The Devil is in the Points: Weakly Semi-Supervised Instance Segmentation via Point-Guided Mask Representation

Beomyoung Kim\textsuperscript{1,2} Joonhyun Jeong\textsuperscript{1,2} Dongyoon Han\textsuperscript{3} Sung Ju Hwang\textsuperscript{2} \\
NAVER Cloud, ImageVision\textsuperscript{1} KAIST\textsuperscript{2} NAVER AI Lab\textsuperscript{3}

Abstract

In this paper, we introduce a novel learning scheme named weakly semi-supervised instance segmentation (WSSIS) with point labels for budget-efficient and high-performance instance segmentation. Namely, we consider a dataset setting consisting of a few fully-labeled images and a lot of point-labeled images. Motivated by the main challenge of semi-supervised approaches mainly derives from the trade-off between false-negative and false-positive instance proposals, we propose a method for WSSIS that can effectively leverage the budget-friendly point labels as a powerful weak supervision source to resolve the challenge. Furthermore, to deal with the hard case where the amount of fully-labeled data is extremely limited, we propose a MaskRefineNet that refines noise in rough masks. We conduct extensive experiments on COCO and BDD100K datasets, and the proposed method achieves promising results comparable to those of the fully-supervised model, even with 50% of the fully labeled COCO data (38.8% vs. 39.7%). Moreover, when using as little as 5% of fully labeled COCO data, our method shows significantly superior performance over the state-of-the-art semi-supervised learning method (33.7% vs. 24.9%). The code is available at \url{https://github.com/clovaai/PointWSSIS}.

1. Introduction

Recently proposed instance segmentation methods \cite{5, 8, 9, 13, 15, 16, 23, 32, 42, 45} have achieved remarkable performance owing to the availability of abundant of segmentation labels for training. However, compared to other label types (e.g., bounding box or point), segmentation labels necessitate delicate pixel-level annotations, demanding much more monetary cost and human effort. Consequently, weakly-supervised instance segmentation (WSIS) and semi-supervised instance segmentation (SSIS) approaches have gained attention to reduce annotation costs. WSIS approaches alternatively utilize inexpensive weak labels such as image-level labels \cite{1, 26, 52}, point labels \cite{11, 26, 27} or bounding box labels \cite{19, 31, 43}. Besides, SSIS approaches \cite{47, 51} employ a small amount of pixel-level (fully) labeled data and a massive amount of unlabeled data. Although they have shown potential in budget-efficient instance segmentation, there still exists a large performance gap between theirs and the results of fully-supervised learning methods.

Specifically, SSIS approaches often adopt the following training pipeline: (1) train a base network with fully labeled data, (2) generate pseudo instance masks for unlabeled images using the base network, and (3) train a target network using both full and pseudo labels. The major challenge of SSIS approaches comes from the trade-off between the number of missing (i.e., false-negative) and noise (i.e., false-positive) samples in the pseudo labels. Namely, some strategies for reducing false-negatives, which is equivalent to increasing true-positives, often end up increasing false-positives accordingly; an abundance of false-negatives or false-positives in pseudo labels impedes stable convergence of the target network. However, optimally reducing false-negatives/positives while increasing true-positives is quite challenging and remains a significant challenge for SSIS.
To address this challenge, we first revisit the fundamental behavior of the instance segmentation framework. Most existing instance segmentation methods adopt a two-step inference process as shown in Figure 1: (1) generate instance proposals where an instance is represented as a box [9, 16, 21, 32] or point [42, 44–46] in proposal branch, and (2) produce instance masks for each instance proposal in mask branch. As shown in Figure 1, if the network fails to obtain an instance proposal (i.e., false-negative proposal), it cannot produce the corresponding instance mask. Although the network could represent the instance mask in the mask branch, the absence of the proposal becomes the bottleneck for producing the instance mask. From the behavior of the network, we suppose that addressing the bottleneck in the proposals is a shortcut to the success of the SSIS.

Motivated by the above observations, we rethink the potential of using point labels as weak supervision. The point label contains only a one-pixel categorical instance cue but is budget-friendly as it is as easy as providing image-level labels by human annotators [3]. We note that the point label can be leveraged as an effective source to (i) resolve the performance bottleneck of the instance segmentation network and (ii) optimally balance the trade-off between false-negative and false-positive proposals. Thus, we formulate a new practical training scheme, Weakly Semi-Supervised Instance Segmentation (WSSIS) with point labels. In the WSSIS task, we utilize a small amount of fully labeled data and a massive amount of point labeled data for budget-efficient and high-performance instance segmentation.

