Weakly Supervised Instance Segmentation by Deep Community Learning

Jaedong Hwang[1*]   Seohyun Kim[1*]   Jeany Son[2]   Bohyung Han[1]

[1]ECE & ASRI, Seoul National University, Seoul, Korea

[2]ETRI, Daejeon, Korea

[1]{jd730, goodbye61, bhhan}@snu.ac.kr, [2]jeany@etri.re.kr

Abstract

We present a weakly supervised instance segmentation algorithm based on deep community learning with multiple tasks. This task is formulated as a combination of weakly supervised object detection and semantic segmentation, where individual objects of the same class are identified and segmented separately. We address this problem by designing a unified deep neural network architecture, which has a positive feedback loop of object detection with bounding box regression, instance mask generation, instance segmentation, and feature extraction. Each component of the network makes active interactions with others to improve accuracy, and the end-to-end trainability of our model makes our results more robust and reproducible. The proposed algorithm achieves state-of-the-art performance in the weakly supervised setting without any additional training such as Fast R-CNN and Mask R-CNN on the standard benchmark dataset. The implementation of our algorithm is available on the project webpage: https://cv.snu.ac.kr/research/WSIS_CL.

1. Introduction

Object detection and semantic segmentation algorithms have achieved great success in recent years thanks to the advent of large-scale datasets [12, 29] as well as the development of deep learning technologies [15, 19, 31, 32]. However, most of existing image datasets have relatively simple forms of annotations such as image-level class labels, while many practical tasks require more sophisticated information such as bounding boxes and areas corresponding to object instances. Unfortunately, the acquisition of the complex labels needs significant human efforts, and it is challenging to construct a large-scale dataset containing such comprehensive annotations.

Instead of standard supervised learning formulations [5, 8, 18, 19], we tackle a more challenging task, weakly supervised instance segmentation, which relies only on image-
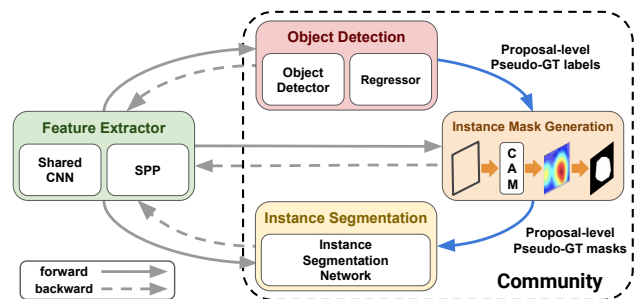
---

* Equal contribution.

Figure 1. The proposed community learning framework for weakly supervised instance segmentation. Our model is composed of object detection module, instance mask generation module, instance segmentation module and feature extractor, which constructs a positive feedback loop within a community. It first identifies positive detection bounding boxes from the detection module and generates pseudo-ground-truths of instance segmentation using class activation maps. The model is trained with multi-task loss of the three components using the pseudo-ground-truths. The final segmentation masks are obtained from the ensemble of outputs from instance mask generation and segmentation modules.

level class labels for instance-wise segmentation. This task shares critical limitations with many weakly supervised object recognition problems; trained models typically focus too much on discriminative parts of objects in the scene, and, consequently, fail to identify whole object regions and extract accurate object boundaries in a scene. Moreover, there are additional challenges in handling two problems jointly, weakly supervised object detection and semantic segmentation; incomplete ground-truths incur noisy estimation of labels in both tasks, which makes it difficult to take advantage of the joint learning formulation. For example, although object detection methods typically employ object proposals to provide rough information of object location and size, a naïve application of instance segmentation module to weakly supervised object detection results may not be successful in practice due to noise in object proposals.

Our approach aims to realize the goal using a deep neural network with multiple interacting task-specific components that construct a positive feedback loop. The whole model

is trained in an end-to-end manner and boosts performance of individual modules, leading to outstanding segmentation accuracy. We call such a learning concept *community learning*, and Figure 1 illustrates its application to weakly supervised instance segmentation. The community learning is different from multi-task learning that attempts to achieve multiple objectives in parallel without tight interaction between participating modules. The contributions of our work are summarized below:

- We introduce a deep community learning framework for weakly supervised instance segmentation, which is based on an end-to-end trainable deep neural network with active interactions between multiple tasks: object detection, instance mask generation, and object segmentation.

- We incorporate two empirically useful techniques for object localization, class-agnostic bounding box regression and segmentation proposal generation, which are performed without full supervision.

- The proposed algorithm achieves substantially higher performance than the existing weakly supervised approaches on the standard benchmark dataset without post-processing.

The rest of the paper is organized as follows. We briefly review related works in Section 2 and describe our algorithm with community learning in Section 3. Section 4 analyzes the experimental results on a benchmark dataset.

2. Related Works

This section reviews existing weakly supervised algorithms for object detection, semantic segmentation, and instance segmentation.

