WSOD^2: Learning Bottom-up and Top-down Objectness Distillation for Weakly-supervised Object Detection

Zhaoyang Zeng[1,2]*, Bei Liu[3], Jianlong Fu[3], Hongyang Chao[1,2], Lei Zhang[3]

[1] School of Data and Computer Science, Sun Yat-sen University

[2] The Key Laboratory of Machine Intelligence and Advanced Computing (Sun Yat-sen University), Ministry of Education

[3] Microsoft Research

zengzhy5@mail2.sysu.edu.cn; {bei.liu, jianf, leizhang}@microsoft.com; isschhy@mail.sysu.edu.cn

Abstract

We study on weakly-supervised object detection (WSOD) which plays a vital role in relieving human involvement from object-level annotations. Predominant works integrate region proposal mechanisms with convolutional neural networks (CNN). Although CNN is proficient in extracting discriminative local features, grand challenges still exist to measure the likelihood of a bounding box containing a complete object (i.e., "objectness"). In this paper, we propose a novel ***WSOD*** *framework with* ***O****bjectness* ***D****istillation (i.e.,* ***WSOD***^{2}*) by designing a tailored training mechanism for weakly-supervised object detection. Multiple regression targets are specifically determined by jointly considering bottom-up (BU) and top-down (TD) objectness from low-level measurement and CNN confidences with an adaptive linear combination. As bounding box regression can facilitate a region proposal learning to approach its regression target with high objectness during training, deep objectness representation learned from bottom-up evidences can be gradually distilled into CNN by optimization. We explore different adaptive training curves for BU/TD objectness, and show that the proposed WSOD*^{2} *can achieve state-of-the-art results.*

1. Introduction

The capability of recognizing and localizing objects in an image reveals a deep understanding of visual information, and has attracted many attentions in recent years. Significant progresses have been achieved with the development of convolutional neural network (CNN) [5, 14, 19, 27]. However, current state-of-the-art object detectors mostly rely on a large scale of training data which requires manually annotated bounding boxes (e.g., PASCAL VOC 2007/2012 [7], MS COCO [22], Open Images [20]). To relieve the heavy labeling effort and reduce cost, weakly-supervised object detection paradigm has been proposed by leveraging only image-level annotations [2, 30, 37, 38].

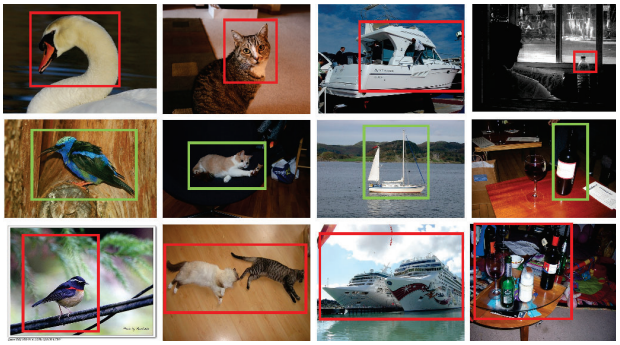


Figure 1: Typical weakly-supervised object detection results produced by OICR [30]. We can observe the partial, correct, and oversized detection results for an object instance in the first, second, and third row, respectively.

To address weakly-supervised object detection (WSOD) task, most previous works adopt multiple instance learning method to transform WSOD into multi-label classification problems [2, 18]. Later on, online instance classifier refinement (OICR) [29] and proposal cluster learning (PCL) [28] are proposed to learn more discriminative instance classifiers by explicitly assigning instance labels. Both OICR and PCL adopt the idea of utilizing the outputs of initial object detector as pseudo ground truths, which has been shown benefits in improving the classification ability of WSOD. However, a classification model often targets at detecting the existence of objects for a category, while it is not able to predict the location, size and the number of objects in images. This weakness usually results in the detection of partial or oversized bounding boxes, as shown in the first and third rows in Figure 1. The performances of OICR and PCL heavily rely on the accuracy of the initial object detection results, which limit further improvement with large margins. Also, they neglect learning bounding box regression, which plays an important role in the design of mod-

*This work was performed when Zhaoyang Zeng was visiting Microsoft Research as a research intern.

ern object detectors [3, 4, 13, 21, 24]. C-WSL integrates bounding box regressors into OICR framework to reduce localization errors, however, it relies on a greedy ground truths selection strategy which requires additional counting annotations [9].

Existing works that rely on the initial weakly-supervised object detection results try to learn the object boundary from feature maps by convolutional neural network (CNN). Although CNN is an expert to learn discriminative local features of an object with image-level labels in a top-down fashion (we call it top-down classifiers in this work), it performs poorly in detecting whether a bounding box contains a complete object without the ground truth for supervision.

Some low-level feature based object evidences (e.g. color contrast [23] and superpixels straddling [1]) have been proposed to measure a generic *objectness* that quantifies how likely a bounding box contains an object of any class in a bottom-up way. Inspired by these bottom-up object evidences, in this work, we explore to use their advantage for improving the capability of a CNN model in capturing objectness in images. We propose to integrate these bottom-up evidences that are good at discovering boundary and CNN with powerful representation ability in a single network.

We propose a **WSOD** framework with **O**bjectness **D**istillation (WSOD^2) to leverage bottom-up object evidences and top-down classification output with a novel training mechanism. First, given an input image with thousands of region proposals (e.g., generated by Selective Search [33]), we learn several instance classifiers to predict classification probabilities of each region proposal. Each of these classifiers can help to select multiple high-confident bounding boxes as possible object instances (i.e., pseudo classification and bounding box regression ground truths). Second, we incorporate a bounding box regressor to fine-tune the location and size of each proposal. Third, as each bounding box cannot capture precise object boundaries by CNN features alone, we combine bottom-up object evidences and top-down CNN confidence scores in an adaptive linear combination way to measure the objectness of each candidate bounding box, and assign labels for each region proposal to train the classifiers and regressor.

For some discriminative small bounding boxes that CNN prefers, the bottom-up object evidence (e.g., superpixels straddling) tends to be very low. WSOD^2 can regulate pseudo ground truths to satisfy both higher CNN confidence and low-level object completeness. In addition, a bounding box regressor is integrated to reduce the localization error, and augment the effect of bottom-up object evidences during training at the same time. We design an adaptive training strategy to make the guidance gradually distilled, which enables that a CNN model can be trained strong enough to represent both discriminative local and boundary information of objects when the model converges.

