# Pointly-Supervised Instance Segmentation

Bowen Cheng[1*] Omkar Parkhi[2] Alexander Kirillov[2]
[1]UIUC [2]Facebook AI

## Abstract

*We propose an embarrassingly simple point annotation scheme to collect weak supervision for instance segmentation. In addition to bounding boxes, we collect binary labels for a set of points uniformly sampled inside each bounding box. We show that the existing instance segmentation models developed for full mask supervision can be seamlessly trained with point-based supervision collected via our scheme. Remarkably, Mask R-CNN trained on COCO, PASCAL VOC, Cityscapes, and LVIS with only 10 annotated random points per object achieves 94%–98% of its fully-supervised performance, setting a strong baseline for weakly-supervised instance segmentation. The new point annotation scheme is approximately 5 times faster than annotating full object masks, making high-quality instance segmentation more accessible in practice.*

*Inspired by the point-based annotation form, we propose a modification to PointRend instance segmentation module. For each object, the new architecture, called Implicit PointRend, generates parameters for a function that makes the final point-level mask prediction. Implicit PointRend is more straightforward and uses a single point-level mask loss. Our experiments show that the new module is more suitable for the point-based supervision.* [1]

## 1. Introduction

The task of instance segmentation requires an algorithm to locate objects and delineate them with pixel-level binary masks. Manual annotation of object masks for training is significantly more complex and time-consuming than other forms of image annotation like image-level categories [1, 9, 13, 15, 34, 60, 61] or per-object bounding boxes [2, 21, 23, 52]. For example, it takes on average 79.2 seconds per instance to create a polygon-based object mask in COCO [33], whereas a bounding box for an object can be annotated ∼11 times faster in only 7 seconds [41].

Weakly-supervised methods, that use easier to acquire ground truth annotation forms, make instance segmentation more accessible for new categories or scene types, as the efforts required to collect the data are lower. Such models showed a great progress on smaller datasets under a fixed annotation time budget [4, 28], however, they are still far behind fully-supervised methods for large-scale datasets like COCO. The recent BoxInst model [52] outperforms previous weakly-supervised approaches with box supervision but achieves only 85% of its fully-supervised counterpart performance on COCO. A natural question emerges: Is object mask training data necessary to get closer to the fully-supervised performance? And is there an easier to collect annotation form for the instance segmentation task?

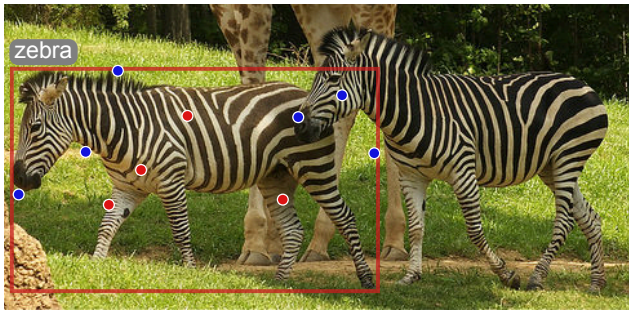


Figure 1. **Our point-based instance annotation scheme.** We collect object bounding boxes with points *randomly sampled* inside each box and annotated as the object (red) or background (blue). Our experiments show that a bounding box and 10 annotated points per instance are approximately 5 times faster to collect than the standard object mask annotation and such ground truth is sufficient to train a standard model like Mask R-CNN [19] to achieve 94%–98% of its fully-supervised performance on various datasets.

Beyond bounding boxes and image-level categories, point clicks and squiggles are the other time-efficient annotation forms most commonly used in interactive segmentation scenarios [5, 29, 30, 35, 58]. Several semantic and instance segmentation methods directly use them for training [4, 31]. In our paper, we present a new instance segmentation annotation scheme that collects both bounding boxes and point-based annotation (see Figure 1). Unlike previous works where points are clicked by annotators, we follow a different process where points are sampled randomly inside an object bounding box and an annotator is asked to classify each point as the object or background. This point-based annotation scheme is simple and we empirically find such annotation to perform well on large-scale datasets.