2.1. Weakly Supervised Object Detection

Weakly Supervised Object Detection (WSOD) aims to localize objects in a scene only with image-level class labels. Most of existing methods formulate WSOD as Multiple Instance Learning (MIL) problems [11] and attempt to learn detection models via extracting pseudo-ground-truth labels [4, 35, 36, 42]. WSDDN [4] combines classification and localization tasks to identify object classes and their locations in an input image. However, this technique is designed to find only a single object class and instance conceptually and often fails to solve the problems involving multiple labels and objects. Various approaches [22, 34, 35, 36, 38] tackle this issue by incorporating additional components, but they are still prone to focus on the discriminative parts of objects instead of whole object regions. Recently, there are several research integrating semantic segmentation to improve detection performance [10, 28, 33, 40]. WCCN [10] and TS2C [40] filter

out object proposals using semantic segmentation results, but still have trouble in identifying spatially overlapped objects in the same class. Meanwhile, SDCN [28] utilizes semantic segmentation result to refine pseudo-ground-truths. WS-JDS [33] leverages weakly supervised semantic segmentation module that estimates importance for object proposals. Although the core idea is valuable and the segmentation module improves detection performance, the instance segmentation performance improvement is marginal compared to simple box masking of its baselines [4, 22].

2.2. Weakly Supervised Semantic Segmentation

Weakly Supervised Semantic Segmentation (WSSS) is a task to estimate pixel-level semantic labels in an image based on image-level class labels only. Class Activation Map (CAM) [43] is widely used for WSSS because it generates class-specific likelihood maps using the supervision for image classification. SPN [24], one of the early works that exploit CAM for WSSS, combines CAM with superpixel segmentation result to extract accurate class boundaries in an image. AffinityNet [2] propagates the estimated class labels using semantic affinities between adjacent pixels. Ge *et al*. [14] employ a pretrained object detector to obtain segmentation labels. Recent approaches [21, 24, 26, 27, 39, 41] often train their models end-to-end. DSRG [21] and MCOF [39] propose iterative refinement procedures starting from CAM. FickleNet [26] performs stochastic feature selection in its convolutional layers and captures the regularized shapes of objects.

2.3. Instance Segmentation

Instance segmentation can be regarded as a combination of object localization and semantic segmentation, which needs to identify individual object instances. There exist several fully supervised approaches [5, 8, 18, 19]. Haydr *et al*. [18] utilize Region Proposal Network (RPN) [32] to detect individual instances and leverage Object Mask Network (OMN) for segmentation. Mask R-CNN [19], Masklab [5] and MNC [8] have similar procedures to predict their pixel-level segmentation labels.

There have been recent works for Weakly Supervised Instance Segmentation (WSIS) based on image-level class labels only [1, 13, 25, 44, 45]. Peak Response Map (PRM) [44] takes the peaks of an activation map as pivots for individual instances and estimates the segmentation mask of each object using the pivots. Instance Activation Map (IAM) [45] selects pseudo-ground-truths out of precomputed segment proposals based on PRM to learn segmentation networks. Label-PEnet [13] combines various components with different functionalities to obtain the final segmentation masks. However, it involves many duplicate operations across the components and requires very complex training pipeline. There are a few attempts to gener-

ate pseudo-ground-truth segmentation maps based on weak supervision and forward them to the well-established network [19] for instance segmentation [1, 25]. To improve accuracy, the algorithms often employ post-processing such as MCG proposals [3] or denseCRF [23].

3. Proposed Algorithm

This section describes our community learning framework based on an end-to-end trainable deep neural network for weakly supervised instance segmentation.

3.1. Overview and Motivation

One of the most critical limitations in a naïve combination of detection and segmentation networks for weakly supervised instance segmentation is that the learned models often attend to small discriminative regions of objects and fail to recover missing parts of target objects. This is partly because segmentation networks rely on noisy detection results without proper interactions and the benefit of the iterative label refinement procedure is often saturated in the early stage due to the strong correlation between outputs from two modules.

To alleviate this drawback, we propose a deep neural network architecture that constructs a circular chain along with the components and generates desirable instance detection and segmentation results. The chain facilitates the interactions along individual modules to extract useful information. Specifically, the object detector generates proposal-level pseudo-ground-truth labels. They are used to create pseudo-ground-truth masks for instance segmentation module, which estimates the final segmentation labels of individual proposals using the masks. These three network components make up a community and collaborate to update the weights of the backbone network for feature extraction, which leads to regularized representations robust to overfitting to poor local optima.

3.2. Network Architecture

Figure 2 presents the network architecture of our weakly supervised object detection and segmentation algorithm. As mentioned earlier, the proposed network consists of four parts: feature extractor, object detector with bounding box regressor, instance mask generator and instance segmentation module. Our feature extraction network is made of shared fully convolutional layers, where the feature of each proposal is obtained from the Spatial Pyramid Pooling (SPP) [20] layers on the shared feature map and fed to the other modules.

3.2.1 Object Detection Module

For object detection, a 7×7 feature map is extracted from the SPP layer for each object proposal and forwarded to the last residual block (res5). Then, we pass these features to both the detector and the regressor. Since this process is compatible with any end-to-end trainable object detection network based on weak supervision, we adopt one of the most popular weakly supervised object detection networks, referred to as OICR [36], which has three refinement layers after the base detector. For each image-level class label, we extract foreground proposals based on their estimated scores corresponding to the label and apply a non-maximum suppression (NMS) to reduce redundancy. Background proposals are randomly sampled from the proposals that are overlap with foreground proposals below a threshold. Among the foreground proposals, the one with the highest score for each class is selected as a pseudo-ground-truth bounding box.

