a greedy method to select possible weak-to-strong box pairs and design a *box transfer function* to learn a transformation from weak boxes to strong boxes. For images without ground truth boxes, we apply the learned box transfer function on them to generate pseudo complete bounding boxes. These generated pseudo boxes are trained together to maintain the model's diversity and avoid overfitting into the small subset.

We explore our method in two different settings. 1) Semi-supervised learning: Only a small proportion of images have box annotations, irrespective of these images' class labels. 2) Few-shot learning: Only a few shots of images have box annotations for each class while different classes have the same number of images. We compare our method with most available methods under these two settings on PASCAL VOC dataset. Our method achieves state-of-art accuracy in most experiments.

2 Related Work

2.1 Weakly Supervised Object Detection.

WSOD has attracted much attention in the last decade. In recent years, due to the breakthrough brought by deep learning, a large amount of deep models are proposed for WSOD. Most of these models follow a similar pipeline: box proposal initialization, proposal classification, proposal refinement and retraining. We classify these models into three categories according to their mainly contributed pipeline steps.

For proposal initialization, Selective Search [Uijlings *et al.*, 2013] and Edge Boxes [Zitnick and Dollár, 2014] are the most commonly used proposal methods. Zhu et al. [2017] propose to train an integrated weakly supervised object proposal network, based on the class activation maps. Different from [Zhu *et al.*, 2017], Tang et al. [2018] propose to use a multi-layer fused attention map to refine the initial proposals.

For proposal classification, Bilen and Vedaldi [2016] employ a two-branch network, called WSDDN, to simultaneously perform region selection and classification. However, WSDDN only uses an image-level classification loss which makes it tend to recognize the most discriminative parts. To address the problem, [Diba *et al.*, 2017] and [Lai and Gong, 2017] introduce salient segmentation attention maps to regularize the training. The introduced segmentation regularization is helpful but not enough to address the huge gap between those discriminative parts and the complete bounding boxes. Our method explicitly formulates the transfer function to correct the proposals instead of adding regularization. This makes our method more effective than previous models.

For proposal refinement and retraining, most of these methods are heuristic. Tang et al. [2017] employ WSDDN as the basic network and refine the prediction with several Online Instance Classifier Refinement (OICR) branches. Ge et al. [2018] propose a multi-task ensemble framework to refine the weakly detection results by some other tasks. After proposal refinement, the refined proposals are commonly used to train a strongly supervised methods. Zhang et al. [2018] heuristically employ an easy to hard training strategy to discover reliable proposals. Most of these refinement and retraining strategies heavily rely on the initialization. If the initial model is trapped in false incomplete detection, these strategies can not efficiently correct it.

2.2 Few Shot Object Detection

Dong et al. [2018] propose to generate trustworthy training samples for few-shot object detection task. They iteratively train model and select high-confidence samples for retraining. Compared with them, our proposed task has more annotations because we have image-level labels for all images. Through experiments, the image-level labels are essential for accuracy improvement. We argue that our task is reasonable since attaining image-level labels is much easier than bounding boxes.

2.3 Semi Supervised Object Detection

Yan et al. [2017] have designed an EM model for semi supervised object detection task. They combine the ground truth boxes with image-level labels to iteratively retrain the EM model. The biggest difference between their model and ours is that we explicitly separate the *weak boxes* and *strong boxes* while Yan et al. just put them together. As we shown above, these two boxes may have large difference in size and putting them together may be contradictory in training. Wang et al. [2018] design a similar model with [Yan *et al.*, 2017] which employs Faster RCNN as a base network. Experiments confirm that our method outperforms both of Yan and Wang's model in various semi supervised settings.

2.4 Partially Supervised Learning

Hu et al. [2018] propose a new partially supervised training paradigm that enables training instance segmentation models on a large data set which has box annotations, but only a small fraction of which have mask annotations. The proposed paradigm is quite similar to our task. The major difference is that we have a large data set with only image-level annotations while they have box annotations. Since the state-of-arts WSOD accuracy is still unsatisfactory while the weakly semantic segmentation methods [Khoreva *et al.*, 2017] are approaching their supervised counterparts, we suppose the task from image-level annotation to box is harder than the task from box to segmentation mask.