Under the WSSIS setting, we filter out the proposals to keep only true-positive proposals using the point labels. Then, given the true-positive proposals, we exploit the mask representation of the network learned by fully labeled data to produce high-quality pseudo instance masks. For properly leveraging point labels, we consider the characteristics of the feature pyramid network (FPN) [34], which consists of multi-level feature maps for multi-scale instance recognition. Each pyramid level is trained to recognize instances of particular sizes, and extracting instance masks from unfit levels often causes inaccurate predictions, as shown in Figure 4. However, since point labels do not have instance size information, we handle this using an effective strategy named Adaptive Pyramid-Level Selection. We estimate which level is the best fit based on the reliability of the network (i.e., confidence score) and then adaptively produce an instance mask at the selected level.

Meanwhile, on an extremely limited amount of fully labeled data, the network often fails to sufficiently represent the instance mask in the mask branch, resulting in rough and noisy mask outputs. In other words, the true-positive proposal does not always lead to a true-positive instance mask in this case. To cope with this limitation, we propose a MaskRefineNet to refine the rough instance mask. The MaskRefineNet takes three input sources, i.e., image, rough mask, and point; the image provides visual information about the target instance, the rough mask is used as the prior knowledge to be refined, and the point information explicitly guides the target instance. Using the richer instructive input sources, MaskRefineNet can be stably trained even with a limited amount of fully labeled data.

To demonstrate the effectiveness of our method, we conduct extensive experiments on the COCO [35] and BDD100K [49] datasets. When training with half of the fully labeled images and the rest of the point labeled images on the COCO dataset (i.e., 50% COCO), we achieve a competitive performance with the fully-supervised performance (38.8% vs. 39.7%). In addition, when using a small amount of fully labeled data, e.g., 5% of COCO data, the proposed method shows much superior performance than the state-of-the-art SSIS method [47] (33.7% vs. 24.9%).

In summary, the contributions of our paper are

- We show that point labels can be leveraged as effective weak supervisions for budget-efficient and high-performance instance segmentation. Further, based on this observation, we establish a new training protocol named Weakly Semi-Supervised Instance Segmentation (WSSIS) with point labels.

- To further boost the quality of the pseudo instance masks when the amount of fully labeled data is extremely limited, we propose the MaskRefineNet, which refines noisy parts of the rough instance masks.

- Extensive experimental results show that the proposed method can achieve competitive performance to those of the fully-supervised models while significantly outperforming the semi-supervised methods.

2. Related Work

2.1. Instance Segmentation

Mask R-CNN [16] is the most widely used method for instance segmentation. They represent an instance as a bounding box and produce the instance mask after pooling each box region. These box-based approaches have many variants, such as [9, 21, 32, 40] and have shown state-of-the-art results. Meanwhile, there is a different type of approach, named point-based approaches [42, 44–46]. They represent an instance as a point and generate the instance mask using the point-encoded mask representation. For example, SOLOv2 [45] extracts point-encoded kernel parameters and generates instance masks with a dynamic convolution scheme. We note that the inference pipeline of these two approaches is the same as shown in Figure 1; they generate proposals in the form of either bounding boxes or points and then produce an instance mask for each proposal. Here, the proposal is indispensable for producing the instance mask.
2.2. Budget-Efficient Instance Segmentation

Instance segmentation requires a huge amount of instance-level segmentation labels. However, the annotation cost of segmentation labels is much higher than other labels. According to seminar works [3, 4], the annotations time is measured on VOC dataset [14] as follows: image-level (20.0 s/img), point (23.3 s/img), bounding box (38.1 s/img), full mask (239.7 s/img). To reduce the annotation cost, weakly-supervised instance segmentation (WSIS) and semi-supervised instance segmentation (SSIS) have been actively studied. The WSIS methods exploit the activation maps generated by self-attention of the network trained with only cost-efficient labels such as image-level [1, 26, 52], point [11, 26, 27], and bounding box [19, 31, 43] labels. Meanwhile, the SSIS methods [47, 51] use a small amount of fully labeled data and an abundant amount of unlabeled data. Utilizing the knowledge of the segmentations learned with the fully labeled data, they generate pseudo instance masks for the unlabeled data. Although they can reduce the annotation cost, their performance is still far behind those of the fully-supervised models.