*Work done during an internship at Facebook AI Research.
[1]Project page: https://bowenc0221.github.io/point-sup

With randomly sampled points, the annotation process can be easily simulated with existing instance segmentation ground truth. This property allows us to test the new annotation scheme and show its efficacy on multiple large-scale datasets without a major annotation effort that is usually a prerequisite for other annotation schemes [4, 31].

The key advantage of the point-based annotation produced by our scheme is in its ability to seamlessly supervise instance segmentation models that directly predict object masks with no changes to their architectures or training pipelines. In our experiments, we train Mask R-CNN [19], PointRend [25], and CondInst [51] with this point-based supervision. For each object, the standard mask loss is computed for ground truth points by interpolating mask predictions at these points. We show that the point-based annotation obtained via our scheme is applicable across different categories and scene types by simulating it on COCO [32], PASCAL VOC [12], LVIS [17], and Cityscapes [11] datasets. Mask R-CNN trained with only 10 annotated points per object achieve 94%–98% of its fully-supervised performance on these datasets. In addition, we propose a simple point-based data augmentation strategy and explore self-training paradigm for point-based supervision to show that the gap to the full supervision can be further reduced. Finally, we show that point-based pre-training is matching mask-based in a transfer learning setup.

To analyze the performance/annotation time trade-off of the new annotation scheme, we created a simple labeling tool and measured that a trained human annotator classifies a point in 0.9 seconds on average. This means, together with a bounding box annotation that can be done in 7 seconds via extreme point clicking [41], the total annotation time for 10 points per object is 16 seconds ($7 + 10 \cdot 0.9$). This is 5 times faster than polygon-based mask annotation in COCO. For a new dataset, the point-based annotation from our scheme allows the standard models to achieve performance close to full supervision at a fraction of the full data collection time.

In our experiments, we observe that PointRend [25] supervised with point-based annotations performs on par with Mask R-CNN [19] supervised in the same way, whereas with the standard full mask supervision it performs better. Inspired by this finding, we propose **Implicit PointRend**. Instead of a coarse mask prediction used in PointRend to provide region-level context to distinguish objects, for each object Implicit PointRend generates different parameters for a function that makes the final point-wise mask prediction. The new model is simpler than PointRend: (1) it does not require an importance point sampling during training and (2) it uses a single point-level mask loss instead of two mask losses. Implicit PointRend can be trained directly with point supervision without any intermediate prediction interpolation steps. Our experiments demonstrate that the new module outperforms PointRend with point supervision.

## 2. Related Works

**Fully-supervised instance segmentation** models that currently dominate popular COCO benchmark [33] predict object masks directly. Mask R-CNN-based methods [19] make such predictions on a region level from features that correspond to detected bounding boxes, whereas methods like YOLACT++ [6] or CondInst [51] make image-level mask predictions. In both cases ground truth masks are directly used for supervision. Alternatively, bottom-up segmentation methods output masks indirectly, forming them from a set of predicted cues such as instance boundaries [24], energy levels [3], and offset vectors to center points [8]. We show that methods that use direct mask supervision can be trained with point-based annotation from our annotation scheme without significant modifications.

**Weakly-supervised instance segmentation** methods usually use image-level category labels or bounding boxes. With image-level labels, such methods rely on segmentation proposals [43, 53] either re-ranking them [60] or generating pseudo-ground truth [1, 2, 61]. These methods show promising results on smaller datasets, but no competitive results have been shown on a large-scale dataset like COCO [33]. With bounding box supervision, SDI [23] proposes a multi-stage training procedure that generates pseudo-ground truth with GrabCut [48]. BBTP [21] trains Mask R-CNN [19] using a bounding box tightness prior. Recently proposed BoxInst [52] supervises the mask branch of CondInst [51] with a projection loss that forces horizontal/vertical lines inside bounding boxes to predict at least one foreground pixel and an affinity loss that forces pixels with similar colors to have the same label. BoxInst outperforms previous approaches, achieving an impressive 85% of fully-supervised performance on COCO [33]. In our work, we show that additional point-based supervision allows us to get much closer to the quality of fully-supervised models. In addition, our experiments suggest that our point-based annotation scheme has a better performance/annotation time trade-off for a wide variety of annotation efforts.

