Instance-Aware, Context-Focused, and Memory-Efficient Weakly Supervised Object Detection

Zhongzheng Ren[1,2*] Zhiding Yu[2] Xiaodong Yang[2*] Ming-Yu Liu[2]
Yong Jae Lee[3] Alexander G. Schwing[1] Jan Kautz[2]
[1]University of Illinois at Urbana-Champaign [2]NVIDIA [3]University of California, Davis

[1]NVIDIA [2]UIUC [3]UC Davis

Abstract

Weakly supervised learning has emerged as a compelling tool for object detection by reducing the need for strong supervision during training. However, major challenges remain: (1) differentiation of object instances can be ambiguous; (2) detectors tend to focus on discriminative parts rather than entire objects; (3) without ground truth, object proposals have to be redundant for high recalls, causing significant memory consumption. Addressing these challenges is difficult, as it often requires to eliminate uncertainties and trivial solutions. To target these issues we develop an instance-aware and context-focused unified framework. It employs an instance-aware self-training algorithm and a learnable Concrete DropBlock while devising a memory-efficient sequential batch back-propagation. Our proposed method achieves state-of-the-art results on COCO (12.1% AP, 24.8% AP_{50}), VOC 2007 (54.9% AP), and VOC 2012 (52.1% AP), improving baselines by great margins. In addition, the proposed method is the first to benchmark ResNet based models and weakly supervised video object detection. Refer to our project page for code, models, and more details: https://github.com/NVlabs/wetectron. The full paper is available at https://arxiv.org/pdf/2004.04725.pdf.

1. Introduction

Recent works on object detection [9, 8] have achieved impressive results. However, the training process often requires strong supervision in terms of precise bounding boxes. Obtaining such annotations at a large scale can be costly, time-consuming, or even infeasible. This motivates weakly supervised object detection (WSOD) methods [2, 10] where detectors are trained with weaker forms of supervision such as image-level category labels. These works typically formulate WSOD as a multiple instance learning task, treating the set of object proposals in each image as a bag. The selection of proposals that truly cover objects is modeled using learnable latent variables.

While alleviating the need for precise annotations, existing weakly supervised object detection methods [2, 10, 3] often face three major challenges due to the underdetermined and ill-posed nature:

(1) Instance Ambiguity. This arguably biggest challenge subsumes two common types of issues: (a) *Missing Instances:* Less salient objects in the background with rare poses and smaller scales are often ignored. (b) *Grouped Instances:* Multiple instances of the same category are grouped into a single bounding box when spatially adjacent. Both issues are caused by bigger or more salient boxes receiving higher scores than smaller or less salient ones.

(2) Part Domination. Predictions tend to be dominated by the most discriminative parts of an object. This issue is particularly pronounced for classes with big intra-class difference. For example, on classes such as animals and people, the model often turns into a 'face detector' as faces are the most consistent appearance signal.

(3) Memory Consumption. Existing proposal generation methods often produce redundant number of proposals. Without ground-truth localization, maintaining the proposal number is necessary to achieve a reasonable recall rate and good performance. This often causes huge memory consumption, especially for video object detection. Due to the large number of proposals, most memory is consumed in the intermediate layers after ROI-Pooling.

To address the above three challenges, we propose a unified weakly supervised learning framework that is instance-aware and context-focused. The proposed method tackles **Instance Ambiguity** by introducing an advanced self-training algorithm where instance-level pseudo ground-truth, in forms of category labels and regression targets are computed by considering more instance-associative spatial diversification constraints (Sec. 3.1). We address **Part Domination** by introducing a parametric spatial dropout termed 'Concrete DropBlock.' This module is learned end-to-end to maximize the detection objective and thus encourages the whole framework to consider context rather than focusing on the most discriminative parts as detailed

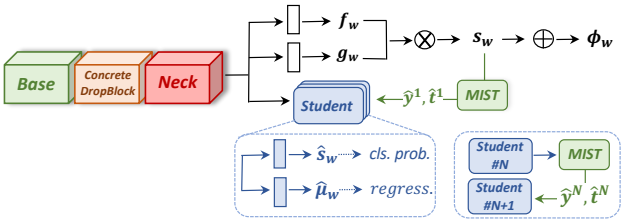


Figure 1: The overall framework. ROI-Pooling and the operations in Eq. (1) are abstracted away for simplicity.

in Sec. 3.2. Finally, to ease **Memory Consumption**, our method adopts a sequential batch back-propagation algorithm which batch-processes data at the most memory-heavy stage. This permits the assessment of larger deep models such as ResNet [5] for WSOD, and the exploration of weakly supervised video object detection. More details are discussed in Sec. 3.3.