Bounding box regression is typically conducted under full supervision to refine the proposals corresponding to objects. However, learning a regressor in our problem is particularly challenging since it is prone to be biased by discriminative parts of objects; such a characteristic is difficult to control in a weakly supervised setting and is aggravated in class-specific learning. Hence, unlike [15, 16, 32], we propose a class-agnostic bounding box regressor based on pseudo-ground-truths to avoid overly discriminative representation learning and provide better regularization effect. Note that a class-agnostic regressor has not been explored actively yet since fully supervised models can exploit accurate bounding box annotations and learning a regressor with weak labels only is not common. If a proposal has a higher IoU with its nearest pseudo-ground-truth proposal than a threshold, the proposal and the pseudo-ground-truth proposal are paired to learn the regressor.

3.2.2 Instance Mask Generation (IMG) Module

This module constructs pseudo-ground-truth masks for instance segmentation using the proposal-level class labels given by our object detector. It takes the feature of each proposal from the SPP layers attached to multiple convolutional layers as shown in Figure 2. Since the IMG module utilizes hierarchical representations from different levels in a backbone network, it can deal with multi-scale objects effectively.

We construct pseudo-ground-truth masks for individual proposals by integrating the following additional features into CAM [43]. First, we compute a background class activation map by augmenting a channel corresponding to the background class. This map is useful to distinguish objects from the background. Second, instead of the Global Average Pooling (GAP) adopted in the standard CAM, we employ the weighted GAP to give more weights to the center pixels within proposals. It computes a weighted average of the input feature maps, where the weights are given by an
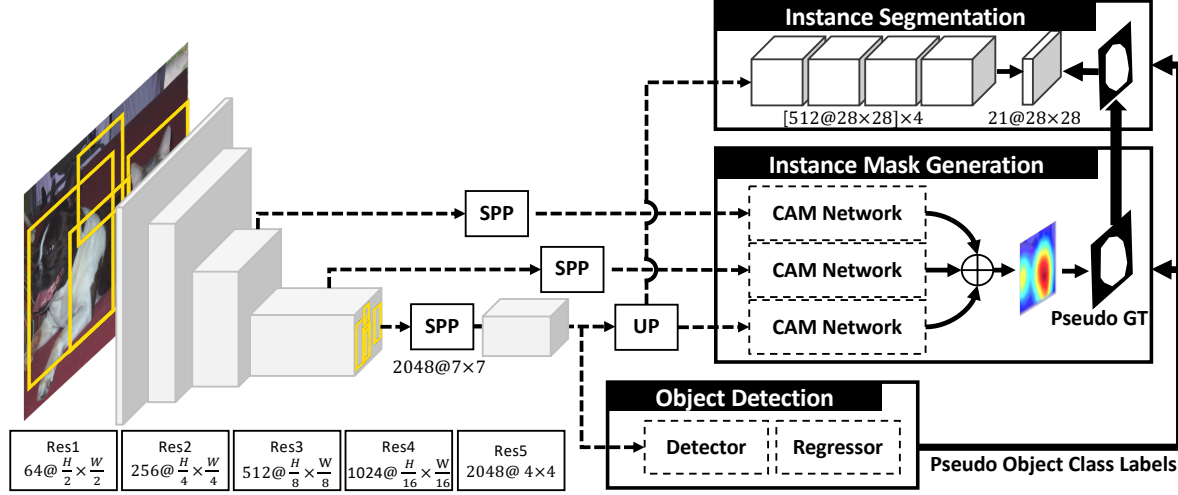
Figure 2. The proposed network architecture for weakly supervised instance segmentation. Our end-to-end trainable network consists of four parts: (a) *feature extraction network* computes the shared feature maps and provides proposal-level features with the other networks, (b) *object detection network* identifies the location of objects and gives a pseudo-label of object class to each proposal, (c) *instance mask generation network* constructs the class activation map for given proposals using predicted pseudo-labels from the detector, (d) *instance segmentation network* predicts segmentation masks and is learned with the outputs of the above networks as pseudo-ground-truths.

isotropic Gaussian kernel. Third, we convert input features f of the CAM module to log scale values, *i.e.*, $\log(1 + f)$, which penalizes excessively high peaks in the CAM and leads to spatially regularized feature maps appropriate for robust segmentation.

The output of the IMG module, denoted by $\mathbf{M}$, is an average of three CAMs to which min-max normalizations [30] are applied. For each selected proposal, the pseudo-ground-truth mask $\widetilde{\mathbf{M}} \in \mathbb{R}^{(C+1) \times T^2}$ for instance segmentation is given by the following equation using the three CAMs, $\mathbf{M}_k$ ($k = 1, 2, 3$),

$$\widetilde{\mathbf{M}} = \delta \left[\frac{1}{3} \sum_{k=1}^{3} \mathbf{M}_k > \xi \right], \tag{1}$$

where $\mathbf{M}_k \in \mathbb{R}^{(C+1) \times T^2}$ is the kth CAM whose size is $T \times T$ for all classes including background, $\delta[\cdot]$ is an element-wise indicator function, and ξ is a predefined threshold.