2.3. Weakly Semi-Supervised Object Detection

There exist some previous attempts to tackle the weakly semi-supervised object detection problem using point labels (WSSOD) [10, 50]. Namely, they use a few box-labeled data and a lot of point-labeled data. Leveraging the point labels, they show improved detection performances compared to the semi-supervised setting. Object detection and instance segmentation tasks share a similar goal: both are object-level recognition tasks. However, we point out that the motivation for leveraging point labels is different. We focus on the fundamental drawback of the instance segmentation network to handle the trade-off between false-negative and false-positive proposals. In contrast, PointDETR [10] leverages the point labels as input queries for single-level feature map inference of DETR [6] architecture, and Group R-CNN [50] employs the point labels to filter and augment proposals with improved positive sample assignments. In addition, we propose the MaskRefineNet for high-fidelity mask refinement to handle the distinct challenge of instance segmentation, which is a pixel-level recognition task.

3. Proposed Method

3.1. Motivation

Existing instance segmentation methods typically adopt a two-step inference process: (1) generate proposals where each instance is represented as bounding box [9, 16, 21, 32] or point [42, 44–46] in proposal branch, and (2) produce a mask for each instance in mask branch. Figure 1 provides an intuition that the performance of the instance segmentation network critically depends on the correctness of proposals at the proposal branch. Thus, improving the proposal branch may lead to a significant performance improvement in semi-supervised instance segmentation (SSIS).

To delve deeper into the problem, we adjust a confidence threshold in the proposal branch to verify the influence of the proposal on the output instance mask as shown in Figure 2. At a low threshold of 0.1 with a larger number of proposals, we obtain more true-positive masks but much more false-positive masks as well (see Figure 2a). The reason is that false-positive proposals (e.g., misclassified or erroneously localized proposals) often lead to noisy mask predictions. Conversely, when we increase the threshold to 0.5, we lose several true-positive masks that were detected at lower thresholds (see Figure 2b). In other words, although the mask branch could represent the instance mask, the absence of the thresholded proposal results in missing instance masks. However, finding an optimal threshold per instance is impractical, and balancing between true-positive and false-positive proposals still remains a challenging problem in SSIS.
Figure 3. Overview of the proposed method. Top: (step 1) training the teacher network and MaskRefineNet with fully labeled data. Bottom: (step 2) under the point label guidance, pseudo labels are generated through the teacher network and further refined using MaskRefineNet. Then, the student network is trained on both the pseudo-labeled data and the fully labeled data.

3.2. Weakly Semi-Supervised Instance Segmentation using Point Labels

From the above observations, we can expect that obtaining correct instance proposals will yield accurate mask representations to improve an SSIS network. To this end, we revisit the point label, which is a one-pixel categorical instance representation cue. The annotation budget for point labels is known as cost-efficient by the literature [3, 4].

Task definition. We propose a new training protocol named Weakly Semi-Supervised Instance Segmentation (WSSIS) using point labels and verify that the budget-friendly point labels can provide effective guidance. The training protocol employs point-labeled data with a small amount of fully labeled data, which yields reduced annotation costs.

Training baseline. Figure 3 shows our proposed baseline of a two-step learning procedure for WSSIS: (1) train a teacher network using only the full labels; (2) train a student network using both the full and pseudo labels generated by the teacher network along with the point labels. Generating high-quality pseudo labels is crucial for WSSIS, so we employ point labels as guidance for filtering the proposals to remain only true-positive proposals. Then, given the filtered proposals, we generate instance masks by exploiting the mask representation of the teacher network. Note that the proposed architecture is a baseline for the proposed task so that one can explore a more advanced training scheme.

FPN head. Most existing instance segmentation approaches [9, 16, 42, 45] adopt Feature Pyramid Network (FPN) [34] architecture for multi-scale instance prediction. Namely, SOLOv2 [45] employs a 5-level feature pyramid (P2−P6), and each pyramid level recognizes instances of particular sizes. When combined with using point labels for sampling proposals, a careful approach to which level to extract proposals based on the size of the instance is demanding. Otherwise, generated instance masks are often noisy as shown in Figure 4 below.