3 Method

Let $\mathcal{I}$ be all the images having image-level annotations. We split $\mathcal{I}$ into two parts, $\mathcal{I} = \mathcal{A} \cup \mathcal{B}$, where images in $\mathcal{A}$ have

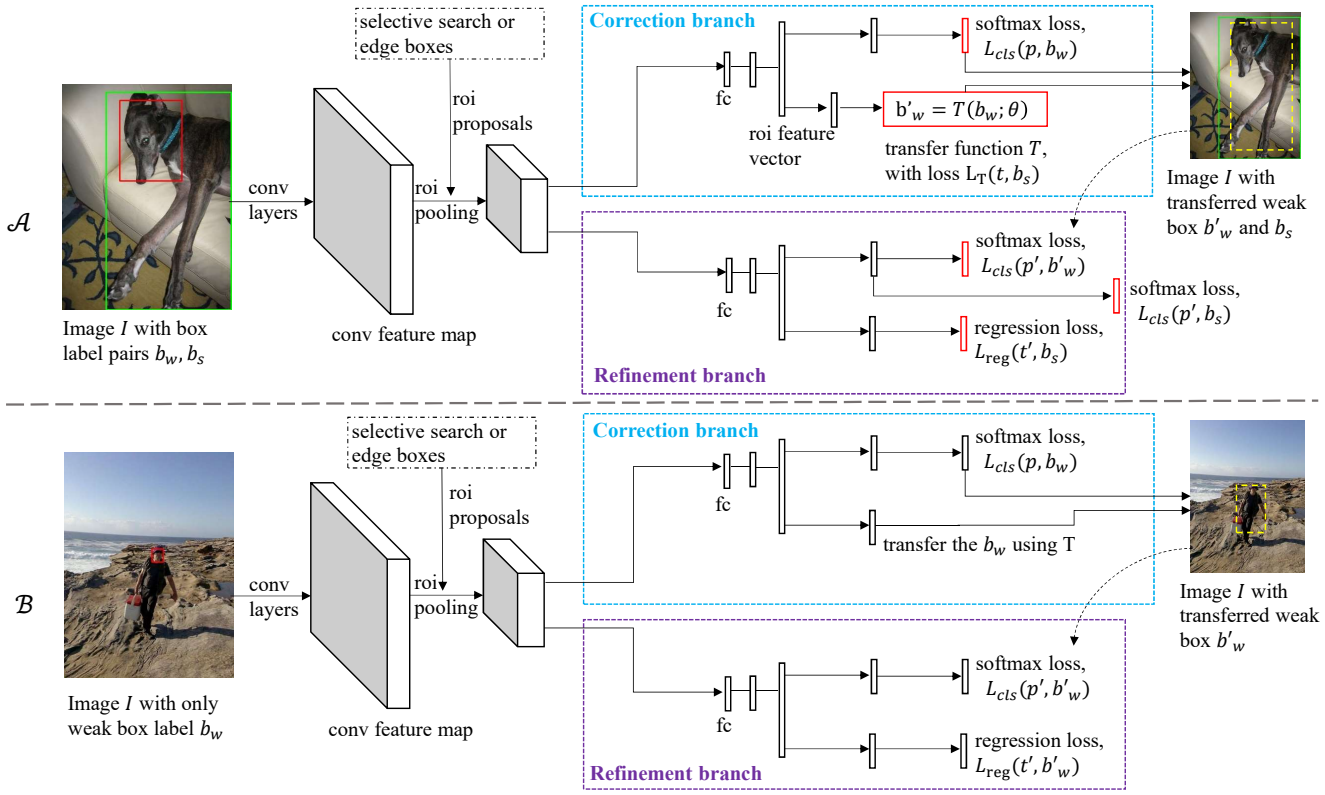

Figure 2: Our Box Correction Network (BCNet) Model architecture. The whole dataset is split into two parts, $\mathcal{A}$ and $\mathcal{B}$. Images in $\mathcal{A}$ have both image-level and box annotations while images in $\mathcal{B}$ have only image-level annotations. Our method firstly generates weak boxes b_w for all images and calculates the weak-to-strong box pairs for images in $\mathcal{A}$. Then the images and boxes are fed into a box correction network branch to train a transfer function from weak boxes to strong boxes. $L_{cls}(p, b_w)$ means that the classification loss function's predict target is calculated using roi and weak boxes. $L_T(t, b_s)$ means that this transfer loss function's offset target is calculated using roi and strong boxes. We use T to transfer the weak boxes into b'_w and then train a new refinement branch to jointly fit b'_w and b_s.

both image-level and box annotations while images in $\mathcal{B}$ are only image-level annotated. Note that under the low-shot settings, $\mathcal{A}$ is a small part of $\mathcal{I}$ (i.e. 10%) and $\mathcal{A} \cap \mathcal{B} = \emptyset$.