**Point-based supervision** has been studied in a variety of tasks, including action localization [37, 38], object detection [42, 46], object counting [27], and semantic segmentation [4, 44]. For instance segmentation, point clicks are often used in interactive pipelines [5, 29, 30, 35, 58]. While very powerful, such systems are more complex than our scheme, as they are trained with full supervision and require to repeatedly run an inference model during annotation. Laradji *et al*. [28] present a proposal-based instance segmentation method that uses a single point per instance as supervision. In contrast, we use multiple points and a bounding box annotation together. Furthermore, instead of collecting clicks, we randomly generate point locations and ask annotators to classify points as object or background.

## 3. Pointly-Supervised Instance Segmentation

### 3.1. Annotation format and collection

The format of point-based annotation collected via the new scheme is conceptually simple. In addition to the standard bounding box annotation, we assume $N$ points within each object bounding box to be labeled as the object *vs*. background (Figure 1). We refer to this annotation as $\mathcal{P}_N$.

**Random points instead of clicks.** Previous point-based annotation schemes collected annotators clicks [4, 27, 28, 38, 42, 44]. Such strategy, however, can lead to data with low variability as human clicks are often correlated [10, 14]. Bearman *et al*. [4] confirm this by showing that training with randomly located ground truth points is superior for their semantic segmentation model than training with clicks. Following these findings we randomly sample point locations within each object bounding box and an annotator is asked to classify each point as the object or background.

**Collection and simulation.** The annotation scheme for collecting $\mathcal{P}_N$ is straightforward. First, bounding boxes for objects are collected using any off-the-shelf solution [18, 26, 41]. Next, for each object, $N$ random point locations within its bounding box are sequentially presented to an annotator for a binary object/background classification given the bounding box and object category label.

Unlike click-based schemes where human is involved in the point selection process, the new annotation scheme can be seamlessly simulated for datasets that have full instance segmentation ground truth by generating a bounding box for each instance mask and classifying randomly sampled points/pixels inside the box based on the corresponding positions in the ground truth mask. By simulating the annotation from our scheme, we are able to ablate its design and verify our results on a diverse set of large-scale datasets without time consuming data collection efforts.

**Annotation time.** To measure the annotation time, we build a simple tool for our point-based annotation scheme (see the supplement for details). With this tool we annotated 100 objects from the COCO [33] and LVISv1.0 [17] datasets, by an annotating contractor company with human annotators similar to Amazon Mechanical Turk (AMT), and found that it takes 0.9 seconds on average to classify a point. Note, that a bounding box can be annotated in 7 seconds using extreme points method [41]. Thus, the total annotation time of $\mathcal{P}_N$ per object is 7 (bounding box) $+ 0.9 \cdot N$ (binary classification for $N$ points) seconds.

In our experiments we found that $\mathcal{P}_{10}$ provides a good trade-off between annotation time (16 seconds per object) and performance (94%-98% of fully-supervised performance). Such annotation is $\sim$5 times faster to label than a polygon-based instance mask in COCO that takes on average 79.2 seconds for a spotted object [33].

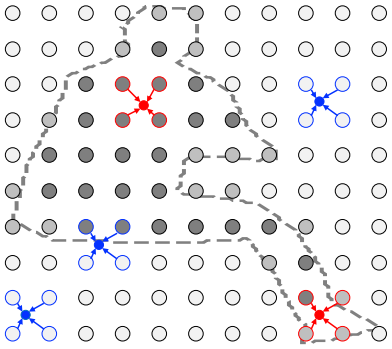

Figure 2. **Illustration of point supervision.** For a $10 \times 10$ prediction mask on the regular grid (darker color indicates foreground prediction), we get predictions at the exact locations of ground truth points (red and blue indicate foreground and background ground truth points respectively) with bilinear interpolation. Note, that the object contour line is only for the purpose of illustration.