3.2.3 Instance Segmentation Module

For instance segmentation, the output of the res5 block is upsampled to $T \times T$ activation maps and provided to four convolutional layers along with ReLU layers and the final segmentation output layer as illustrated in Figure 2. This module learns a pixel-wise binary classification label for each proposal based on the pseudo-ground-truth mask $\widetilde{\mathbf{M}}^c$, provided by the IMG module. The predicted mask of each proposal is a class-specific binary mask, where the class label c is determined by the detector. Note that our model is

compatible with any semantic segmentation network.

3.3. Losses

The overall loss function of our deep community learning framework is given by the sum of losses from the three modules as

$$\mathcal{L} = \mathcal{L}_{\text{det}} + \mathcal{L}_{\text{img}} + \mathcal{L}_{\text{seg}}, \tag{2}$$

where $\mathcal{L}_{\text{det}}$, $\mathcal{L}_{\text{img}}$, and $\mathcal{L}_{\text{seg}}$ denote detection loss, instance mask generation loss, and instance segmentation loss, respectively. The three terms interact with each other to train the backbone network including the feature extractor in an end-to-end manner.

3.3.1 Object Detection Loss

The object detection module is trained with the sum of classification loss $\mathcal{L}_{\text{cls}}$, refinement loss $\mathcal{L}_{\text{refine}}$, and bounding box regression loss $\mathcal{L}_{\text{reg}}$. The features extracted from the individual object proposals are given to the detection module based on OICR [36]. Image classification loss $\mathcal{L}_{\text{cls}}$ is calculated by computing the cross-entropy between image-level ground-truth class label $\boldsymbol{y} = (y_1, \ldots, y_C)^{\mathrm{T}}$ and its corresponding prediction $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_C)^{\mathrm{T}}$, which is given by

$$\mathcal{L}_{\text{cls}} = -\sum_{c=1}^{C} y_c \log \phi_c + (1 - y_c) \log(1 - \phi_c), \tag{3}$$

where C is the number of classes in a dataset. As in the original OICR, the pseudo-ground-truth of each object proposal in the refinement layers is obtained from the outputs

of their preceding layers, where the supervision of the first refinement layer is provided by WSDDN [4]. The loss of the kth refinement layer is computed by a weighted sum of losses over all proposals as

$$\mathcal{L}_{\text{refine}}^k = -\frac{1}{|R|} \sum_{r=1}^{|R|} \sum_{c=1}^{C+1} w_r^k y_{cr}^k \log x_{cr}^k, \qquad (4)$$

where x_{cr}^k denotes a score of the rth proposal with respect to class c in the kth refinement layer, w_r^k is a proposal weight obtained from the prediction score in the preceding refinement layer, and $|R|$ is the number of proposals. In the refinement loss function, there are $C + 1$ classes because we also consider background class.

Regression loss $\mathcal{L}_{\text{reg}}$ employs smooth ℓ_1-norm between a proposal and its matching pseudo-ground-truth, following the bounding box regression literature [15, 32]. The regression loss is defined as follows:

$$\mathcal{L}_{\text{reg}} = \frac{1}{|R|} \sum_{r=1}^{|R|} \sum_{j=1}^{|\mathcal{G}|} q_{rj} \sum_{k \in \{x,y,w,h\}} \text{smooth}_{\ell_1}(t_{rjk} - v_{rk}), \qquad (5)$$

where $\mathcal{G}$ is a set of pseudo-ground-truths, q_{rj} is an indicator variable denoting whether the rth proposal is matched with the jth pseudo-ground-truth, v_{rk} is a predicted bounding box regression offset of the kth coordinate for r^{th} proposal and t_{rjk} is the desirable offset parameter of the kth coordinate between the rth proposal and the jth pseudo-ground-truth as in R-CNN [16].

The detection loss $\mathcal{L}_{\text{det}}$ is the sum of image classification loss, bounding box regression loss, and K refinement losses, which is given by

$$\mathcal{L}_{\text{det}} = \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{reg}} + \sum_{k=1}^{K} \mathcal{L}_{\text{refine}}^k, \qquad (6)$$

where $K = 3$ in our implementation.