To the best of our knowledge, this work is the first to explore bottom-up object evidences in weakly-supervised object detection task. The contribution can be summarized as follows:

1. We propose to combine bottom-up object evidences with top-down class confidence scores in weakly-supervised object detection task.
2. We propose WSOD^2 (WSOD with objectness distillation) to distill object boundary knowledge in CNN by a bounding box regressor and an adaptive training mechanism.
3. Our experiments on PASCAL VOC 2007/2012 and MS COCO datasets demonstrate the effectiveness of the proposed WSOD^2.

2. Related Work

2.1. Weakly-supervised Object Detection

Weakly-supervised object detection has attracted many attentions in recent years. Most existing works adopt the idea of multiple-instance learning [2, 6, 17, 28, 29, 31, 34] to transform weakly-supervised object detection into multi-label classification problems. Bilen *et al.* [2] proposes WSDDN which performs multiplication on the score of classification and detection branches, so that high-confident positive samples can be selected. Tang *et al.* [28] and Tang *et al.* [29] find that online transforming image-level label into instance-level supervision is an effective way to boost the accuracy, and thus propose to online refine several branches of instance classifiers based on the outputs of previous branches. As class activation map produced by a classifier can roughly localize the object [39, 40], Wei *et al.* [36] tries to utilize it to generate course detection results, and use them as reference for the later refinement. Most previous works rely heavily on pseudo ground truths mining, either online (inside training loop) or offline (after training). Such pseudo ground truths are determined by classification confidence [28, 29] or hand-crafted rules [9, 38], which are not accurate to measure the objectness of regions.

2.2. Bounding Box Regression

Bounding box regression is proposed in [12], and is adopted by almost all recent CNN-based fully-supervised object detectors [3, 4, 13, 21, 24] since it can reduce the localization errors of predicted boxes. However, only a few works introduce bounding box into weakly-supervised object detection due to the lack of supervision. Some works consider bounding box regression as a post-processing module. Among which, OICR [29] directly uses the detection results of training set to train Fast R-CNN. W2F [38] designs some strategies to offline select pseudo ground truth with high precision, based on the output of OICR. Differently, Gao *et al.* [9] integrate bounding box regressors into OICR inside training loop which leverage addition counting

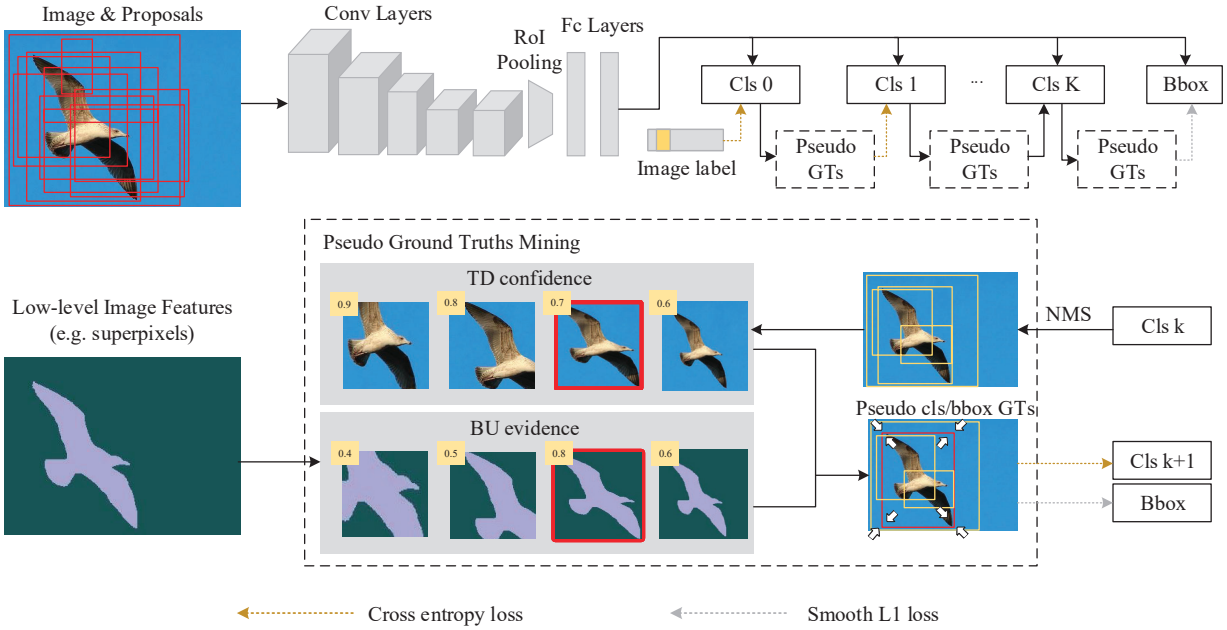


Figure 2: The framework of WSOD^2. Image with label and pre-computed proposals will be fed into a CNN to obtain region features. The region features will then be passed through several classifiers and a bounding box regressor. Non-maximum suppression (NMS) is applied to mine positive samples from the predictions. Top-down (TD) confidence and bottom-up (BU) evidence are computed by the classification branch and low-level image feature, respectively. They are combined to assign class labels and regression targets for each proposal. "Cls" indicates classifier, and "Bbox" indicates bounding box regressor. The white arrows indicate the optimization directions for two exemplar region proposals. [Best viewed in color]

information to help selecting pseudo ground truths.

In this paper, we integrate bounding box regressor into weakly-supervised detector, and assign regression targets by novelly leveraging bottom-up object evidence.

3. Approach

The overview of our proposed **w**eakly-**s**upervised **o**bject **d**etector with objectness **d**istillation (**WSOD**^{2}) is illustrated in Figure 2. We first adopt a based multiple instance detector (i.e. Cls 0) to obtain the initial detected object bounding boxes. Based on the localization of each proposed bounding box, we compute the bottom-up object evidence. Such evidence serves as guidance to transform image-level labels into instance-level supervision. We optimize the whole network in an end-to-end and adaptive fashion. In this section, we will introduce WSOD^2 in detail.