Our method starts from obtaining weak and strong boxes. We pretrain a weakly supervised object detection model and use it to generate pseudo bounding boxes for all images in $\mathcal{I}$. These generated pseudo bounding boxes are denoted as their weak boxes, b_w. For images in $\mathcal{A}$, their manually labeled bounding boxes are denoted as strong boxes, b_s. Each box is defined by a four-tuple (x, y, h, w) that specifies its center coordinate (x, y) and its height and width (h, w). Weak boxes are $(b_w^x, b_w^y, b_w^w, b_w^h)$ and strong boxes are $(b_s^x, b_s^y, b_s^w, b_s^h)$.

In general, our method is trying to learn a transfer function T that can predict strong boxes from weak boxes,

$$b_s = T(b_w; \theta). \tag{1}$$

Before learning the transfer function between weak and strong boxes, we need to generate one-to-one, weak-to-strong box pairs. Only images in $\mathcal{A}$ can be used to generate the one-to-one pairs. We design a greedy algorithm to accomplish this. Given an image with its box b_w and b_s, we firstly calculate the overlaps between b_w and b_s. Then, each weak box is assigned a corresponding strong box which owns the largest overlap with the same label as b_w. If the largest overlap is smaller than a threshold, this pair will be removed.

Having generated the weak boxes for $\mathcal{B}$ and weak-to-strong boxes for $\mathcal{A}$, we'll utilize both images in $\mathcal{A}$ and $\mathcal{B}$ to train a box correction network for the low-shot weakly supervised object detection task. The overall network structure is shown in Figure 2. Details will be introduced respectively in next subsections.

3.1 Box Correction

Given the weak and strong box pairs for $\mathcal{A}$, instead of directly predicting strong boxes b_s, we formulate the transfer function to predict the offset between b_w and b_s. The offset targets are defined as $t = (t^x, t^y, t^w, t^h)$, in which

$$\begin{aligned} t^x = (b_s^x - b_w^x)/b_w^w,\ t^y = (b_s^y - b_w^y)/b_w^h, \\ t^w = log(b_s^w/b_w^w),\ t^h = log(b_s^h/b_w^h). \end{aligned} \tag{2}$$

We can easily attain b_s by t and b_w.

This formulation is similar to the regression in Fast-RCNN [Girshick, 2015] while we employ it to transfer different boxes instead of refining location. For clarity, we denote this offset prediction function as F and our general transfer

function T is replaced by F,

$$t = F(b_w; \theta). \tag{3}$$

The offset prediction function F can be implemented as a fully connected neural network. In training, we apply a $smooth_{L1}$ loss function to learn F,

$$L_{loc}(t, v) = \sum_{i \in \{x,y,w,h\}} smooth_{L1}(v^i - t^i), \tag{4}$$

in which v is the coordinates and height/width offset predicted by F.

Notice that the box-annotated data set $\mathcal{A}$ is a very small part of $\mathcal{I}$. If we only train the transfer function T with the weak-to-strong box pairs, the number of training set may be too small such that the model may easily overfit into these limited samples. In order to enrich the training samples, we integrate the transfer function T together with roi proposal classification. In Fast-RCNN, some roi proposals are pre-calculated by Selective Search [Uijlings *et al.*, 2013]. These proposals are utilized to jointly learn object classification and bounding box regression. The original loss function of Fast-RCNN is

$$L(p, u, t, v) = L_{cls}(p, u) + L_{reg}(t, v), \tag{5}$$

where p is the roi class predictions and t is the predicted offset between rois and targets. u is the class label and v is the target offset. Only rois with foreground label will contribute to the regression loss L_{reg}.