Tackling the aforementioned three challenges via our proposed framework leads to state-of-the-art performance on several popular datasets, including COCO, VOC 2007 and 2012. The effectiveness and robustness of each proposed module are demonstrated by detailed ablation studies, and further verified through qualitative results.

2. Background

Bilen and Vedaldi [2] are among the first to develop an end-to-end deep WSOD framework based on the idea of multiple instance learning. Specifically, given an input image $i \in I$ and the corresponding set of pre-computed [12] proposals R, an ImageNet pre-trained neural network is used to produce classification logits $f_w(c,r) \in \mathbb{R}$ and detection logits $g_w(c,r) \in \mathbb{R}$ for every object category $c \in C$ and for every region $r \in R$. The vector w subsumes all trainable parameters. Two score matrices: $s(c|r)$ of a region r being classified as category c, and $s(r|c)$ of detecting region r for category c are obtained through

$$s_w(c|r) = \frac{\exp f_w(c,r)}{\sum_{c \in C} \exp f_w(c,r)}, \text{ and } s_w(r|c) = \frac{\exp g_w(c,r)}{\sum_{r \in R} \exp g_w(c,r)}. \tag{1}$$

The final score $s_w(c,r)$ for assigning category c to region r is computed via an element-wise product: $s_w(c,r) = s_w(c|r)s_w(r|c) \in [0,1]$. During training, $s_w(c,r)$ is summed for all regions $r \in R$ to obtain the image evidence $\phi_w(c) = \sum_{r \in R} s_w(c,r)$. The loss is then computed via:

$$\mathcal{L}_{\text{img}}(i;w) = -\sum_{c \in C} y(c) \log \phi_w(c), \tag{2}$$

where $y(c) \in \{0,1\}$ is the ground truth (GT) class label indicating image-level existence of category c. For inference, $s_w(c,r)$ is used for prediction followed by standard non-maximum suppression (NMS) and thresholding.

To integrate online self-training, [3, 13] use region score $s_w(c,r)$ as teacher to generate instance-level pseudo category label $\hat{y}(c,r) \in \{0,1\}$ for every region $r \in R$. This is done by treating the top-scoring region and its highly-overlapped neighbors as the positive examples for class c. The extra student layer is then trained for region classification via:

$$\mathcal{L}_{\text{roi}}(i;w) = -\frac{1}{|R|} \sum_{c \in C} \hat{y}(c,r) \log \hat{s}_w(c|r), \tag{3}$$

where $\hat{s}_w(c|r)$ is the output of this layer. During testing, the student prediction $\hat{s}_w(c|r)$ will be used rather than $s_w(c,r)$. We build upon this formulation and develop two additional novel modules as described subsequently.

Algorithm 1 Multiple Instance Self-Training

Input: Image I, class label y, proposals R, threshold τ, percentage p
Output: Pseudo boxes $\hat{R}^1$

```
Feed I into model; get ROI scores s
for ground-truth class c do
    R(c)_sorted ← SORT(s(c,*))    //sort ROIs by scores of class c
    R'(c) ← top p percent of R(c)_sorted
    R̂(c) ← r'_0                   // save first region (top-scoring) r'_0 ∈ R'
    for i in {2 ... |R'(c)|} do    // start from the second highest
        APPEND(R̂(c), r'_i) if IoU(r'_i, r̂_j) < τ, ∀ r̂_j ∈ R̂(c)
return R̂(c)
```