3.1. Based Multiple Instance Detector

In weakly-supervised object detection, only image-level annotations are available. To better understand semantic information inside an image, we need to go deep into region-level, and analyze the characteristic of each box. We first build a base detector to obtain initial detection result. We follow WSDDN [2] to adopt the idea of multiple instance learning [32] to optimize the base detector by transforming WSOD into multi-label classification problem. Specifically, given an input image, we first generate region proposals R by Selective Search [33] and extract region features $\mathbf{x}$ by a CNN backbone, an RoI Pooling layer and two fully-connected layers.

Region features $\mathbf{x}$ are then fed into two streams by two individual fully-connected layers, and the two produced feature matrices are denoted as $\mathbf{x}^c, \mathbf{x}^d \in \mathbb{R}^{C\times|R|}$, where C indicates the class number and $|R|$ denotes the proposal number. Two softmax functions are applied on $\mathbf{x}^c$ and $\mathbf{x}^d$ towards two distinct directions as follows:

$$[\sigma^c]_{ij} = \frac{e^{[\mathbf{x}^c]_{ij}}}{\sum_{k=1}^{C} e^{[\mathbf{x}^c]_{kj}}}, \quad [\sigma^d]_{ij} = \frac{e^{[\mathbf{x}^d]_{ij}}}{\sum_{k=1}^{|R|} e^{[\mathbf{x}^d]_{ik}}}, \tag{1}$$

where $[\sigma^c]_{ij}$ denotes the prediction of i^{th} class label for j^{th} region proposal, and $[\sigma^d]_{ij}$ is the weight learned of j^{th} region proposal for i^{th} class. We compute the proposal scores by element-wise product $s = \sigma^c \odot \sigma^d$, and aggregate over the region dimensions to obtain image-level score vector $\phi = [\phi_1, \phi_2, \cdots, \phi_C]$ by $\phi_c = \sum_{r=1}^{|R|} [s]_{cr}$. In such way, we can utilize the image-level class label as supervision and apply binary cross-entropy loss to optimize the base detector. The base loss function is denoted as:

$$L_{base} = -\sum_{c=1}^{C} (\hat{\phi}_c log(\phi_c) + (1-\hat{\phi}_c) log(1-\phi_c)), \tag{2}$$

where $\hat{\phi}_c = 1$ indicates that the input image contains c^{th} class, and $\hat{\phi}_c = 0$ otherwise. The prediction score s is considered as initial detection result. However, it is not precise enough and can be further refined as discussed in [29].

3.2. Bottom-up and Top-Down Objectness

The essence of an object detector is a bounding box ranking function, in which objectness measurement is an important factor. It is common to consider classification confidence as objectness score in recent CNN-based detectors [13, 24, 25]. However, such strategy has a flaw in weakly-supervised scenario that it is difficult for trained detectors to distinguish complete objects from discriminate object parts or irrelevant background. To relieve this issue, we explore bottom-up object evidences (e.g., superpixels straddling) which play important roles in traditional object detection.

As stated in [1], objects are standalone things with well-defined boundaries and centers. Thus, we expect a box with a complete object to have a higher objectness score than a partial, oversized or background box. Bottom-up object evidence summarizes the boundary characteristic of common objects, which can help make up for the boundary discovering weakness of CNN.

We propose to integrate bottom-up object evidence to train weakly-supervised object detectors. Specifically, inspired by OICR [29], we build K instance classifiers on top of **x**, consider the output of k^{th} classifier as the supervision of $(k+1)^{th}$ one, and exploit bottom-up object evidence to guide the network training. Each classifier is implemented by a fully-connected layer and a softmax layer along $C+1$ categories (we consider background as 0^{th} class). Formally, for k^{th} classifier, we define the refinement loss function of k^{th} classifier as:

$$L^k_{ref} = -\frac{1}{|R|}\sum_{r\in R}(w^k_r \cdot CE(p^k_r, \hat{p}^k_r)), \tag{3}$$

where p^k_r denotes the $\{C+1\}$-dim output class probability of proposal r, and $\hat{p}^k_r$ indicates its ground truth one-hot label. $CE(p^k_r, \hat{p}^k_r) = -\sum_{c=0}^{C}\hat{p}^k_{rc}log(p^k_{rc})$ is a standard cross entropy function. Since the real instance-level ground truth labels are unavailable, we use an online strategy to dynamically select pseudo ground truth labels of each proposal in training loop, which will be further explained in Sec 3.4. We online assign loss weight w^k_r based on the objectness of proposal r. Specifically, we first extract bottom-up evidence of r and denote it as $O_{bu}(r)$, then integrate $O_{bu}(r)$ with $O^k_{td}(r)$, which is the class confidence produced by k^{th} classifier. w^k_r is a linear combination of bottom-up evidence and top-down confidence as follows:

$$w^k_r = \alpha O_{bu}(r) + (1-\alpha)O^k_{td}(r), \tag{4}$$

where α denotes the impact factor of bottom-up object evidence. Three terms in Eqn 4 are defined as follows:

Bottom-up object evidence O_{bu}**.** We mainly adopt **S**uperpixels **S**traddling(**SS**) as bottom-up evidence in this work, and we also explore other three evidences: textbfMulti-scale **S**aliency(**MS**), **C**olor **C**onstrast(**CC**) and **E**dge **D**ensity(**ED**). Experiment details of these evidences can be found in Sec. 4.2.

Top-down Class Confidence O_{td}**.** We compute top-down confidence of current branch based on the output of previous branch. Specifically, once we obtain class probability p^{k-1}_r of $(k-1)^{th}$ branch, top-down class confidence of k^{th} branch is computed as:

$$O^k_{td}(r) = \sum_{c=0}^{C}(p^{k-1}_{rc} \cdot \hat{p}^k_{rc}). \tag{5}$$

Since $\hat{p}^k$ is a one-hot vector, only one value of p^{k-1} will be picked to computed $O^k_{td}(r)$.

Impact Factor α**.** α is the impact factor to balance the effect of bottom-up object evidence and top-down class confidence, which is computed by some weight decay functions. Such design enables boundary knowledge to be distilled into CNN, which will be detailed discussed in Sec. 3.4.

As bottom-up object evidence and top-down class confidence can measure how likely a box contain a object from the perspective of boundary and semantic information, we consider these two representations as bottom-up and top-down objectness, respectively.

3.3. Bounding Box Regression

Bottom-up object evidence is capable to discovery object boundary, so we explore how to make it guide the pre-computed bounding boxes updated during training. An intuitive idea is to integrate bounding box regression to refine the positions and sizes of proposals.