3.3.2 Instance Mask Generation Loss

For training CAMs in the IMG module, we adopt average classification scores from three refinement branches of our detection network. The loss function of the kth CAM network, denoted by $\mathcal{L}_{\text{cam}}^k$, is given by a binary cross entropy loss as

$$\mathcal{L}_{\text{cam}}^k = -\frac{1}{|R|} \sum_{r=1}^{|R|} \sum_{c=1}^{C+1} \widetilde{y}_{rc} \log p_{rc}^k + (1 - \widetilde{y}_{rc}) \log(1 - p_{rc}^k), \qquad (7)$$

where $\widetilde{y}_{rc}$ is an one-hot encoded pseudo-label from detection branch of the rth proposal for class c, and p_{rc}^k is a softmax score of the same proposal for the same class obtained

by the weighted GAP from the last convolutional layer. The instance mask generation loss is the sum of all the CAM losses as shown in the following equation:

$$\mathcal{L}_{\text{img}} = \sum_{k=1}^{3} \mathcal{L}_{\text{cam}}^k. \qquad (8)$$

3.3.3 Instance Segmentation Loss

The loss in the segmentation network is obtained by comparing the network outputs with the pseudo-ground-truth $\widetilde{\mathbf{M}}$ using a pixel-wise binary cross entropy loss for each class, which is given by

$$\mathcal{L}_{\text{seg}} = -\frac{1}{T^2} \sum_{r}^{|R|} \sum_{c}^{C+1} \sum_{(i,j) \in T \times T} m_{rc}^{ij} \log s_{rc}^{ij} \qquad (9)$$
$$+ (1 - m_{rc}^{ij}) \log \left(1 - s_{rc}^{ij}\right),$$

where m_{rc}^{ij} means a binary element at (i, j) of $\widetilde{\mathbf{M}}$ for the rth proposal, and s_{rc}^{ij} is the output value of the segmentation network, $\mathbf{S} \in \mathbb{R}^{|R| \times (C+1) \times T^2}$, at location (i, j) of the rth proposal.

3.4. Inference

Our model sequentially predicts object detection and instance segmentation for each proposal in a given image. For object detection, we use the average scores of three refinement branches in the object detection module. Each regressed proposal is labeled as the class that corresponds to the maximum score. We apply a non-maximum suppression with IoU threshold 0.3 to the proposals. The survived proposals are regarded as detected objects and used to estimate pseudo-labels for instance segmentation.

For instance segmentation, we select the foreground activation map of the identified class c, $\mathbf{M}^c$, from the IMG module and the corresponding segmentation score map, $\mathbf{S}^c$, from instance segmentation module for each detected object. The final instance segmentation label is given by the ensemble of two results,

$$\mathbf{O}^c = \delta \left[\frac{\mathbf{M}^c + \mathbf{S}^c}{2} > \xi \right], \qquad (10)$$

where $\mathbf{O}^c$ is a binary segmentation mask for detected class c, $\delta[\cdot]$ is an element-wise indicator function, and ξ is a threshold identical used in Eq. (1). For post-processing, we utilize Multiscale Combinatorial Grouping (MCG) proposals [3] as used in PRM [44]. Each instance segmentation mask is substituted as a max overlap MCG proposal. Since the MCG proposal is a group of superpixels, it contains boundary information. Hence, if a segmentation output covers overall shape well, MCG proposal is greatly helpful to catch details of an object.

Table 1. Instance segmentation results on the PASCAL VOC 2012 segmentation *val* set with two different types of supervisions ($\mathcal{I}$: image-level class label, $\mathcal{C}$: object count). The numbers in red and blue denote the best and the second best scores without Mask R-CNN re-training, respectively.

Method	Supervision	Post-procesing	$mAP_{0.25}$	$mAP_{0.5}$	$mAP_{0.75}$	ABO
WISE [25] w/ Mask R-CNN	$\mathcal{I}$	✓	49.2	41.7	23.7	55.2
IRN [1] w/ Mask R-CNN	$\mathcal{I}$	✓	-	46.7	-	-
Cholakkal *et al*. [7]	$\mathcal{I}+\mathcal{C}$	✓	48.5	30.2	14.4	44.3
PRM [44]	$\mathcal{I}$	✓	44.3	26.8	9.0	37.6
IAM [45]	$\mathcal{I}$	✓	45.9	28.3	11.9	41.9
Label-PEnet [13]	$\mathcal{I}$	✓	49.2	30.2	**12.9**	41.4
Ours	$\mathcal{I}$		**57.0**	**35.9**	5.8	**43.8**
	$\mathcal{I}$	✓	**56.6**	**38.1**	**12.3**	**48.2**

4. Experiments

This section describes our setting for training and evaluation and presents the experimental results of our algorithm in comparison to the existing methods. We also analyze various aspects of the proposed network.

4.1. Training

We use Selective Search [37] for generating bounding box proposals. All fully connected layers in the detection and the IMG modules are initialized randomly using a Gaussian distribution $(0, 0.01^2)$. The learning rate is 0.001 at the beginning and reduced to 0.0001 after 90K iterations. The hyper-parameter in the weight decay term is 0.0005, the batch size is 2, and the total training iteration is 120K. We use 5 image scales of $\{480, 576, 688, 864, 1000\}$, which are based on the shorter size of an image, for data augmentation and ensemble in training and testing. The NMS threshold for selecting foreground proposals is 0.3 and ξ in Eq (1) is set to 0.4 following MNC [8]. For regression, a proposal is associated with a pseudo-ground-truth if the IoU is larger than 0.6. The output size T of the IMG and instance segmentation modules is 28. Our model is implemented on PyTorch and the experiments are conducted on a single NVIDIA Titan XP GPU.