Strategy of using pyramid-level adaptively. Since points do not contain instance size information, we estimate which pyramid level is proper for each point. To this end, we propose a strategy named Adaptive Pyramid-Level Selection, which adaptively selects a pyramid level that is expected to produce the most appropriate instance mask based on the reliability of the network. Namely, we rescale the coordinate of point labels according to the resolution of each level and extract confidence scores for all levels. Then, we generate an instance mask only from the pyramid level with the maximum confidence score, as illustrated in Figure 4. Formally, there is $N$ proposal branches $\{F^p_i\}_{i=1}^N$, and we follow the configuration of FPN [34] with $N=5$. For each point label $(x, y, c)$, where $c$ denotes category id, we extract an instance proposal and confidence score $(P_i, s_i) = F^p_i(x, y, c)$. Regarding the confidence score as the reliability of the prediction, we adaptively select a pyramid level $k$ with the maximum score, $k = \arg\max_{k \in \{1, 2, ..., N\}} s_k$. Finally, at the mask branch $F^m$, we generate a pseudo instance mask $M = \sigma(F^m(P_k))$, where $\sigma$ is sigmoid function.
3.3. Mask Refinement Network

With a sufficient amount of fully labeled data (e.g., using 50% images), the teacher network can afford to generate reasonable pseudo-instance masks given true-positive proposals. However, when the amount of fully labeled data is extremely small (e.g., using only 1% images), the mask representation by the network would produce rough instance masks; it means the true-positive proposal could not ensure that the instance mask is a true-positive.

To handle such a challenging case, we propose a simple yet effective post-hoc mask refinement method named MaskRefineNet. Figure 3 shows that MaskRefineNet can refine the rough mask output from the teacher network based on three input sources, including input image, rough mask, and point information. Specifically, we loosely crop each instance region in the input image, rough mask, and point information, and resize them to $256 \times 256$, then concatenate them together into an input tensor. For the point information, we transform the point label to the form of a heatmap where each point is encoded into a 2D gaussian kernel with a sigma of 6. The effectiveness of the MaskRefineNet can be attributed to two reasons; (1) it leverages the prior knowledge of the teacher network; since MaskRefineNet takes the rough mask predictions from the teacher network as the input, it learns how to calibrate common errors of predictions from the teacher network; (2) it takes guidance from the input point that is likely to provide an accurate target instance cue for recognizing overlapping instances and falsely predicted pixels. Consequently, MaskRefineNet refines the missing & noisy parts and disentangles the crowded target instances in the rough mask as shown in Figure 5 with the help of the point guidance.

4. Experiments

4.1. Datasets

We evaluate our method on the COCO 2017 dataset [35] that contains 118,287 training samples and 5,000 validation samples for 80 common object categories. To validate our method under the WSSIS regime, we randomly sample subsets containing 1%, 2%, 5%, 10%, 20%,~50% of the COCO training dataset. COCO 10% means using 10% of the fully labeled data and the rest of 90% of the point labeled data. We use a centroid point of an instance mask label as a point label. In addition, we conduct experiments on BDD100K dataset [49], which is a large-scale driving scene dataset with diverse scene types and 8 classes. The BDD100K dataset contains 7k mask-labeled images and 67k box-labeled images, and we use the center of the box as the point label for this dataset.

4.2. Implementation Details

We adopt SOLOv2 [45] as the baseline instance segmentation network since it is a point-based and box-free straightforward method. For both teacher and student networks, we use the same ResNet-101 [17] backbone network and follow the default training recipe and network setting as in [45]. For the MaskRefineNet, we adopt the ResNet-101 FPN [34] architecture and produce the output only from the highest resolution pyramid level, P2. We set the batch size of 16, the learning rate of 1e-4 with cosine decay scheduling, dice loss [37], and input size of $256 \times 256$ for training the MaskRefineNet. After training the teacher network, the MaskRefineNet is trained by taking the rough mask outputs from the teacher network. We implement the proposed method using Pytorch [39] and train on 8 V100 GPUs.

Following the labeling budget calculation in [3,4], we estimate the labeling budget for the COCO trainset as follows: Full mask (645.9s/img), Bounding box (127.5s/img), Point (87.9s/img), Image-level (80s/img). Detailed calculation method is described in our supplementary material.

4.3. Experimental Results

We compare the proposed method against two baselines with the same network architecture and optimization strategy. The first is training with only fully labeled data, and the second is training with fully labeled and unlabeled data, which is a semi-supervised setting. For the second baseline, we generate pseudo instance masks for the unlabeled data from the teacher network without any weak labels. As shown in Figure 6, our method achieves remarkable performances on all COCO subsets. Especially, the performance gap between ours and the baselines is notably larger when we use smaller subsets with fully labeled images, e.g., COCO 1% or 5%. Compared to the fully-supervised setting (COCO 100%), our method with COCO 50% shows
a highly competitive result (38.8% vs. 39.7%). Moreover, the qualitative results in Figure 7 show that ours with COCO 5% can properly segment instances of various sizes. These results demonstrate that the cost-efficient point labels can be leveraged as an effective source for instance segmentation.