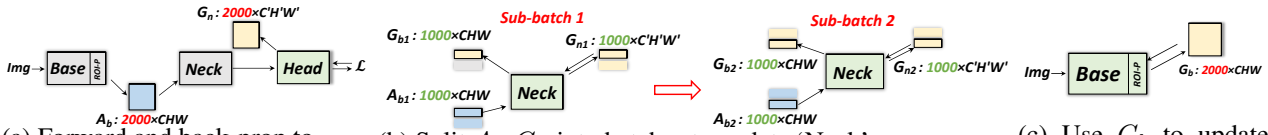
3. Approach

For aforementioned three challenges, we develop an instance-aware and context-focused framework outlined in Fig. 1. It contains a new self-training algorithm to reduce instance ambiguity. It reduces part-domination via 'Concrete DropBlock', and is more memory friendly.

3.1. Multiple instance self-training (MIST)

We propose a self-training novel algorithm to generate diverse yet representative pseudo boxes as detailed in Alg. 1. Specifically, we first sort all the scores across the set R for each class c that appears in the category-label. We then pick the top p percent of the ranked regions to form an initial candidate pool $R'(c)$. Note that the size of the candidate pool $R'(c)$, *i.e.*, $|R'(c)|$ is image-adaptive and content-dependent by being proportional to $|R|$. Intuitively, $|R|$ is a meaningful prior for the overall objectness of an input image. A diverse set of high-scoring non-overlapping regions are then picked from $R'(c)$ as the pseudo boxes $\hat{R}(c)$ using non-maximum suppression. Even though being simple, this effective algorithm leads to significant performance improvements as shown in Sec. 4.

Self-Training with Regression. Bounding box regression is another module that plays an important role in supervised object detection but is missing in online self-training methods. To close the gap, we encapsulate a classification layer and a regression layer into 'student blocks' as shown via blue boxes in Fig. 1. We jointly optimize them using



(a) Forward and back-prop to update 'Head'. A_b, G_n saved.

(b) Split A_b, G_n into batches to update 'Neck'. G_b accumulated.

(c) Use G_b to update 'Base' network.

Figure 5: Seq-BBP: blue, yellow, and green blobs represent activation, gradients, and the module that is being updated.

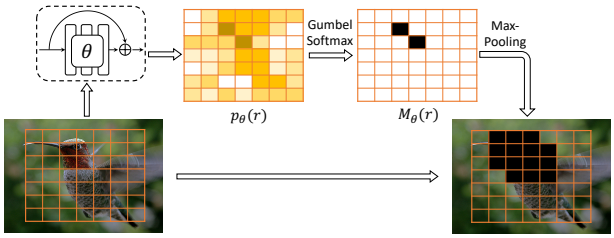


Figure 6: Illustration of the Concrete DropBlock.

pseudo-labels $\hat{R}$. The predicted bounding boxes from the regression layer are referred to via $\mu_w(r)$ for all regions $r \in R$. For each region r, if it is highly overlapping with a pseudo-box $\hat{r} \in \hat{R}$ for ground-truth class c, we generate the regression target $\hat{t}(r)$ by using the coordinates of $\hat{r}$ and by marking the classification label $\hat{y}(c, r) = 1$. The complete region-level loss for training the student block is:

$$\mathcal{L}_{\text{roi}}(i; w) = \frac{1}{|R|} \sum_{r \in R} \lambda_r (\mathcal{L}_{\text{smooth-L1}}(\hat{t}(r), \mu_w(r)) - \frac{1}{|C|} \sum_{c \in C} \hat{y}(c, r) \log \hat{s}_w(c|r)), \quad (4)$$

where $\mathcal{L}_{\text{smooth-L1}}$ is the Smooth-L1 objective used in [9] and λ_r is a per-region weighting scalar used in [11].

3.2. Concrete DropBlock

Because of the intra-category variation, existing WSOD methods often mistakenly only detect the discriminative parts of an object rather than its full extent. A natural solution for this issue encourages the network to focus on the context which can be achieved by dropping the most discriminative parts. Hence, spatial dropout is an intuitive fit.