Bounding box regression is a necessary component in typical fully-supervised object detector, as it is able to reduce localization errors. Although bounding box annotations are unavailable in weakly-supervised object detection, some existing works [9, 28, 30, 38] shows that online or offline mining pseudo ground truths and regressing them can boost the performance a lot. Inspired by this idea, we integrate a bounding box regressor on the top of **x**, and make it can be online updated. The bounding box regressor has the same formulation as in Fast R-CNN [11]. For region proposal r, the regressor predicts offsets of locations and sizes $t_r = (t^x_r, t^y_r, t^w_r, t^h_r)$, and is further optimized as follows:

$$L_{box} = \frac{1}{|R_{pos}|}\sum_{r=1}^{|R_{pos}|}(w^K_r \cdot smooth_{L1}(t_r, \hat{t}_r)), \tag{6}$$

where $\hat{t}_r$ is computed by the coordinates and sizes difference between r and $\hat{r}$ as described in [12], where $\hat{r}$ indicates the regression reference. R_{pos} indicates positive (non-background) regions, which will be explained in Sec. 3.4. $smooth_{L1}$ function is the same function as defined in [25]. w^K_r denotes the regression loss weights computed by the last classification branch. We compute pseudo regression

reference $\hat{r}$ based on the influence of w_r^K which evaluates the objectness of a proposal as we stated in Sec. 3.2:

$$\hat{r} = \underset{\{m \in M(K,R) | IoU(m,r) > T_{iou}\}}{\arg\max} (w_m^K), \tag{7}$$

where M is positive sample mining function which will be explained in Sec 3.4, and T_{iou} is a specific IoU threshold. Eqn 7 enables each positive region sample to approach a nearby box which has the high objectness.

We adopt bounding box regression to augment the box prediction during training. We update Eqn 4 as:

$$w_r^k = \alpha O_{bu}(r') + (1-\alpha) O_{td}^k(r), \tag{8}$$

where r' is r offset by t_r. We keep $O_{td}^k(r)$ unchanged because O_{td}^k contains a RoI feature warping operation, which will be affected by bounding box prediction. In this new formulation, the localization of proposals is online updated. The updated boxes may achieve higher objectness, which means more precise and complete regression targets have higher probability to be selected.

3.4. Objectness Distillation

Eqn 3 has the similar formulation as knowledge distillation [15, 16], where the external knowledge comes from bottom-up and top-down objectness. Inside which, α is a weight to balance each knowledge. At the beginning of training, top-down classifiers are not reliable enough, so we expect bottom-up evidences to take the dominant place in the combination (i.e. Eqn 4). With the guidance of bottom-up evidences, the network will try to regulate the confidence distribution of top-down classifiers to comply with bottom-up evidences. We call this process objectness distillation.

As the training proceeds, the reliability of O_{td} increases, and O_{td} inherits the boundary decision ability from O_{bu}, while it still keeps the semantic understanding ability because of the classification supervision. Therefore, α can gradually move the attention from bottom-up object evidences to top-down CNN confidences. Specifically, α is computed by some weight decay functions. We survey several weight decay functions including polynomial, cosine and constant functions, and we will compare the effectiveness of different functions in Sec 4.2.

Except α, to enable objectness distillation, we also need to determine $\hat{p}_r^k$. We want to leverage bottom-up evidences to enhance boundary representation while keep the semantic recognition ability, thus we utilize output from previous branch of classifier to mine positive proposals.

Given the output from $(k-1)^{th}$ classifier, we mine pseudo ground truths by following steps:

1. We apply Non-Maximum Suppression (NMS) on R based on class probability p_r^{k-1} of each proposal r using a pre-defined threshold T_{nms}. We denote the kept boxes as R_{keep}.

2. For each category $c (c > 0)$, if $\hat{\phi}_c = 1$, we seek all boxes from R_{keep} whose class confidences on category c are greater than another pre-defined threshold T_{conf}, and assign these boxes category label c. Specially, if no box is selected, we seek the one with highest score. The set of all seek boxes is denoted as R_{seek}.

3. For each seed box in R_{seek}, we seek all its neighbor boxes in R. Here we consider a box is the neighbor of another box if their Intersection over Union (IoU) is greater than a threshold T_{iou}. We denote the set of all neighbor boxes as $R_{neighbor}$. All neighbor boxes will be assigned the same class label as their seed boxes. Other non-seed and non-neighbor boxes will be considered as background. We transform the assigned labels to one-hot vector to obtain all $\hat{p}_r^k$.

4. Finally, we consider the union set of R_{seek} and $R_{neighbor}$ as the positive proposals: $R_{pos} = R_{seek} \cup R_{neighbor}$.

We group the above operations into function $M(k, R)$ which will return the set of positive proposals, as we mentioned in Sec 3.2 and Sec 3.3. By such way, close positive samples will be assigned same category label, while sample with high objectness will receive high weight. Such information will be distilled into CNN by optimization, thus CNN will gradually increase the ability of discovering object boundary.

3.5. Training and Inference Details

Training. The overall learning target is formulated as:

$$L = L_{base} + \lambda_1 \sum_{k=1}^{K} L_{ref}^k + \lambda_2 L_{box}, \tag{9}$$

where λ_1 and λ_2 are hyper-parameters to balance loss weights.. We adopt $\lambda_1 = 1$ and $\lambda_2 = 0.3$, and follow [29] to set $K = 3$. Since the supervision of all K classifiers comes from previous branches, we set $\alpha = 0$ in the first $2,000$ iterations for warm-up. When mining pseudo ground truths, typically we follow [38] to set $T_{nms} = 0.3, T_{conf} = 0.7, T_{iou} = 0.5$.

Inference. Our model have K refinement classifiers and one bounding box regressor. For each predicted box, we follow [29] to average the outputs from all K classifiers to produce the class confidence, and adjust its position and size using the bounding box regressor. Finally, we apply NMS with threshold 0.3 to remove redundant detected boxes.