We also compare against other methods that use various weak labels based on the labeling budget in Table 1. According to the type and the number of labels, we calculate the labeling budget following the aforementioned cost. All methods use the same amount of total training images, so the labeling time of unlabeled data is treated as zero. Compared to the state-of-the-art semi-supervised method, NB [47], ours show superior performances when using the same amount of fully labeled data, especially when using 5% of fully labeled data (33.7% vs. 25.6%). We also achieve higher performance with a lower labeling budget (35.8% with a budget of 196.7 vs. 35.5% with a budget of 265.2). In addition, compared to the state-of-the-art box-supervised method, BoxInst [43], our efficiency is better (33.7% with a budget of 158.5 vs. 33.2% with a budget of 174.5). This result demonstrates the effectiveness of budget-friendly point labels with the proposed method. Furthermore, we emphasize the potential of our method for performance improvement when using more labeling budget.

Also, we conduct experiments on BDD100K dataset. As we increase the amount of point labeled data with a fixed amount of 7k fully labeled data, the performance is gradually improved, as in Table 6. Especially, when leveraging all available point labels (67k), ours can achieve significant performance improvements compared to using only 7k fully labeled data (22.1% → 27.9%).

**4.4. Ablation Study**

We conduct an ablation study of our method on the COCO 10% setting. Unless otherwise specified, we measure the quality of pseudo labels generated by the teacher network using randomly sampled 5,000 images in the rest 90% of COCO data, we name it COCO train5K.

**Effect of Point Labels.** In Table 5, we verify the effectiveness of each weak label candidate (i.e., unlabeled, image-level, and point label) in instance segmentation. For this analysis, we measure the quality of pseudo labels and the performance of the student network on the COCO 2017 validation set. When the unlabeled data is leveraged as a weak label, we should carefully tune the confidence threshold to balance between false-negative and false-positive proposals; the average recall (AP_{100}) and precision (AP) largely vary according to the confidence threshold. It implies that human effort for tuning the threshold is required for target datasets, and this global threshold may not be optimal for every instance. Leveraging the image-level label as a weak label can eliminate the misclassified proposals, boosting the performance from 25.9% to 29.5%. However, the performance gap with the fully-supervised setting is still significant (29.5% vs. 39.0%). When we leverage the point label as a weak label, we filter out the proposals to keep only true-positive proposals, deprecating the requirement of the confidence threshold. It makes a more straightforward and effec-
Table 2. Impact of choosing features adaptively. \( P_n \) denotes \( n \)-th feature pyramid. \( s_i \) is the confidence score of the proposal in \( i \)-th pyramid level.

<table>
<thead>
<tr>
<th>Label Types</th>
<th>COCO train(5k )</th>
<th>COCO val</th>
</tr>
</thead>
<tbody>
<tr>
<td>( U (\tau=0.1) )</td>
<td>( 6.0 )</td>
<td>( 6.0 )</td>
</tr>
<tr>
<td>( U (\tau=0.3) )</td>
<td>( 13.1 )</td>
<td>( 22.9 )</td>
</tr>
<tr>
<td>( U (\tau=0.5) )</td>
<td>( 12.2 )</td>
<td>( 19.3 )</td>
</tr>
<tr>
<td>( T (\tau=0.3) )</td>
<td>( 19.5 )</td>
<td>( 33.1 )</td>
</tr>
<tr>
<td>( \mathcal{P} )</td>
<td>( 28.6 )</td>
<td>( 56.7 )</td>
</tr>
<tr>
<td>( \mathcal{P}^\dagger )</td>
<td>( 39.1 )</td>
<td>( 65.3 )</td>
</tr>
</tbody>
</table>

Table 5. Impact of using point labels. We have the notations: \( U \) (unlabeled data), \( T \) (image-level label), \( \mathcal{P} \) (point label), and \( \mathcal{F} \) (full label). We use COCO train\(5k \) to measure the accuracy of pseudo labels and COCO val to evaluate the baseline network trained with the pseudo labels. \( \tau \) is a confidence threshold in the proposal branch. \( \dagger \) means applying our point-guided MaskRefineNet.