For object detection, naïve spatial dropout is limited since the discriminative parts of objects differ in location and size. Therefore, a more structured DropBlock [4] is proposed. Specifically, during training, spatial points on feature maps are sampled randomly as blob centers. Then the square regions around these centers of size $H \times H$ are zeroed out across all channels on the feature map. Finally, the feature values are re-scaled by a factor of the area of the feature map over the area of the un-dropped region, so that no normalization has to be applied for inference where all features are kept.

DropBlock is essentially a non-parametric regularization technique. But it doesn't satisfactorily address the 'part domination' issue: ideally, discriminative parts should be dropped more frequently than other areas. To fix this, we develop *Concrete DropBlock*: a data-driven and parametric variant of DropBlock which is learned end-to-end to drop the most relevant regions as shown in Fig. 6. Given an input image, the feature maps $\psi_w(r) \in \mathbb{R}^{H \times H}$ are computed for each region $r \in R$ using the layers up until ROI-Pooling. H is the ROI-Pooling output dimension. We then feed $\psi_w(r)$ into a convolutional residual block to generate a probability map $p_\theta(r) \in \mathbb{R}^{H \times H}$ $\forall r \in R$ where θ subsumes the trainable parameters of this module. Each element of $p_\theta(r)$ is regarded as an independent Bernoulli variable, and this probability map is transformed via a spatial Gumbel-Softmax [6, 7] into a hard mask $M_\theta(r) \in \{0, 1\}^{H \times H}$ $\forall r \in R$. This operation is a differentiable approximation of probabilistic sampling. To avoid trivial solutions (*e.g.*, everything will be dropped or a certain area is dropped consistently), we apply a threshold τ such that $p_\theta(r) = \min(p_\theta(r), \tau)$ guarantees that the computed $M_\theta(r)$ is sparse. To finally generate the structured mask and to normalize the features we follow DropBlock.

Importantly, during training, we jointly optimize the original network parameters w and the residual block parameters θ through a minmax objective:

$$w^*, \theta^* = \arg\min_w \max_\theta \sum_{i \in I} \mathcal{L}_{\text{img}}(i; w, \theta) + \mathcal{L}_{\text{roi}}(i; w, \theta). \quad (5)$$

By maximizing the original loss w.r.t. the Concrete DropBlock parameters, the Concrete DropBlock will learn to drop the most discriminative parts of the objects, as it is the easiest way to increase the training loss. This forces the object detector to also look at the context regions. We found this strategy to improve performance especially for non-rigid object categories, which usually have a large intra-class difference.

3.3. Sequential batch back-propagation

In this section, we discuss how to handle memory limitations for WSOD models. Vanilla back-propagation is computationally efficient yet memory-demanding. It becomes even worse for WSOD models as the activation (after ROI-Pooling) grow from $1 \times CHW$ (image feature) to $N \times CHW$ (ROI-features) where N is in the thousands. To address it, we propose a sequential computation in the 'Neck' sub-module as depicted in Fig. 5. During forward, the input is first passed through the 'Base' and 'Neck,' with

Name	Val-AP	Val-AP$_{50}$	Test-AP	Test-AP$_{50}$
Faster R-CNN	21.2	41.5	21.5	42.1
WSDDN [2]	-	-	-	11.5
PCL [10]	8.5	19.4		
WSOD2 [13]	10.8	22.7	-	-
OICR [11]+Ens+FRCNN	7.7	17.4	-	-
PCL [10]+Ens.+FRCNN	9.2	19.6	-	-
Ours (single-model)	**11.4**	**24.3**	**12.1**	**24.8**

Table 1: Single model results (VGG16) on COCO.