4.2. Datasets and Evaluation Metrics

We use PASCAL VOC 2012 segmentation dataset [12] to evaluate our algorithm. The dataset is composed of 1,464, 1,449, and 1,456 images for training, validation, and testing, respectively, for 20 object classes. We use the standard augmented training set (*trainaug*) with 10,582 images to learn our network, following the prior segmentation research [1, 6, 7, 13, 17, 44, 45]. In our weakly supervised learning scenario, we only use image-level class labels to train the model. Detection and instance segmentation accuracies are measured on PASCAL VOC 2012 segmentation validation (*val*) set.

We employ the standard mean average precision (mAP)

Table 2. Instance segmentation results on the PASCAL VOC 2012 segmentation *train* set. [1, 25] report results without Mask R-CNN obtained from their original papers.

	WISE [25]	IRN [1]	Ours
$mAP_{0.5}$	25.8	37.7	**39.2**

to evaluate object detection performance, where a bounding box is regarded as a correct detection if it overlaps with a ground-truth more than a threshold, *i.e.* IoU > 0.5. CorLoc [9] is also used to evaluate the localization accuracy on the *trainaug* dataset. For instance segmentation task, we evaluate performance of an algorithm using mAPs at IoU thresholds 0.25, 0.5 and 0.75. We also use Average Best Overlap (ABO) to present overall instance segmentation performance of our model.

4.3. Comparison with Other Algorithms

We compare our algorithm with existing weakly supervised instance segmentation approaches [7, 13, 44, 45]. Table 1 shows that our algorithm generally outperforms the prior arts even without post-processing. Note that our post-processing using MCG proposals [3] improves mAP at high thresholds and ABO significantly, and leads to outstanding performance in terms of both mAP and ABO after all. We believe that such large gaps come from the effective regularization given by our community learning. The accuracy of our model is not as good as the method given by Mask R-CNN re-training [1, 25], but direct comparison is not fair due to the retraining issue. Table 2 illustrates that our model outperforms the methods without re-training on *train* split.

4.4. Ablation Study

We discuss the contribution of each component in the network and the effectiveness of our training strategy. We also compare two different regression strategies—class-agnostic vs. class-specific—using detection scores. Note that we present the results without post-processing for the

Table 3. Contribution of individual components integrated into our algorithm study. The evaluation is performed on PASCAL VOC 2012 segmentation *val* set for mAP and *trainaug* set for CorLoc (* indicates that detection bounding boxes are used as segmentation results as well).

Architecture	Instance Segmentation $\text{mAP}_{0.5}$	Object Detection mAP	Object Detection CorLoc
Detector	18.8*	45.3	63.6
Detector + IMG	32.8	48.6	66.3
Detector + IMG + IS	33.7	49.2	66.8
Detector + REG + IMG + IS	**35.9**	**53.2**	**70.8**

Table 4. Accuracy of the variants of IMG module with background class (BG), weighted GAP (wGAP), and feature smoothing (FS), based on the ResNet50 backbone without REG

	BG	BG + wGAP	BG + FS	wGAP + FS	All
$\text{mAP}_{0.5}$	28.8	30.0	31.8	27.4	**33.7**

ablation study to verify the contribution of each component clearly.

4.4.1 Network Components

We analyze the effectiveness of individual modules for instance segmentation and object detection. For comparisons, we measure $\text{mAP}_{0.5}$ for instance segmentation and mAP for object detection on PASCAL VOC 2012 segmentation *val* set while computing CorLoc on the *trainaug* set. Note that the instance segmentation accuracy of the detection-only model is given by using detected bounding boxes as segmentation masks. All models are trained on PASCAL VOC 2012 segmentation *trainaug* set.

Table 3 presents that the IMG and Instance Segmentation (IS) modules are particularly helpful to improve accuracy for both tasks. By adding the two components, our model achieves accuracy gain in detection by 3.9% and 3.2% points in terms of mAP and CorLoc, respectively, compared to the baseline detector. Additionally, bounding box regression (REG) enhances performance by generating better pseudo-ground-truths.

4.4.2 IMG module

We further investigate the components in the IMG module and summarize the results in Table 4. All results are from the experiments without bounding box regression to demonstrate the impact of individual components clearly. All the three tested components make substantial contribution for performance improvement. The background class activation map models background likelihood within a bounding box explicitly and facilitates the comparison with fore-

Table 5. Comparison our model with a combination of OICR and AffinityNet on PASCAL VOC 2012 segmentation *val* set

Model	OICR + AffinityNet	OICR (ResNet50) + AffinityNet	Ours
$\text{mAP}_{0.5}$	27.3	33.3	**35.9**

Table 6. Comparison of class-agnostic regressor and class-specific regressor into our algorithm in terms of detection performance. The evaluation is performed on PASCAL VOC 2012 segmentation *val* set for mAP and *trainaug* set for CorLoc.

Model	mAP	CorLoc
Ours w/o REG	49.2	66.8
Ours (class-specific)	50.4	68.4
Ours (class-agnostic)	**53.2**	**70.1**

ground counterparts. The feature smoothing regularizes excessively discriminative activations in the inputs to CAM module while the weighted GAP pays more attention to the proper region for segmentation.