We adapt this loss function to introduce the T function as

$$L_{\mathcal{A}}(p, u_w, t, v_s) = L_{cls}(p, u_w) + L_T(t, v_s). \tag{6}$$

Here, u_w is the class label calculated from weak boxes b_w, and v_s is the offset target between rois and strong boxes b_s. Only rois with foreground label in u_w will contribute to the regression loss L_T.

Comparing with the original loss function, the regression part in the adapted function is no longer used for refining location. In our loss function, the classification loss attempt to select rois who have a large overlap with weak boxes b_w while the regression loss is used to predict the offset between these selected rois and paired strong boxes b_s. This way ensures the transfer function T can be trained by all possible proposals, thus enabling a better transfer between weak and strong boxes.

3.2 Balanced Training

For images in $\mathcal{A}$, the loss function is shown as equation 6. For images in $\mathcal{B}$, since they only have weak boxes, as a result, we can only train the classification part of loss function

$$L_{\mathcal{B}}(p, u_w, t, v_s) = L_{cls}(p, u_w). \tag{7}$$

Note that the size of $\mathcal{A}$ is much smaller than $\mathcal{B}$ in our proposed low-shot tasks. In order to fully exploit all the images in $\mathcal{B}$, we use the trained box correction function to generate pseudo corrected boxes for images in $\mathcal{B}$ and use all of the transferred boxes in $\mathcal{A} \cup \mathcal{B}$ to learn a new refinement branch of proposal classification and regression.

For each training iteration, one image z^a from $\mathcal{A}$ and one image z^b from $\mathcal{B}$ are trained together to make balanced training samples. z^a and z^b are firstly fed into a box correction branch with the following loss function,

$$\begin{aligned} L_1 =& L_{\mathcal{A}} + L_{\mathcal{B}} \\ =& L_{cls}(p^a, u_w^a) + L_T(t^a, v_s^a) + L_{cls}(p^b, u_w^b) \end{aligned} \tag{8}$$

The outputs of the first box correction branch are transferred boxes, denoted as $b_w'^a$ and $b_w'^b$. These boxes are in turn used to train a second refinement branch. We suppose the second branch is functionally the same as Fast-RCNN whose regression loss is used to refine location so that a classification loss by b_s^a is utilized to jointly train this branch as shown in Figure 2. The entire loss of the second branch is

$$\begin{aligned} L_2 =& L'_{\mathcal{A}} + L'_{\mathcal{B}} + L'_{strong} \\ =& L_{cls}(p'^a, u_w'^a) + L_{reg}(t'^a, v_s'^a) + L_{cls}(p'^b, u_w'^b) \\ &+ L_{reg}(t'^b, v_w'^b) + L_{cls}(p'^a, u_s^a), \end{aligned} \tag{9}$$

where u_s^a is the class label calculated from strong boxes for z^a. In Figure 2, we replace u, v with b_s or b_w to show it more clearly that which boxes are used in particular loss.

3.3 Multi-Stage Correction

During experiments, we find that the learned transfer function T is able to transfer the weak boxes towards the strong ones. However, in some cases, if the strong boxes are much bigger than the weak boxes, T can only enlarge the boxes a little but the enlarged ones are still much smaller than the strong boxes. We think the result is caused by the $smooth_{L1}$ loss function,

$$smooth_{L1}(x) = \begin{cases} 0.5x^2, & if|x| < 1 \\ |x| - 0.5 & otherwise. \end{cases}$$

If the strong boxes are much bigger than the weak boxes, $|x|$ may be very large while training. Though $smooth_{L1}$ has the same gradient when $|x| \geq 1$, it can not effectively give enough loss penalty on these samples to cover the huge gap between the strong and weak boxes.

To address this problem, we introduce a simple but effective technique, the multi-stage correction. The intuition is straightforward. If a single transfer function can only make a limited step from the weak boxes towards the strong boxes, why not continuously make several steps? In the proposed multi-stage correction, a later stage uses the transferred boxes produced by its previous stage as the weak boxes and learns a new transfer function T' for current stage. Only the last stage has the classification loss by the strong boxes which is $L_{cls}(p', b_s)$ in Figure 2 because we treat it as a correct box detection branch rather than a weak-to-strong transfer branch.