Model	Proposal	Backbone	AP	AP$_{50}$
Faster R-CNN	RPN	R101-C4	27.2	48.4
Ours	MCG	VGG16	**11.4**	**24.3**
Ours	MCG	R50-C4	**12.6**	**26.1**
Ours	MCG	R101-C4	**13.0**	**26.3**

Table 2: Single model results (ResNet) on COCO 2014 val.

only the activation A_b after the 'Base' stored. The output of the 'Neck' then goes into the 'Head' for its first forward and backward pass to update the weights of the 'Head' and the gradients G_n as shown in Fig. 5 (a). To update the 'Neck', we split the ROI-features into batches and run BP on small batches sequentially. Hence we avoid storing memory-consuming feature maps and their gradients within the 'Neck' as shown in Fig. 5 (b). The gradient G_b is accumulated to update the parameters of the 'Base' via regular back-propagation as illustrated in Fig. 5 (c).

4. Experiments

We conduct experiments on COCO and PASCAL VOC, and report the standard metrics.

4.1. Overall performance

VGG16-COCO. We compare to several recent WSOD methods on COCO in Tab. 1. Our single model without any post-processing outperforms all previous approaches (w/ bells and whistles) by a great margin.

ResNet-COCO. ResNet models have never been used for WSOD due to the large memory consumption. With the training techniques introduced in Sec. 3.3, we provide the first benchmark for COCO using ResNet-50 and ResNet-101 in Tab. 2.

VGG16-VOC. To fairly compare with most previous WSOD works, we also evaluate on VOC, and report the results in Tab. 3. All entries in this table are single model results. Our single-model results surpass all previous approaches on the publicly available 2007 test set and private 2012 test set. In addition, our single model also performs better than all previous methods with bells and whistles (*e.g.*, '+FRCNN': supervised re-training, '+Ens.': model ensemble).

*http://host.robots.ox.ac.uk:8080/anonymous/DCJ5GA.html

Methods	Proposal	07-AP$_{50}$	12-AP$_{50}$
Faster R-CNN	RPN	**69.9**	**67.0**
PCL [10]	SS	43.5	40.6
WSOD2 [13]	SS	**53.6**	47.2
C-MIDN [3]	SS	52.6	**50.2**
Pred Net [1]+Ens.+FRCNN	SS	53.6	49.5
C-MIDN [3]+FRCNN	SS	53.6	50.3
Ours (single)	SS	**54.9**	**52.1***

Table 3: Single model (VGG16) detection results on VOC.

5. Conclusion

In this paper, we address three major issues of WSOD. For each we have proposed a solution and demonstrated their effectiveness. We achieve state-of-the-art results on popular datasets (COCO, VOC 07 and 12) and are the first to benchmark ResNet backbones.

References

[1] Aditya Arun, C. V. Jawahar, and M. Pawan Kumar. Dissimilarity coefficient based weakly supervised object detection. In *CVPR*, 2019. 4

[2] H. Bilen and A. Vedaldi. Weakly supervised deep detection networks. In *CVPR*, 2016. 1, 2, 4

[3] Yan Gao, Boxiao Liu, Nan Guo, Xiaochun Ye, Fang Wan, Haihang You, and Dongrui Fan. C-midn: Coupled multiple instance detection network with segmentation guidance for weakly supervised object detection. In *ICCV*, 2019. 1, 2, 4

[4] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V. Le. Dropblock: A regularization method for convolutional networks. In *NIPS*, 2018. 3

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2

[6] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *ICLR*, 2017. 3

[7] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *ICLR*, 2017. 3

[8] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016. 1

[9] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *TPAMI*, 2016. 1, 3

[10] Peng Tang, Xinggang Wang, Song Bai, Wei Shen, Xiang Bai, Wenyu Liu, and Alan Yuille. PCL: Proposal cluster learning for weakly supervised object detection. *TPAMI*, 2018. 1, 4

[11] Peng Tang, Xinggang Wang, Xiang Bai, and Wenyu Liu. Multiple instance detection network with online instance classifier refinement. In *CVPR*, 2017. 3, 4

[12] J.R.R. Uijlings, K.E.A. van de Sande, T. Gevers, and A.W.M.Smeulders. Selective search for object recognition. *IJCV*, 2013. 2

[13] Zhaoyang Zeng, Bei Liu, Jianlong Fu, Hongyang Chao, and Lei Zhang. WSOD2: Learning bottom-up and top-down objectness distillation for weakly-supervised object detection. In *ICCV*, 2019. 2, 4