<table>
<thead>
<tr>
<th>Label Types</th>
<th>( \mathcal{F} 7k )</th>
<th>( \mathcal{F} 7k + \mathcal{P} 20k )</th>
<th>( \mathcal{F} 7k + \mathcal{P} 40k )</th>
<th>( \mathcal{F} 7k + \mathcal{P} 67k )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mathcal{F} 7k )</td>
<td>( 22.1 )</td>
<td>( 26.7 )</td>
<td>( 27.3 )</td>
<td>( 27.9 )</td>
</tr>
<tr>
<td>( \mathcal{F} 7k + \mathcal{P} 20k )</td>
<td>( 40.2 )</td>
<td>( 44.4 )</td>
<td>( 44.5 )</td>
<td>( 44.8 )</td>
</tr>
<tr>
<td>( \mathcal{F} 7k + \mathcal{P} 40k )</td>
<td>( 21.2 )</td>
<td>( 27.8 )</td>
<td>( 28.9 )</td>
<td>( 29.2 )</td>
</tr>
<tr>
<td>( \mathcal{F} 7k + \mathcal{P} 67k )</td>
<td>( 40.2 )</td>
<td>( 44.4 )</td>
<td>( 44.5 )</td>
<td>( 44.8 )</td>
</tr>
</tbody>
</table>

Table 6. Quantitative results on BDD100K validation set. We report the AP scores with different training regimes concerning the number of point labels.

Effect of Adaptive Pyramid-Level Selection. We quantitatively analyze the behavior of the FPN in Table 2. When we generate pseudo masks from all pyramid feature maps (P2~P6), we achieve an inferior quality of 10.3% because the outputs from unfit pyramid levels are pretty noisy, as shown in Figure 4. Using our Adaptive Pyramid-Level Selection strategy, we choose one appropriate pyramid level based on the reliability of the network, achieving the improved quality of 28.6%. The result demonstrates that the proposed strategy is highly effective in leveraging the behavior of the FPN structure for generating high-quality pseudo labels. Also, the result of 30.9% when using ground-truth instance size information leaves us room for improvement of our method.

Effect of MaskRefineNet. In Figure 6, we conduct experiments on various subsets without the MaskRefineNet. When the teacher network has enough mask representation ability as in the COCO 50% setting, the improvement of the MaskRefineNet is marginal (38.3% vs. 38.8%). However, the MaskRefineNet yields a considerable performance improvement, especially in the limited number of fully labeled data settings, e.g., COCO 1% (14.3%→24.0%) and COCO 5% (29.0%→33.7%) settings. The result demonstrates that MaskRefineNet is a remarkably effective method to improve the quality of pseudo labels in the limited quantity of fully labeled data conditions.

In addition, we analyze the effect of input sources of the MaskRefineNet in Table 3. Before applying the MaskRefineNet, the quality of pseudo labels is measured as 28.6%. When the MaskrefineNet only takes an image as input, the accuracy of the pseudo labels drastically reduces to 14.8% since the network fails to converge due to the ab-
sence of prior knowledge. When taking the rough mask as an input source, the quality of the pseudo labels improves from 14.8% to 29.7% because the MaskRefineNet takes the knowledge of the teacher network for fast and stable convergence. However, the improvement is still minor compared to the model without MaskRefineNet. When additionally taking the point information as an input source, the quality of pseudo labels dramatically improves to 39.1%. The reason is that the point information is used as a guidance seed for the target instance, helping a more accurate segment of occluded instances and refining the missing predictions in the rough mask, as shown in Figure 5.

5. Conclusion and Limitation

In this paper, we proposed a novel and practical weakly semi-supervised instance segmentation scheme leveraging point labels as weak supervision for cost-efficient and high-performance instance segmentation. We motivated that the main performance bottleneck of modern instance segmentation frameworks arises from the instance proposal extraction. To this end, we proposed a method that can effectively exploit the budget-friendly point labels as weak supervision to resolve the bottleneck. Moreover, we presented the MaskRefineNet to deal with hard learning scenarios where the amount of fully labeled data is extremely limited. Owing to the effectiveness of the proposed method, we can generate high-quality pseudo instance masks, achieving promising instance segmentation results. Despite our remarkable results driven by cost-efficient point labels, we have a limit to straightforwardly exploiting the tremendous amount of unlabeled image pools, such as web-crawling images without any annotations. Our future direction may involve incorporating unlabeled images in our framework to extend its application to semi-supervised learning scenarios.

Acknowledgment. We thank Youngjoon Yoo and ImageVision team members for valuable discussions and feedback. This work was supported by the Engineering Research Center Program through the National Research Foundation of Korea (NRF) funded by the Korean Government MSIT (NRF-2018R1A5A1059921) and Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2019-0-00075, Artificial Intelligence Graduate School Program (KAIST)).
References