3.4 Image-level Regularization

In the multi-stage model, our method actually follows an iterative training strategy, training a function to transfer boxes and then using the transferred boxes for retraining. One concern over this strategy is that it may amplify the noise from the first stage. To avoid this dilemma, we introduce an image-level regularization function in the final stage since all the images have image-level annotations.

Methods, Backbone	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	moto	person	plant	sheep	sofa	train	tv	mAP(%)
Only image-level labels (lower bounds):																					
WSDDN[Bilen and Vedaldi, 2016], AlexNet	42.9	56.0	32.0	17.6	10.2	61.8	50.2	29.0	3.8	36.2	18.5	31.1	45.8	54.5	10.2	15.4	36.3	45.2	50.1	43.8	34.5
OICR[Tang *et al.*, 2017], VGG	58.5	63.0	35.1	16.9	17.4	63.2	60.8	34.4	8.2	49.7	41.0	31.3	51.9	64.8	13.6	23.1	41.6	48.4	58.9	58.7	42.0
OICR+FasterRCNN, VGG	65.5	67.2	47.2	21.6	22.1	68.0	68.5	35.9	5.7	63.1	49.5	30.3	64.7	66.1	13.0	25.6	50.0	57.1	60.2	59.0	47.0
[Ge *et al.*, 2018]+FasterRCNN, VGG	64.3	68.0	56.2	36.4	23.1	68.5	67.2	64.9	7.1	54.1	47.0	57.0	69.3	65.4	20.8	23.2	50.7	59.6	65.2	57.0	51.2
Image label and semi-supervised boxes:																					
EM [Yan *et al.*, 2017], AlexNet, 10% boxes	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	48.3*
Our BCNet, AlexNet, 10% boxes	58.3	60.4	38.3	27.8	27.7	62.9	70.1	63.6	27.9	53.2	47.2	52.1	67.0	64.5	57.5	24.1	48.7	46.1	65.0	56.6	51.0
[Wang *et al.*, 2018], VGG, 16% boxes	58.2	75.9	56.6	45.2	39.6	73.2	75.8	77.2	38.4	65.7	61.0	72.3	78.6	67.3	68.1	33.0	61.5	61.1	72.1	66.7	62.4
Our BCNet, VGG, 16% boxes	63.7	77.2	62.9	48.0	39.7	73.3	76.0	78.0	39.4	72.9	56.1	75.4	79.9	69.5	70.2	31.0	60.6	62.2	75.0	68.6	64.0
Only few-shot boxes:																					
[Dong *et al.*, 2018], ResNet101, 4 shots	46.6	55.6	37.9	26.1	27.9	46.6	57.9	58.1	24.1	37.6	12.8	33.1	51.4	59.7	40.1	17.5	36.1	52.0	61.4	52.1	41.7
Image label and few-shot boxes:																					
[Dong *et al.*, 2018], ResNet101, 4 shots	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	48.0*
Our BCNet, VGG, 4 shots	60.5	65.6	47.3	27.4	29.1	69.9	67.9	62.7	23.5	63.3	36.8	49.6	58.4	67.5	54.3	18.0	55.3	56.6	60.2	66.0	52.0
Our BCNet, ResNet101, 4 shots	61.6	68.1	46.5	30.4	24.9	67.0	64.5	66.2	26.5	64.6	41.1	56.3	57.6	61.7	53.6	22.7	52.5	61.5	66.0	63.7	52.8
Fully supervised (upper bounds):																					
Fast RCNN, VGG	74.5	78.3	69.2	53.2	36.6	77.3	78.2	82.0	40.7	72.7	67.9	79.6	79.2	73.0	69.0	30.1	65.4	70.2	75.8	65.8	66.9
Faster RCNN, VGG	70.0	80.6	70.1	57.3	49.9	78.2	80.4	82.0	52.2	75.3	67.2	80.3	79.8	75.0	76.3	39.1	68.3	67.3	81.1	67.6	69.9
More results:																					
Our BCNet, VGG, 10% boxes	64.7	73.1	55.2	37.0	39.1	73.3	74.0	75.4	35.9	69.8	56.3	74.7	77.6	71.6	66.9	25.4	61.0	61.4	73.8	69.3	61.8
Our BCNet, ResNet101, 10% boxes	68.3	72.0	61.2	48.1	40.8	73.3	73.4	77.8	37.0	69.7	58.3	78.2	80.0	67.5	70.5	27.4	62.9	63.6	73.4	63.6	63.4
Our BCNet, ResNet101, 16% boxes	67.3	74.2	65.2	51.7	40.8	74.1	72.7	77.2	39.2	70.3	59.9	77.2	78.5	69.9	68.6	30.6	60.0	68.2	75.9	66.8	64.4
Our BCNet, VGG, 10 shots	59.7	69.1	44.6	29.4	40.1	69.2	73.2	72.9	32.9	58.1	53.3	66.7	71.3	66.0	61.7	24.6	53.0	62.0	67.2	67.4	57.1
Our BCNet, ResNet101, 10 shots	63.4	69.4	54.7	39.5	35.9	70.6	71.8	71.8	33.5	64.6	50.0	65.3	72.7	62.5	61.6	29.2	54.5	63.3	66.7	69.4	58.5
Our BCNet, ResNet101, 20 shots	66.5	67.6	56.7	40.5	40.4	72.8	71.3	76.6	39.4	65.0	54.1	71.4	72.9	66.6	66.0	26.1	59.0	65.5	67.7	67.6	60.7