**Annotation quality.** Collected point labels have 90% agreement with COCO ground truth masks. In most cases, the misclassification occurs very close to object boundaries or in regions where the ground truth polygons are imprecise. The agreement is around 95% when more accurate LVIS ground truth masks are used. Our experiments show no significant drop in performance if point labels are simulated with 95% accuracy. In real world scenarios, such errors are usually fixed with a verification step [17, 26, 33].

### 3.2. Training with points

The key advantage of such point-based annotation is its ability to seamlessly supervise off-the-shelf instance segmentation models that make mask predictions on a regular grid (*e.g*., a $28 \times 28$ RoI prediction of Mask R-CNN [19] or an image-level per-pixel prediction of CondInst [51]). In a fully-supervised setting, these models are trained by extracting a matching regular grid of labels from ground truth masks. In contrast, with point supervision, we approximate predictions in the locations of ground truth points from the prediction on the grid using bilinear interpolation (see Figure 2). Once we have predictions and ground truth labels at the same points, a loss can be applied in the same way as with full supervision and its gradients will be propagated through bilinear interpolation. Note, that such supervision does not require any changes to the architecture. In our experiments we use cross-entropy loss on points, however other losses like dice loss [39] can be used as well.

For region-based models, some ground truth points may lay outside of predicted boxes and we choose to ignore such points during training. Contrarily, for models that yield image-level masks, additional background points can be sampled from the outside of ground truth boxes. However, in this paper we study the most basic setup where only $N$ annotated ground truth points are used to train all models.

**Data augmentation** during training is a crucial component of modern segmentation models. The point-based annotation is compatible with all common augmentations like scale jitter, crop, or horizontal flip of an input image. In our experiments, we observe that the gap in performance between point and full mask supervision is larger for longer training (3× schedule [55]) and models with higher-capacity backbones (*e.g.*, ResNeXt-101 [57]). We hypothesize that this is caused by a reduced variability of training data when only a few points are available and propose a extremely simple *point-based* data augmentation strategy: instead of using all available ground truth points for a box at each iteration, we subsample half of all available points at every training iteration (5 points for $\mathcal{P}_{10}$ ground truth). In the next section we show that our augmentation strategy improves performance for higher-capacity models.

### 3.3. Experiments with point supervision

In this section we first ablate the design of the new annotation scheme on the COCO dataset [33] using Mask R-CNN [19]. Then, we demonstrate the effectiveness of the point-based supervision across 4 different datasets and show that it is applicable to a diverse set of instance segmentation models. Finally, we explore the annotation time and performance trade-off of the new annotation scheme.

**Implementation details.** We use Mask R-CNN [19] with a ResNet-50-FPN [20, 32] backbone and the default training schedules and parameters from Detectron2 [55] for each dataset (3× schedule is used for COCO [33]). Apart from the mask branch loss, there are no other differences between full mask and the point-based supervision in our experiments. Unless stated otherwise, we do not use point-based data augmentations described in section 3.2.

**Datasets.** The main dataset in our experiments is COCO [33] which has 118k training and 5k validation images with 80 common categories. We also conduct experiments using: PASCAL VOC dataset [12] which is smaller than COCO with ∼10k images and 20 categories; Cityscapes [11] ego-centric street-scene dataset that has 2975 train and 500 validation high-resolution images with high-quality instance-level annotations for 9 categories; LVISv1.0 [17] that uses the same set of images as COCO but has more than 1000 categories annotated in a federated fashion. We use the standard evaluation metrics: AP50 [12] for PASCAL VOC and AP [33] for all other datasets.

For all datasets in our experiments, we simulate point-based ground truth by randomly sampling pixels inside each ground truth bounding box and selecting their labels based on the corresponding ground truth mask.