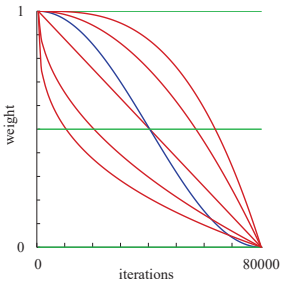
4. Experiments

4.1. Experimental Setup

Datasets and evaluation metrics. We evaluate our approach on three object detection benchmarks: PASCAL

Evidence	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
N/A	58.5	63.5	46.3	25.0	18.7	66.4	63.6	55.7	26.4	45.7	42.2	43.8	48.5	63.5	15.0	24.5	44.3	49.8	62.3	54.3	45.9
CC	62.0	64.5	44.9	24.5	19.6	70.3	62.9	52.6	20.6	54.5	44.2	49.0	55.7	64.9	15.1	22.0	49.2	56.2	52.7	58.6	47.2
ED	52.7	60.2	44.2	32.2	20.6	65.8	60.8	67.0	21.8	57.7	38.1	51.0	57.5	66.2	15.0	25.0	52.2	54.1	61.0	37.8	47.0
MS	62.0	66.2	41.2	25.1	19.2	68.1	61.5	60.7	12.2	52.9	47.9	61.6	58.8	65.6	18.1	17.6	47.2	59.0	54.3	51.4	47.5
SS	61.3	63.6	44.6	26.6	21.0	65.5	61.2	49.0	25.1	52.6	44.2	58.3	64.1	65.8	16.7	21.9	49.6	53.7	59.4	57.8	**48.1**
CC+ED+MS+SS	59.5	57.6	43.1	29.7	19.7	65.4	59.7	68.1	21.5	57.6	45.7	50.5	58.4	64.0	14.6	17.2	50.4	61.2	64.9	50.0	47.9

Table 1: Ablation experiments on bottom-up object evidences. We integrate each evidence into WSOD^2, and report the mean average precision (mAP) of PASCAL VOC 2007 test split. We also combine all evidences by simply average, the result is listed in the last row.



(a)

α decay function	γ	mAP
$\alpha = \gamma$ (green curves, from up to down $\gamma = 0, \frac{1}{2}, 1$)	0	45.9
	$1/2$	47.2
	1	48.1
$\alpha = -(\frac{n}{N})^{\gamma} + 1$ (red curves, from up to down $\gamma = 3, 2, 1, \frac{1}{2}, \frac{1}{3}$)	3	49.2
	2	49.0
	1	**50.3**
	$1/2$	46.5
	$1/3$	46.3
$\alpha = \frac{(1+cos(\frac{n\pi}{N}))}{2}$ (blue curve)	-	49.7

(b)

Figure 3: Ablation study of weight decay functions for α. (a) Weight decay curves of different function. (b)mAP of different decay setting on PASCAL VOC 2007 test split. n and N indicate current step and total step number, respectively. [Best viewed in color]

VOC 2007 & 2012 [7] and MS COCO [22]. After removing the bounding box annotations provided by these datasets, we only use images and their label information for training. PASCAL VOC 2007 and 2012 consists of $9,962$ and $22,531$ images of 20 categories, respectively. For PASCAL VOC, we train on *trainval* split ($5,011$ images for 2007 and $11,540$ for 2012), report mean average precision (mAP) on *test* split, and also adopt correct localization (CorLoc) on *trainval* split to measure the localization accuracy. Both two metrics are performed under the condition of $IoU > 0.5$ as a standard setting. MS COCO contains 80 categories. We train on *train2014* split and evaluate on *val2014* split, which consists of $82,783$ and $40,504$ images, respectively. We report $AP@.50$ and $AP@[.50:.05:.95]$ on *val2014*.

Implementation details. We adopt VGG16 [26] as the CNN backbone, and use parameters pre-trained on ImageNet [19] for initialization. We randomly initialize the weights of all new layers using Gaussian distributions with 0-mean and standard deviations 0.01 (except 0.001 for bounding box regressor), and initialize all new biases to 0. We follow a widely-used setting [2, 29, 30, 37] to use Selective Search [33] to generate about $2,000$ proposals for each image. The whole network is end-to-end optimized using SGD with an initial learning rate of 10^{-3}, weight decay of 0.0005 and momentum of 0.9. The overall iteration step number is set to $80,000$ on VOC 2007, and the learning rate will be divided by 10 at $40,000^{th}$ step. For VOC 2012 we double the iteration step number and learning rate decay step is also doubled to $80,000^{th}$ step. For MS COCO we set iteration step number to $360,000$, and make learning rate decay at $180,000^{th}$ step. We follow [28, 29] to adopt multi-scale settings in training. Specifically, the short edge of the input image will be randomly re-scaled to a scale in $\{480, 576, 588, 864, 1280\}$, and we restrict the length of the long edge not greater than 2000. Besides, horizontal flip of all training images will be also used for training. We report single-scale testing results for ablation study, and report multi-scale testing results when comparing with previous works. All our experiments are implemented based on PyTorch on 4 NVIDIA P100 GPUs.

bbox	NMS	BU	α decay	mAP
				43.3
✓				45.1
✓	✓			45.9
✓	✓	✓		48.1
✓	✓	✓	✓	50.3

Table 2: Ablation study of different components of WSOD^2. ✓ indicates that the component is used. "NMS" is unchecked when proposal with highest confidence for each category is used as seed box.

4.2. Ablation Study

We conduct ablation studies to demonstrate the effectiveness of WSOD^2 on PASCAL VOC 2007.

Bottom-up evidences. For bottom-up object evidence, we test the effect of four evidences in both individual and combined ways. The four evidences are list as follows:

1) **M**ulti-scale **S**aliency(**MS**) which summarizes the saliency over several scales;

2) **C**olor **C**onstrast(**CC**) which computes the color distribution difference with immediate surrounding area;

3) **E**dge **D**ensity(**ED**) which computes the density of edges in the inner rings;

4) **S**uperpixels **S**traddling(**SS**) which analyzes the straddling of all superpixels.

Since the value ranges of different evidences are inconsist, we normalize the computed value to $[0-1]$. For **CC**, **ED** and **MS**, we fix their parameters by setting $\theta^{MS} = 0.2, \theta^{CC} = 2, \theta^{ED} = 2$ empirically due to the lack of supervision. For **SS**, we follow [8] to set $\theta_{\sigma}^{SS} = 0.8, \theta_{k}^{SS} = 300$. We refer the readers to [1] for more details of these four evidences and the meaning of $\theta^{MS}, \theta^{CC}, \theta^{ED}, \theta^{SS}$.