Table 1: mAP (%) on the PASCAL VOC 2007 test set. For results with *, their authors only plot the mAP in figures and what we report are the estimated results. The BCNet refers to our **B**ox **C**orrection **Net**work with multi-stage correction and image-level regularization. 10% boxes means that 10% of all images have box annotations. The 10% images actually correspond to 500 images in VOC 2007, and 16% correspond to 811 images. 4 shots means that each class only has 4 images with box annotations.

For an image, let p_{ic} be the ith roi prediction for class c. We use the maximum p_{ic} across rois to represent the image-level prediction for class c, that is $q_c = \max_i(p_{ic})$. The image-level regularization is defined as the binary cross entropy with q_c and the image-level label y_c,

$$L_{img} = -\frac{1}{C}\sum_c y_c log(q_c) + (1 - y_c) log(1 - q_c). \quad (10)$$

4 Experiments

4.1 Experiment Settings

We evaluate our method on the PASCAL VOC 2007 data set which contains 5011 training images and 4952 images for test. The evaluation metric is mean Average Precision (mAP) which is commonly used in object detection. In experiments, we explore our method in two different settings, semi-supervised and few-shot. For semi-supervised settings, only a limited number of images have box annotations, irrespective of these images' labels. For few-shot settings, different labels have a same number of images with box annotations.

4.2 Implementation Details

Network backbone. We mainly use the VGG16 as base network for our experiments because most of the previous WSOD methods are structured on it. For fair comparison with [Yan *et al.*, 2017], we also report results based on AlexNet. In addition, we give the results on ResNet for reference.

Pretraining. We use OICR [Tang *et al.*, 2017] to generate the weak boxes. In experiments, we find that only using ImageNet pre-trained weight as initialization performs poorly for small networks, e.g. AlexNet, or with very small number of box annotations, e.g. 4 shots. We suppose this is caused by the limited generalization ability in these experiments. Instead of using ImageNet pre-trained weight, we employ a Fast-RCNN which is trained by weak boxes to initialize our model in these experiment.

Training and inference. For fair comparison with the prior arts, we use Edge Boxes as our roi proposal method. SGD is used to optimize the models. In experiments, models are fine-tuned with 60 epochs. The learning rate is 0.001 in the first 40 epochs and will be reduced to 0.0001 in the last 20 epochs. All other hyper-parameters follow those in Fast-RCNN. We resize the image minimum side into $\{400, 600, 750\}$ as data augmentation. To jointly train the model with images from $\mathcal{A}$ and $\mathcal{B}$, we randomly crop the input images by a fixed 600×600 window. In most cases, two-stage box correction branchs are used for multi-stage correction.

Low shot selection. One concern over the experiments is the sensitivity of our method to different selection of box annotated images. To figure out whether different selection of box annotated images can greatly influence the results, we have conducted several experiments and find that for experiments with more than 10% of all or 10 shots box annotations, different selections have very little effect on final accuracy. For experiment with very small number of box annotations, an average result is reported instead.

Reproducibility. We implement our method on Pytorch and Code is available at https://github.com/ptx9363/BCNet.