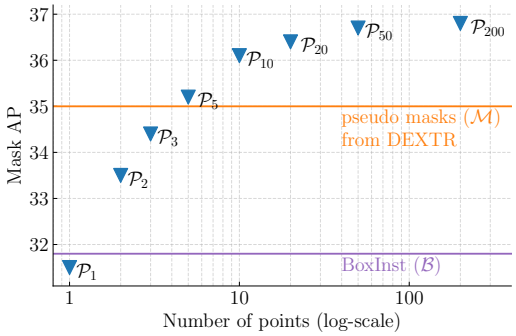


Figure 3. **Training with a different number of points.** Mask R-CNN [19] with a ResNet-50-FPN backbone trained on COCO with as few as 10 labeled points per instance ($\mathcal{P}_{10}$) achieves 36.1 mask AP with diminishing returns from more labeled points. We provide two baselines: (1) BoxInst [52], the best current model that uses box supervision ($\mathcal{B}$) only, and (2) Mask R-CNN trained with pseudo ground truth generated by an interactive segmentation method DEXTR [35] that uses extreme clicks as its input.

#### 3.3.1 Ablation of the annotation design

**Number of points.** In Figure 3 we demonstrate the performance of a Mask R-CNN trained with a varying number of labeled points per instance. The performance rapidly improves with the number of annotated points going up to tens with diminishing returns thereafter. While 20 points ($\mathcal{P}_{20}$) is approximately 0.3 AP better than 10 point supervision ($\mathcal{P}_{10}$), it takes twice as long to annotate. In what follows, we use the 10 points supervision which is ∼5 times faster than polygon-based mask annotation. Two baselines in Figure 3 are: (1) BoxInst [52], the best current model that uses box supervision ($\mathcal{B}$) only, and (2) Mask R-CNN trained with masks generated by DEXTR [35], an interactive segmentation method that predicts a mask from 4 extreme clicks. For this baseline, DEXTR is trained in PASCAL VOC [12] and we simulated extreme clicks on COCO as its input.

**Sensitivity to point locations.** We generate 5 different simulated versions of $\mathcal{P}_{10}$ ground truth annotation for the COCO `train2017` set using different random seeds for the uniform point sampling procedure. Training the same Mask R-CNN [19] model with these dataset versions, we observe very low 0.1 AP variation on the COCO `val2017` set. This result suggests that the point-based annotation from our annotation scheme is very robust with respect to the locations of sampled points. In what follows we conduct experiments with a single set of sampled points per dataset.

**Sensitivity to the annotation quality.** We study the sensitivity to the quality of point-based annotation by changing correct object/background labels for random 5% of points in a simulated ground truth $\mathcal{P}_{10}$ point annotation for COCO. The performance of a Mask R-CNN model trained on the tampered data is only 0.2 AP worse than the model trained on the clean data (35.9 AP *vs.* 36.1 AP). In a more chal-