4.4.3 Comparison to a Simple Algorithm Combination

To demonstrate the benefit of our unified framework, we compare the proposed algorithm with a straightforward combination of weakly supervised object detection and semantic segmentation methods. Table 5 presents the result from a combination of weakly supervised object detection algorithm, OICR [36], and a weakly supervised semantic segmentation algorithm, AffinityNet [2]. Note that both OICR and AffinityNet are competitive approaches in their target tasks. We train the two models independently and combine their results by providing a segmentation label map using AffinityNet for each detection result obtained from OICR. The proposed algorithm based on a unified end-to-end training outperforms the simple combination of two separate modules even without post-processing.

4.4.4 Comparison to Class-Specific Box Regressor

We compare the results from the class-agnostic and class-specific bounding box regressors in terms of mAP and CorLoc. Table 6 presents that bounding box regressors turn out to be learned effectively despite incomplete supervision. It further shows that the class-agnostic bounding box regressor clearly outperforms the class-specific version. We believe that this is partly because sharing a regressor over all classes reduces the bias observed in individual classes and regularizes overly discriminative representations.

4.5. Qualitative Results

Figure 3 shows instance segmentation results from our model after post-processing and identified bounding boxes

Figure 3. Instance segmentation results on PASCAL VOC 2012 segmentation *val* set. Green rectangle is a detected object bounding box.
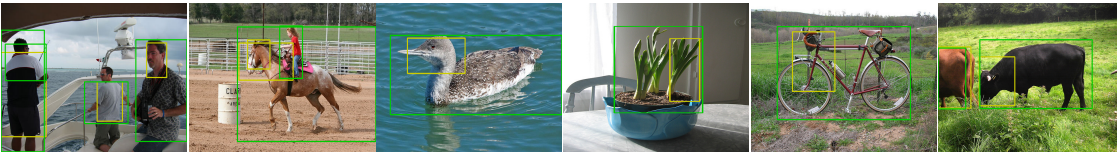

Figure 4. Qualitative results of detection on PASCAL VOC 2012 segmentation *val* set. Green rectangle is generated by our model and yellow one indicates the output of detector-only model (OICR [36] based on ResNet50).

on PASCAL VOC 2012 segmentation *val* set. Refer to our supplementary material for more details. Our model successfully segments whole regions of objects and discriminates each object in a same class within an input image via predicted object proposals. Figure 4 compares detection results from our full model and a detector-only model, OICR with the ResNet50 backbone network, on the same dataset. Our model is more robust to localize a whole object since the features are better regularized by joint learning of Object Detection, IMG, and Instance Segmentation modules.

5. Conclusion

We presented a unified end-to-end deep neural network for weakly supervised instance segmentation via community learning. Our framework trains three subnetworks jointly with a shared feature extractor, which performs object detection with bounding box regression, instance mask generation, and instance segmentation. These components interact with each other closely and form a positive feedback loop with cross-regularization for improving quality of individual tasks. Our class-agnostic bounding box regressor successfully regularizes object detectors even with weak supervisions only while the post-processing based on MCG mask proposals improves accuracy substantially.

The proposed algorithm outperforms the previous state-of-the-art weakly supervised instance segmentation methods and the weakly supervised object detection baseline on PASCAL VOC 2012 with a simple segmentation module. Since our framework does not rely on particular network architectures for object detection and instance segmentation modules, using better detector or segmentation network would improve the performance of our framework.

Acknowledgements

References

[1] Jiwoon Ahn, Sunghyun Cho, and Suha Kwak. Weakly Supervised Learning of Instance Segmentation With Inter-Pixel Relations. In *CVPR*, 2019.

[2] Jiwoon Ahn and Suha Kwak. Learning Pixel-Level Semantic Affinity With Image-Level Supervision for Weakly Supervised Semantic Segmentation. In *CVPR*, 2018.

[3] Pablo Arbeláez, Jordi Pont-Tuset, Jonathan T Barron, Ferran Marques, and Jitendra Malik. Multiscale Combinatorial Grouping. In *CVPR*, 2014.

[4] Hakan Bilen and Andrea Vedaldi. Weakly Supervised Deep Detection Networks. In *CVPR*, 2016.

[5] Liang-Chieh Chen, Alexander Hermans, George Papandreou, Florian Schroff, Peng Wang, and Hartwig Adam. MaskLab: Instance Segmentation by Refining Object Detection With Semantic and Direction Features. In *CVPR*, 2018.

[6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *TPAMI*, 40(4):834–848, 2018.

[7] Hisham Cholakkal, Guolei Sun, Fahad Shahbaz Khan, and Ling Shao. Object Counting and Instance Segmentation With Image-Level Supervision. In *CVPR*, 2019.

[8] Jifeng Dai, Kaiming He, and Jian Sun. Instance-aware Semantic Segmentation via Multi-task Network Cascades. In *CVPR*, 2016.

[9] Thomas Deselaers, Bogdan Alexe, and Vittorio Ferrari. Weakly Supervised Localization and Learning with Generic Knowledge. *IJCV*, 100(3):275–293, 2012.