4.3 Comparison with State-of-the-arts

We evaluate our model on PASCAL VOC 2007 test benchmark and compare it with most available state-of-art methods under semi-supervised or few-shot settings. The results are shown in Table 1.

For methods with image labels and semi-supervised boxes, our method outperforms all of previous methods under the equal semi-supervised settings. Both [Yan *et al.*, 2017] and [Wang *et al.*, 2018] treat the weak boxes and the strong boxes equally while our method explicitly separate them. The comparison results confirm the necessity and effectiveness of this separation technique.

For methods with image labels and few-shot boxes, our method also performs better than these methods in equal settings. Dong et al. [Dong *et al.*, 2018] have proposed to use image-level labels as an object filter in their paper. Compared with them, we have fully utilized the images in $\mathcal{B}$ and used balanced box transfer to get a better prediction.

By comparing our BCNet with common WSOD methods, we find that introducing the box correction transfer network can bring huge improvement on these methods. We would like to highlight that even if only 4 images with box annotations are provided, our method could greatly improve the prediction accuracy of OICR (our base WSOD model) for some classes like person (13.6% to 54.3%), cat (34.4% to 62.7%), dog (31.3% to 49.6%). These huge improvements show that our BCNet is robustly effective for helping some weakly supervised object detection methods which may have serious incomplete detection problem.

By comparing our BCNet with previous few-shot methods, we find that the image-level labels are essential for accuracy improvement (52.0% vs 41.7%). Since attaining image-level labels are much easier than box annotations, e.g. from search engine, we think our model is more practical in real world applications.

4.4 Ablation Study

Effectiveness of components. Firstly we conducted experiments to understand the different contribution of submodules. Tabel 2 shows the ablation results. We can see that the balanced training and box transfer are essential for our method. Without balanced training, we treat images equally from $\mathcal{A}$ and $\mathcal{B}$ while the size of $\mathcal{A}$ is much smaller than $\mathcal{B}$ so that the model can not fit the transfer function T well enough. Without box transfer, we directly finetune the features from WSDO with few shot annotated boxes and the results confirm that transferring boxes is more effective than transferring features.

Amounts of box annotations. We also conducted experiments by varying box annotations proportions in semi-supervised task and number of shots in few-shot task. The results are summarized in Figure 3 and Table 3. As illustrated in Figure 3, our method robustly performs better than the state-of-art method [Yan *et al.*, 2017]. When 40% boxes are annotated, our method achieves 56.0% mAP which is only 1.1 lower than fully supervised upper bound. We compare our method with [Wang *et al.*, 2018] by varying number of annotation shots in Table 3. It's important to note that Wang employs a mixed model which contains several different base networks while our method is structured on a single base network. Therefore, their model owns a much higher upper bound (74.8%) than ours (68.9%) as shown in the table and we think it explains why their model performs better in 20-shot experiments. Even if only be structured on single ResNet101, our method achieves better results in 4-shot and 10-shot experiments. It confirms the effectiveness of our method on few-shot tasks.

BCNet on AlexNet						
w/o balanced training		√	√	√	√	√
w/o box transfer			√	√	√	√
w/o multi-stage				√		√
w/o image-level					√	√
VOC07 mAP(%)	45.6	47.7	49.7	50.6	50.3	51.0

Table 2: Ablation experiments on AlexNet for balanced training, box transfer, multi-stage correction and image-level regularization.

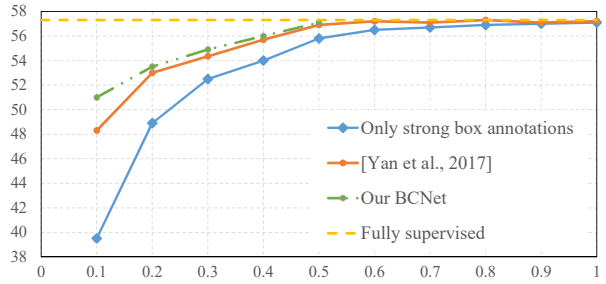

Figure 3: Semi-supervised performance (mAP%) by varying box annotation proportions. For fair comparison, our models are based on AlexNet.

methods	4-shot	10-shot	20-shot	Fully supervised
Our BCNet	52.8	58.5	60.7	68.9
[Wang *et al.*, 2018]	48.0	56.9	63.1	74.8

Table 3: Few-shot performance (mAP%) by varying number of box annotation shots. Wang's results are estimated from their figures.