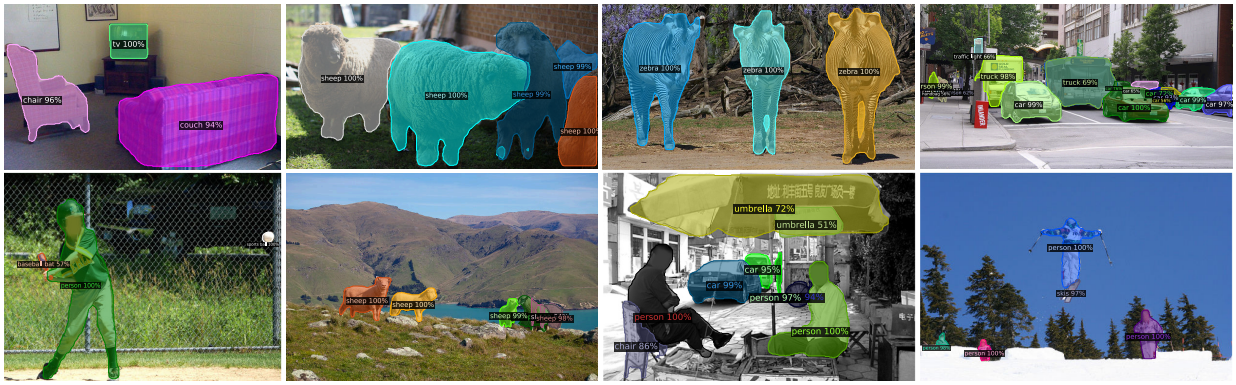

Figure 4. **Mask R-CNN trained with 10 points per object ($\mathcal{P}_{10}$)** on COCO. The model uses a ResNet-50-FPN [19, 32] backbone and is trained with 3× schedule [55] (36.1 AP). We show predictions with confidence scores greater than 0.5.

| supervision | AP |
|---|---|
| $\mathcal{M}$ | 37.2 |
| $\mathcal{P}_{10}$ | 36.1 (97%) |

(a) COCO `val2017` [33].

| supervision | $AP_{50}$ |
|---|---|
| $\mathcal{M}$ | 66.3 |
| $\mathcal{P}_{10}$ | 64.2 (97%) |

(b) PASCAL VOC `val` [12].

| supervision | AP |
|---|---|
| $\mathcal{M}$ | 32.7 |
| $\mathcal{P}_{10}$ | 30.7 (94%) |

(c) Cityscapes `val` [11].

| supervision | AP |
|---|---|
| $\mathcal{M}$ | 22.8 |
| $\mathcal{P}_{10}$ | 21.5 (94%) |

(d) LVISv1.0 `val` [17].

Table 1. **Mask R-CNN with full mask ($\mathcal{M}$) *vs.* new point supervision ($\mathcal{P}_{10}$).** For each object only 10 labeled points are used. For all four datasets with different properties, pointly-supervised models are within 2 AP (94% – 97%) from their fully-supervised counterparts.

| supervision | point aug. | R50 AP | R101 AP | X101 AP |
|---|---|---|---|---|
| $\mathcal{M}$ | - | 37.2 | 38.6 | 39.5 |
| $\mathcal{P}_{10}$ | | 36.1 (97%) | 37.8 (98%) | 38.1 (96%) |
| | ✓ | 36.0 (97%) | 37.8 (98%) | 38.5 (97%) |

Table 2. **Mask R-CNN with higher-capacity backbones and point-based augmentation** on COCO. We use ResNet-50 (R50), ResNet-101 (R101), and ResNeXt101-32×8 (X101) backbones [20, 57] with FPN [32]. The simple point-based data augmentation (*point aug.*) proposed in section 3.2 improves performance (+0.4 AP) for the higher-capacity X101 model.

| supervis. | AP |
|---|---|
| $\mathcal{M}$ | 37.2 |
| $\mathcal{P}_{10}$ | 36.1 (97%) |

(a) Mask R-CNN [19].

| supervis. | AP |
|---|---|
| $\mathcal{M}$ | 37.5 |
| $\mathcal{P}^*_{10}$ | 35.7 (95%) |

(b) CondInst [51].

| supervis. | AP |
|---|---|
| $\mathcal{M}$ | 38.3 |
| $\mathcal{P}^*_{10}$ | 35.7 (93%) |

(c) PointRend [25].

Table 3. **Different models trained with point supervision** on COCO. All models use a ResNet-50-FPN backbone [20, 32]. "∗": We train CondInst and PointRend models with point-based augmentation which improves their performance. While all three models achieve similar performance with point supervision, we observe that the gap between mask and point supervision is the largest for PointRend. We explore PointRend performance and propose an improved version of the module in the next section.

lenging setup, instead of random points, we set wrong labels to 5% of points that are closest to object boundaries. In this case, the drop in performance is slightly larger – 0.4 AP (35.7 AP *vs.* 36.1 AP). Our experiment shows that such point-based annotation does not require perfectly accurate point labeling procedure which reduces the need for additional verification steps in a real-world annotation pipeline.