To easier analyze the effect of these bottom-up evidences, we simply keep $\alpha = 1$ in this ablation experiment for all settings that include these evidences, and $\alpha = 0$ for the method that does not involve any bottom-up evidence as a baseline for comparison. We also test the combination of these four evidences by their average. As discussed in [1],

method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
WSDDN [2]	39.4	50.1	31.5	16.3	12.6	64.5	42.8	42.6	10.1	35.7	24.9	38.2	34.4	55.6	9.4	14.7	30.2	40.7	54.7	46.9	34.8
ContextLocNet [18]	57.1	52.0	31.5	7.6	11.5	55.0	53.1	34.1	1.7	33.1	49.2	42.0	47.3	56.6	15.3	12.8	24.8	48.9	44.4	47.8	36.3
OICR [29]	58.0	62.4	31.1	19.4	13.0	65.1	62.2	28.4	24.8	44.7	30.6	25.3	37.8	65.5	15.7	24.1	41.7	46.9	64.3	62.6	41.2
PCL [28]	54.4	69.0	39.3	19.2	15.7	62.9	64.4	30.0	25.1	52.5	44.4	19.6	39.3	67.7	17.8	22.9	46.6	57.5	58.6	63.0	43.5
Tang *et al.* [30]	57.9	**70.5**	37.8	5.7	21.0	66.1	**69.2**	59.4	3.4	57.1	**57.3**	35.2	64.2	68.6	**32.8**	**28.6**	50.8	49.5	41.1	30.0	45.3
C-WSL [9]	62.9	64.8	39.8	28.1	16.4	69.5	68.2	47.0	27.9	55.8	43.7	31.2	43.8	65.0	10.9	26.1	52.7	55.3	60.2	**66.6**	46.8
MELM [34]	55.6	66.9	34.2	29.1	16.4	68.8	68.1	43.0	25.0	**65.6**	45.3	53.2	49.6	68.6	2.0	25.4	52.5	56.8	62.1	57.1	47.3
ZLDN [37]	55.4	68.5	50.1	16.8	20.8	62.7	66.8	56.5	2.1	57.8	47.5	40.1	69.7	68.2	21.6	27.2	53.4	56.1	52.5	58.2	47.6
WSCDN [35]	61.2	66.6	48.3	26.0	15.8	66.5	65.4	53.9	24.7	61.2	46.2	53.5	48.5	66.1	12.1	22.0	49.2	53.2	**66.2**	59.4	48.3
WSOD^2 (ours)	**65.1**	64.8	**57.2**	**39.2**	**24.3**	**69.8**	66.2	**61.0**	**29.8**	64.6	42.5	**60.1**	**71.2**	**70.7**	21.9	28.1	**58.6**	**59.7**	52.2	64.8	**53.6**
WSOD^{2*} (ours)	68.2	70.7	61.5	42.3	28.0	73.4	69.3	52.3	32.7	71.9	42.8	57.9	73.8	71.4	25.5	29.2	61.6	60.9	56.5	70.7	56.0

Table 3: Mean average precision for different methods on PASCAL VOC 2007 test split. $*$ means training on 07+12 *trainval* splits.

method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	CorLoc
WSDDN [2]	65.1	58.8	58.5	33.1	39.8	68.3	60.2	59.6	34.8	64.5	30.5	43.0	56.8	82.4	25.5	41.6	61.5	55.9	65.9	63.7	53.5
ContextLocNet [18]	83.3	68.6	54.7	23.4	18.3	73.6	74.1	54.1	8.6	65.1	47.1	59.5	67.0	83.5	35.3	39.9	67.0	49.7	63.5	65.2	55.1
OICR [29]	81.7	80.4	48.7	49.5	32.8	81.7	85.4	40.1	40.6	79.5	35.7	33.7	60.5	88.8	21.8	57.9	76.3	59.9	75.3	81.4	60.6
ZLDN [37]	74.0	77.8	65.2	37.0	**46.7**	75.8	83.7	58.8	17.5	73.1	49.0	51.3	76.7	87.4	30.6	47.8	75.0	62.5	64.8	68.8	61.2
PCL [28]	79.6	**85.5**	62.2	47.9	37.0	83.8	83.4	43.0	38.3	80.1	50.6	30.9	57.8	90.8	27.0	58.2	75.3	**68.5**	**75.7**	78.9	62.7
C-WSL [9]	85.8	81.2	64.9	50.5	32.1	**84.3**	85.9	54.7	43.4	80.1	42.2	42.6	60.5	90.4	13.7	57.5	**82.5**	61.8	74.1	**82.4**	63.5
Tang *et al.* [30]	77.5	81.2	55.3	19.7	44.3	80.2	**86.6**	69.5	10.1	87.7	**68.4**	52.1	84.4	**91.6**	**57.4**	**63.4**	77.3	58.1	57.0	53.8	63.8
WSCDN [35]	85.8	80.4	73.0	42.6	36.6	79.7	82.8	66.0	34.1	78.1	36.9	68.6	72.4	**91.6**	22.2	51.3	79.4	63.7	74.5	74.6	64.7
WSOD^2 (ours)	**87.1**	80.0	**74.8**	**60.1**	36.6	79.2	83.8	**70.6**	**43.5**	**88.4**	46.0	**74.7**	**87.4**	90.8	44.2	52.4	81.4	61.8	67.7	79.9	**69.5**
WSOD^{2*} (ours)	89.6	82.4	79.9	63.3	40.1	82.7	85.0	62.8	45.8	89.7	52.1	70.9	88.8	91.6	37.0	56.4	85.6	64.3	74.1	85.3	71.4

Table 4: Correct Localization for different methods on PASCAL VOC 2007 trainval split. $*$ means training on 07+12 *trainval* splits.

method	mAP	CorLoc
OICR [29]	37.9	62.1
PCL [28]	40.6	63.2
Tang *et al.* [30]	40.8	64.9
ZLDN [37]	42.9	61.5
WSCDN [35]	43.3	65.2
WSOD^2 (ours)	**47.2** [1]	**71.9**
WSOD^{2*} (ours)	52.7 [2]	72.2

Table 5: Comparisons with different methods on PASCAL VOC 2012 dataset. $*$ means training on 07+12 *trainval* splits.

method	AP@.50	AP@[.50:.05:.95]
Ge *et al.* [10]	19.3	8.9
PCL [28]	19.4	8.5
PCL + Fast R-CNN [28]	19.6	9.2
WSOD^2 (ours)	**22.7**	**10.8**

Table 6: Experiment results of different methods on MS COCO dataset.

linear combination is not a good way to combine them, we conduct this experiment only for evaluating the effectiveness of bottom-up evidences and inspiring future works.