[10] Ali Diba, Vivek Sharma, Ali Pazandeh, Hamed Pirsiavash, and Luc Van Gool. Weakly Supervised Cascaded Convolutional Networks. In *CVPR*, 2017.

[11] Thomas G Dietterich, Richard H Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence*, 1997.

[12] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 111(1):98–136, 2015.

[13] Weifeng Ge, Sheng Guo, Weilin Huang, and Matthew R. Scott. Label-PEnet: Sequential Label Propagation and Enhancement Networks for Weakly Supervised Instance Segmentation. In *ICCV*, 2019.

[14] 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*, 2018.

[15] Ross Girshick. Fast R-CNN. In *CVPR*, 2015.

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

[17] Bharath Hariharan, Pablo Arbelaez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *ICCV*, 2011.

[18] Zeeshan Hayder, Xuming He, and Mathieu Salzmann. Boundary-Aware Instance Segmentation. In *CVPR*, 2017.

[19] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. In *ECCV*, 2014.

[21] Zilong Huang, Xinggang Wang, Jiasi Wang, Wenyu Liu, and Jingdong Wang. Weakly-Supervised Semantic Segmentation Network with Deep Seeded Region Growing. In *CVPR*, 2018.

[22] Vadim Kantorov, Maxime Oquab, Minsu Cho, and Ivan Laptev. ContextLocNet: Context-aware deep network models for weakly supervised localization. In *ECCV*, 2016.

[23] Philipp Krähenbühl and Vladlen Koltun. Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials. In *NeurIPS*, 2011.

[24] Suha Kwak, Seunghoon Hong, and Bohyung Han. Weakly Supervised Semantic Segmentation Using Superpixel Pooling Network. In *AAAI*, 2017.

[25] Issam H. Laradji, David Vázquez, and Mark W. Schmidt. Where are the Masks: Instance Segmentation with Image-level Supervision. In *BMVC*, 2019.

[26] Jungbeom Lee, Eunji Kim, Sungmin Lee, Jangho Lee, and Sungroh Yoon. FickleNet: Weakly and Semi-supervised Semantic Image Segmentation using Stochastic Inference. In *CVPR*, 2019.

[27] Jungbeom Lee, Eunji Kim, Sungmin Lee, Jangho Lee, and Sungroh Yoon. Frame-to-Frame Aggregation of Active Regions in Web Videos for Weakly Supervised Semantic Segmentation. In *ICCV*, 2019.

[28] Xiaoyan Li, Meina Kan, Shiguang Shan, and Xilin Chen. Weakly Supervised Object Detection with Segmentation Collaboration. *ICCV*, 2019.

[29] 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*, 2014.

[30] S Patro and Kishore Kumar Sahu. Normalization: A Preprocessing Stage. *arXiv preprint arXiv:1503.06462*, 2015.

[31] Joseph Redmon and Ali Farhadi. YOLOv3: An Incremental Improvement. *arXiv*, abs/1804.02767, 2018.

[32] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *NeurIPS*, 2015.

[33] Yunhang Shen, Rongrong Ji, Yan Wang, Yongjian Wu, and Liujuan Cao. Cyclic Guidance for Weakly Supervised Joint Detection and Segmentation. In *CVPR*, 2019.

[34] Jeany Son, Daniel Kim, Solae Lee, Suha Kwak, Minsu Cho, and Bohyung Han. Forget & Diversify: Regularized Refinement for Weakly Supervised Object Detection. In *ACCV*, 2018.

[35] 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.

[36] Peng Tang, Xinggang Wang, Xiang Bai, and Wenyu Liu. Multiple Instance Detection Network with Online Instance Classifier Refinement. In *CVPR*, 2017.

[37] 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.

[38] Fang Wan, Pengxu Wei, Jianbin Jiao, Zhenjun Han, and Qixiang Ye. Min-Entropy Latent Model for Weakly Supervised Object Detection. In *CVPR*, 2018.

[39] Xiang Wang, Shaodi You, Xi Li, and Huimin Ma. Weakly-Supervised Semantic Segmentation by Iteratively Mining Common Object Features. In *CVPR*, 2018.

[40] 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*, 2018.

[41] Yu Zeng, Yunzhi Zhuge, Huchuan Lu, and Lihe Zhang. Joint Learning of Saliency Detection and Weakly Supervised Semantic Segmentation. In *ICCV*, 2019.

[42] Yongqiang Zhang, Yancheng Bai, Mingli Ding, Yongqiang Li, and Bernard Ghanem. W2F: A Weakly-Supervised to Fully-Supervised Framework for Object Detection. In *CVPR*, 2018.

[43] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning Deep Features for Discriminative Localization. In *CVPR*, 2016.

[44] Yanzhao Zhou, Yi Zhu, Qixiang Ye, Qiang Qiu, and Jianbin Jiao. Weakly Supervised Instance Segmentation using Class Peak Response. In *CVPR*, 2018.

[45] Yi Zhu, Yanzhao Zhou, Huijuan Xu, Qixiang Ye, David Doermann, and Jianbin Jiao. Learning Instance Activation Maps for Weakly Supervised Instance Segmentation. In *CVPR*, 2019.