### 3.3.2 Main results

In Table 1 we compare 10 points supervision ($\mathcal{P}_{10}$) to full mask supervision ($\mathcal{M}$) for a Mask R-CNN model with a ResNet-50-FPN [20, 32] backbone on four different datasets. Without any modifications to either the architecture or the default training algorithm/hyper-parameters set in Detectron2 [55], Mask R-CNN trained with $\mathcal{P}_{10}$ achieves 94% – 97% of the fully-supervised ($\mathcal{M}$) model performance. This result shows that such point supervision is widely applicable to datasets at different scales (*e.g.*, from 20 categories in PASCAL VOC [12] to more than 1,000 categories in LVISv1.0 [17]) and in different domains (common objects of COCO [33] or street scenes of Cityscapes [11]). In Figure 4, we visualize predictions of a Mask R-CNN model supervised with points on COCO.

**Higher-capacity backbones and point-based data augmentation.** In Table 2 we compare full and 10 points supervision for Mask R-CNN with different backbones on COCO. The simple point-based augmentation strategy described in section 3.2 does not significantly change the performance of the smaller model. However, for higher-capacity ResNeXt-based model, the point-based augmentation effectively improves performance by 0.4 AP.

**Different models.** Point-based supervision can be applied to a diverse set of models that use a per-pixel mask loss. In our experiments we use two methods in addition to Mask R-CNN: CondInst [51] that makes image-level predictions without an RoI pooling operation[2] and PointRend [25] that

[2] As described in section 3.2, for CondInst [51] we use per-point cross entropy for given ground truth points *without interpolation* and do not su-

| supervis. | ImageNet | COCO-$\mathcal{M}$ |
|---|---|---|
| $\mathcal{M}$ | 66.3 | 74.5 |
| $\mathcal{P}_{10}$ | 64.6 (97%) | 74.0 (99%) |

(a) Mask pre-training.

| pre-training | $AP_{50}$ ($\mathcal{M}$) |
|---|---|
| COCO-$\mathcal{M}$ | 74.5 |
| COCO-$\mathcal{P}_{10}$ | 74.5 (100%) |

(b) Point pre-training.

Table 4. **Pre-training with different supervision** and fine-tuning on PASCAL VOC. All models use a ResNet-50-FPN backbone [20, 32]. (a) Pre-training on datasets with mask annotation (*e.g.*, COCO-$\mathcal{M}$) closes the gap between point-based supervision and full supervision on downstream dataset. (b) Point pre-training (COCO-$\mathcal{P}_{10}$) is as effective as mask pre-training (COCO-$\mathcal{M}$).

makes point-level prediction to iteratively achieve a high resolution. We use the same ResNet-50-FPN [20, 32] backbone for all models and report results on COCO. Unlike Mask R-CNN, both CondInst and PointRend benefit from point-based augmentation even with a smaller backbone; thus, we report their results using the augmentation. In Table 3 we observe that all three models achieve similar performance with point supervision. However, the gap between mask and point supervision for PointRend is significantly larger. The coarse mask head in PointRend makes prediction with low 7×7 resolution making direct point-based training inaccurate. Inspired by this observation, in section 4 we present a new streamlined version of PointRend module with a single mask loss which can be trained with either mask or point supervision.

**Self-training with points.** The gap between point and full mask supervision can be further decreased using self-training paradigm [7, 47, 49, 56, 59, 62]. After training a ResNet-50-FPN based Mask R-CNN with 10 points supervision on COCO (36.1 AP), we collect pseudo-ground truth masks by running inference on COCO train data with ground truth boxes. Mask R-CNN trained on these pseudo-ground truth masks without any changes in the training recipe, achieves 36.7 AP (+0.6 AP) or 98% of fully-supervised Mask R-CNN trained with 6× schedule that matches the training time of our self-training setup.