5 Conclusion and Future Work

In this paper, we are trying to solve the incomplete detection problem that exists in most previous weakly supervised object detection methods. We introduce a low-shot weakly supervised object detection task and propose a novel Box Correction Network (BCNet) to address it. We explicitly separate the incomplete boxes and complete boxes and our BCNet attempts to learn a transfer function to correct those incomplete boxes into complete ones. Experiment results show that BCNet is more effective than previous models.

Exploring more forms of transfer function will be our future works. In addition, how to learn a better transfer function by using advanced semi-supervised method or few shot learning method (e.g. *vat* [Miyato *et al.*, 2018] or *maml* [Finn *et al.*, 2017]) is another problem that we are interested in.

Acknowledgements

This work was supported by the NSFC under Grants 61772301, 61672307 and 61571269.

References

[Bilen and Vedaldi, 2016] Hakan Bilen and Andrea Vedaldi. Weakly supervised deep detection networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2846–2854, 2016.

[Diba *et al.*, 2017] Ali Diba, Vivek Sharma, Ali Pazandeh, Hamed Pirsiavash, and Luc Van Gool. Weakly supervised cascaded convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5131–5139, 2017.

[Dong *et al.*, 2018] X Dong, L Zheng, F Ma, Y Yang, and D Meng. Few-example object detection with model communication. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2018.

[Finn *et al.*, 2017] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135, 2017.

[Ge *et al.*, 2018] Weifeng Ge, Sibei Yang, and Yizhou Yu. Multi-evidence filtering and fusion for multi-label classification, object detection and semantic segmentation based on weakly supervised learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1277–1286, 2018.

[Girshick, 2015] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015.

[Hu *et al.*, 2018] Ronghang Hu, Piotr Dollár, Kaiming He, Trevor Darrell, and Ross Girshick. Learning to segment every thing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4233–4241, 2018.

[Khoreva *et al.*, 2017] Anna Khoreva, Rodrigo Benenson, Jan Hosang, Matthias Hein, and Bernt Schiele. Simple does it: Weakly supervised instance and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 876–885, 2017.

[Lai and Gong, 2017] Baisheng Lai and Xiaojin Gong. Saliency guided end-to-end learning forweakly supervised object detection. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 2053–2059, 2017.

[Miyato *et al.*, 2018] Takeru Miyato, Shin-ichi Maeda, Shin Ishii, and Masanori Koyama. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2018.

[Tang *et al.*, 2017] Peng Tang, Xinggang Wang, Xiang Bai, and Wenyu Liu. Multiple instance detection network with online instance classifier refinement. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2843–2851, 2017.

[Tang *et al.*, 2018] Peng Tang, Xinggang Wang, Angtian Wang, Yongluan Yan, Wenyu Liu, Junzhou Huang, and Alan Yuille. Weakly supervised region proposal network and object detection. In *Proceedings of the European Conference on Computer Vision*, pages 352–368, 2018.

[Uijlings *et al.*, 2013] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 104(2):154–171, 2013.

[Wang *et al.*, 2018] Jiasi Wang, Xinggang Wang, and Wenyu Liu. Weakly-and semi-supervised faster r-cnn with curriculum learning. In *Proceedings of the IEEE 2018 24th International Conference on Pattern Recognition*, pages 2416–2421, 2018.

[Yan *et al.*, 2017] Ziang Yan, Jian Liang, Weishen Pan, Jin Li, and Changshui Zhang. Weakly-and semi-supervised object detection with expectation-maximization algorithm. *arXiv preprint arXiv:1702.08740*, pages 1–9, 2017.

[Zhang *et al.*, 2018] Xiaopeng Zhang, Jiashi Feng, Hongkai Xiong, and Qi Tian. Zigzag learning for weakly supervised object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4262–4270, 2018.

[Zhu *et al.*, 2017] Yi Zhu, Yanzhao Zhou, Qixiang Ye, Qiang Qiu, and Jianbin Jiao. Soft proposal networks for weakly supervised object localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1841–1850, 2017.

[Zitnick and Dollár, 2014] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *European Conference on Computer Vision*, pages 391–405, 2014.