The results are shown in Table 1. From the comparison with the baseline, we can find that the performance can increase significantly with the guidance of bottom-up evidences. Table 1 also includes AP on all categories, from which we find that different evidences may favor different categories. For example, for single evidence, **ED** favors to "boat", while not performs good on "tv'. Moreover, we can find that this result also agrees with the performance that measures objectness of each evidence as reported in [1], which indicates that these bottom-up evidences are positive correlated to object detection performance. From the result of their combination, we can find it achieves better performance than all single evidences except **SS**. We believe that linear average is not a correct way to combine these evidence, and better ways can be explored in the future. We adopt **SS** as bottom-up object evidence in later experiments.

Impact factor α. We test several weight decay functions, including constant ($\alpha = 0, 0.5, 1$), polynomial($\alpha = -(n/N)^\gamma + 1$, where $\gamma = 2, 3, 1, 1/2, 1/3$) and cosine ($\alpha = (1 + cos(n\pi/N)/2)$ functions where n and N indicate current step and total step number, respectively. The results are shown in Figure 3. From the comparison of the first three lines, we find that bottom-up evidences will help the model learn the boundary representation and results in better object detection result. Among different designs, linear decay (i.e., $\alpha = -(n/N) + 1$) performs best and the later experiments are conducted based on this setting. We remain exploration of the best parameters for future study.

Effect of each component. Table 2 shows the effectiveness of each component. We can find that the bounding box regressor brings at least 2.6 mAP improvement. Settings that do not use NMS means directly consider the highest-confident box for each category as seed box as OICR [29]. NMS can also improve 0.8 mAP. Details of bottum-up evidences (BU) and α decay function are discussed above, where both bottom-up evidences and α decay function can bring 2.2 mAP improvement.

4.3. Comparisons with State-of-the-Arts

We evaluate WSOD^2 on PASCAL VOC 2007 & 2012 [7] and MS COCO datasets [22], report the performances and compare with state-of-the-art weakly-supervised detectors. As most of our compared approaches adopt multi-scale testing, we report our multi-scale testing results.

AP evaluation on PASCAL VOC. From Table 3 we can find that WSOD^2 achieves 53.6 mAP on PASCAL VOC 2007, which significantly outperforms other end-to-end trainable models [28, 29, 35] with at least 5.3 mAP. WSOD^2 is also robust on PASCAL VOC 2012 and achieves

[1] http://host.robots.ox.ac.uk:8080/anonymous/AVFPZC.html
[2] http://host.robots.ox.ac.uk:8080/anonymous/Z4VIWW.html

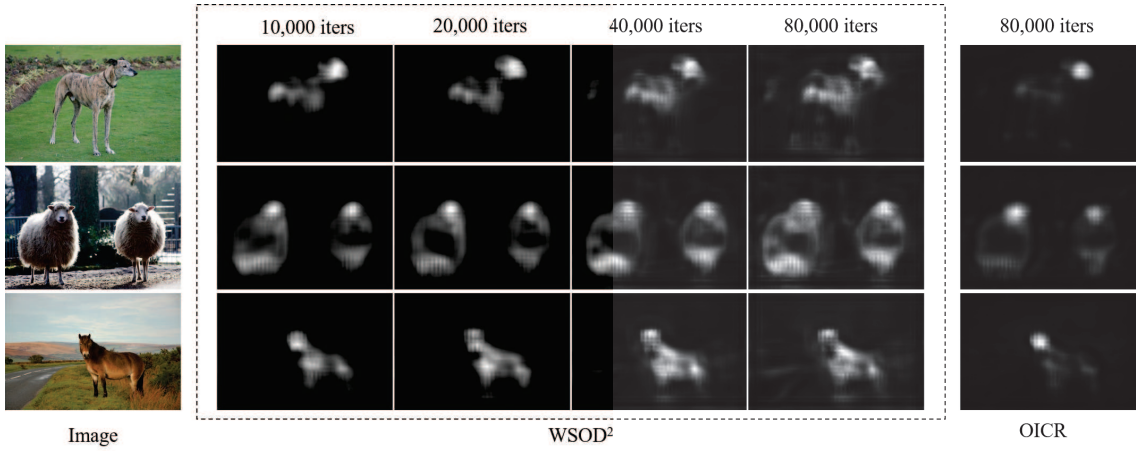


Figure 4: Visualization of *conv5* feature maps. The response maps are generated by average along all feature map channels, and normalized to $(0, 255)$. The feature maps in middle 4 columns are extracted by WSOD^2 at different iterations. The last column is feature maps we extracted by OICR [29].

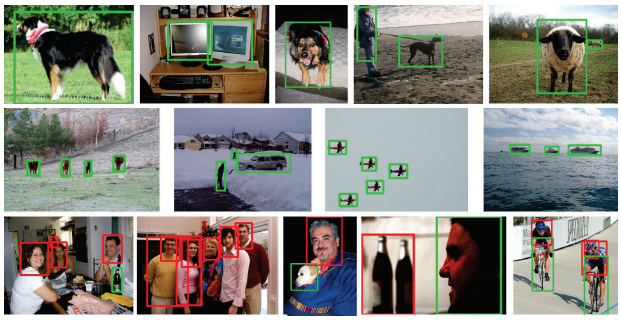


Figure 5: Example results by WSOD^2. Green boxes indicate corrected predictions, and red ones indicate the failure cases. [Best viewed in color]

47.2 mAP, which is shown in Table 5.

Besides, we follow the common setting in fully-supervised object detection to train WSOD^2 on PASCAL VOC 07+12 trainval splits, and denote it as WSOD^2+. Such setting achieves a surprising mAP score 56.1 as shown in the last row of Table3.

CorLoc evaluation on PASCAL VOC. CorLoc evaluates the localization accuracy of detectors on training set. We report results on PASCAL VOC 2007 and 2012 *trainval* split in Table 4 and Table 5, respectively. We can find that WSOD^2 significantly surpasses outperforms other end-to-end trainable models [28, 29, 35] on both PASCAL VOC 2007 and 2012.

AP evaluation on MS COCO. We report results on MS COCO dataset in Table 6. Since few works report results on MS COCO dataset, we only compare performance with [10] and [28]. We can find that WSOD^2 outperforms compared works by at least 2 AP.

4.4. Visualization and Case Study

We make a qualitative analysis of the effectiveness of WSOD^2 compared with OICR. We extract the *conv5* features of trained models, and visualize some cases in Figure 4. The highlighted parts indicate the high response area of the input image in CNN. Compared with OICR, WSOD^2 can gradually transfer the response area from discriminate parts to complete objects.

Figure 5 exhibits some successful and failure cases of WSOD^2. We obverse that WSOD^2 can well handle multiple discrete instances, while there still remains a challenge to solve detection problem in dense scenarios. We also find that for "person" class, most weakly-supervised object detectors tend to find human faces. The reason is that in the current datasets, human face is the most common pattern of "person", while other parts are often missed in the image. This remains a challenging problem and we can consider leveraging human structure prior in the future.

5. Conclusion

In this paper, we propose a novel weakly-supervised object detection with bottom-up and top-down objectness distillation (i.e., WSOD^2) to improve the deep objectness representation of CNN. Bottom-up object evidence, which could measures the probability of a bounding box including a complete object, is utilized to distill boundary features in CNN in an adaptive training way. We also propose a training strategy that integrates bounding box regression and progressive instance classifier in an end-to-end fashion. We conduct experiments on some standard datasets and settings for WSOD task with our approach. Results demonstrate the effectiveness of our proposed WSOD^2 in both quantitative and qualitative way. We also make a thorough analysis on the challenges and possible improvement (e.g., for "person" class) of WSOD problem.

6. Acknowledgments

This work is partially supported by NSF of China under Grant 61672548, U1611461, 61173081, and the Guangzhou Science and Technology Program, China, under Grant 201510010165.

References

[1] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. What is an object? In *CVPR*, pages 73–80, 2010.

[2] Hakan Bilen and Andrea Vedaldi. Weakly supervised deep detection networks. In *CVPR*, pages 2846–2854, 2016.

[3] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *NIPS*, pages 379–387, 2016.

[4] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, pages 764–773, 2017.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.

[6] Ali Diba, Vivek Sharma, Ali Pazandeh, Hamed Pirsiavash, and Luc Van Gool. Weakly supervised cascaded convolutional networks. In *CVPR*, pages 914–922, 2017.

[7] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010.

[8] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 59(2):167–181, 2004.

[9] Mingfei Gao, Ang Li, Ruichi Yu, Vlad I Morariu, and Larry S Davis. C-wsl: Count-guided weakly supervised localization. In *ECCV*, pages 152–168, 2018.

[10] 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 *CVPR*, pages 1277–1286, 2018.

[11] Ross Girshick. Fast R-cnn. In *CVPR*, pages 1440–1448, 2015.

[12] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014.

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-cnn. In *ICCV*, pages 2961–2969, 2017.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[15] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of tricks for image classification with convolutional neural networks. In *CVPR*, pages 558–567, 2019.

[16] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *Computer Science*, 14(7):38–39, 2015.

[17] Zequn Jie, Yunchao Wei, Xiaojie Jin, Jiashi Feng, and Wei Liu. Deep self-taught learning for weakly supervised object localization. In *CVPR*, pages 1377–1385, 2017.

[18] Vadim Kantorov, Maxime Oquab, Minsu Cho, and Ivan Laptev. Contextlocnet: Context-aware deep network models for weakly supervised localization. In *ECCV*, pages 350–365, 2016.

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

[20] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv:1811.00982*, 2018.

[21] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 2117–2125, 2017.

[22] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755, 2014.

[23] Tie Liu, Zejian Yuan, Jian Sun, Jingdong Wang, Nanning Zheng, Xiaoou Tang, and Heung-Yeung Shum. Learning to detect a salient object. *TPAMI*, 33(2):353–367, 2011.

[24] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, pages 21–37, 2016.

[25] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015.

[26] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. 2015.

[27] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015.

[28] Peng Tang, Xinggang Wang, Song Bai, Wei Shen, Xiang Bai, Wenyu Liu, and Alan Loddon Yuille. Pcl: Proposal cluster learning for weakly supervised object detection. *TPAMI*, 2018.

[29] Peng Tang, Xinggang Wang, Xiang Bai, and Wenyu Liu. Multiple instance detection network with online instance classifier refinement. In *CVPR*, pages 2843–2851, 2017.

[30] Peng Tang, Xinggang Wang, Angtian Wang, Yongluan Yan, Wenyu Liu, Junzhou Huang, and Alan Yuille. Weakly supervised region proposal network and object detection. In *ECCV*, pages 352–368, 2018.

[31] Eu Wern Teh, Mrigank Rochan, and Yang Wang. Attention networks for weakly supervised object localization. In *BMVC*, pages 1–11, 2016.

[32] Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification: An overview. *IJDWM*, 3(3):1–13, 2007.

[33] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *IJCV*, 104(2):154–171, 2013.

[34] Fang Wan, Pengxu Wei, Jianbin Jiao, Zhenjun Han, and Qixiang Ye. Min-entropy latent model for weakly supervised object detection. In *CVPR*, pages 1297–1306, 2018.

[35] Jiajie Wang, Jiangchao Yao, Ya Zhang, and Rui Zhang. Collaborative learning for weakly supervised object detection. 2018.

[36] Yunchao Wei, Zhiqiang Shen, Bowen Cheng, Honghui Shi, Jinjun Xiong, Jiashi Feng, and Thomas Huang. Ts2c: tight box mining with surrounding segmentation context for weakly supervised object detection. In *ECCV*, pages 454–470, 2018.

[37] Xiaopeng Zhang, Jiashi Feng, Hongkai Xiong, and Qi Tian. Zigzag learning for weakly supervised object detection. In *CVPR*, pages 4262–4270, 2018.

[38] Yongqiang Zhang, Yancheng Bai, Mingli Ding, Yongqiang Li, and Bernard Ghanem. W2f: A weakly-supervised to fully-supervised framework for object detection. In *CVPR*, pages 928–936, 2018.

[39] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, pages 2921–2929, 2016.

[40] Yi Zhu, Yanzhao Zhou, Qixiang Ye, Qiang Qiu, and Jianbin Jiao. Soft proposal networks for weakly supervised object localization. In *ICCV*, pages 1841–1850, 2017.