**Transfer learning with points.** Pre-training on a large-scale dataset (*e.g.*, COCO) followed by fine-tuning is a widely used paradigm to improve performance on a smaller dataset (*e.g.*, PASCAL VOC). We study point-based annotation under two realistic scenarios: (1) if we have access to mask annotation in the large-scale dataset, Table 4a shows mask pre-training closes the gap between point-based supervision and full supervision on downstream dataset; (2) our point-based annotation scheme is enough to collect a larger dataset for pre-training purpose, as Table 4b suggests that point pre-training is as effective as mask pre-training.

pervise predictions outside of the ground truth boxes. Therefore, we need to filter out any mask prediction outside of predicted boxes. A more sophisticated point selection strategy can be used to avoid it, however, this is not the focus of our paper.

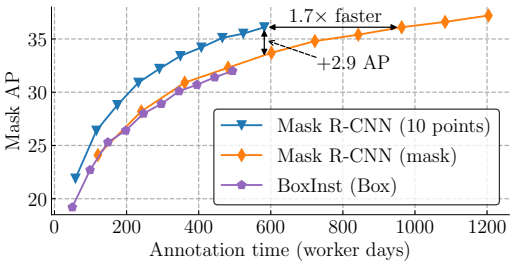


Figure 5. **AP *vs.* annotation time for different supervision types** on COCO `val2017`. To match annotation times between different supervision forms, we train a Mask R-CNN model using from 10% to 100% of COCO `train2017`. Observe that Mask R-CNN trained with the new point-based scheme significantly outperforms models trained with both full mask supervision and weak bounding box supervision under the same computation budget.

### 3.3.3 Annotation time and performance trade-off.

We compare the new point-based annotation scheme with other annotation collection processes for instance segmentation under *the same annotation budget* which we measure as the time required to label training data. For this comparison we use identical Mask R-CNN [19] models for full mask and point supervision. Whereas, for bounding box supervision we use BoxInst [52], a recently proposed instance segmentation model that shows the best performance on COCO among existing weakly-supervised methods with bounding box supervision. Note, that BoxInst is based on CondInst [51] which performs on par or better than Mask R-CNN with full mask and point supervision. For all models we use the standard ResNet-50-FPN backbone [20, 32].

**COCO annotation timings.** An annotation collection for instance segmentation is a multi-stage process. For example, in the COCO annotation pipeline [33], annotators first spot and categorize objects by pointing at them, and then the spotted objects are annotated with polygon-based masks. As discussed in section 3.1, for a spotted object in COCO, it takes on average 7 seconds to annotate its bounding box [41], 16 seconds to annotate the box and 10 points inside, and 79.2 seconds for polygon-based mask annotation [33]. For each annotation form we approximate the total time to label COCO as the time required for the categorization and spotting stages (reported by COCO creators [32], see the supplement) plus the time required to annotate all spotted objects with the corresponding annotation form. In Figure 5 we match the annotation times between different supervision forms by training a Mask R-CNN model using from 10% to 100% of COCO `train2017` data. Our analysis shows that under the same annotation budget, Mask R-CNN trained with points significantly outperforms models trained with the other supervision forms.

## 4. Implicit PointRend Model

Intuitively, point-based annotation from the new scheme should suite well to the PointRend module [25] that yields a mask for a detected object using point-wise predictions. However, as we observed in Table 3, the gap between full mask and point supervision is larger for PointRend than for the other studied methods. While PointRend-based model trained with full mask supervision outperforms Mask R-CNN [19], it performs worse than the baseline with point supervision. We hypothesize that the gap in performance is due to the coarse mask head of the standard PointRend module that outputs a $7 \times 7$ mask prediction and has its own mask loss. With such low resolution if two ground truth points are close to each other but have different labels, then this head does not get a reliable training signal.

Inspired by this observation, we propose a simplified version of the PointRend module which we name Implicit PointRend. For each detected object, instead of a coarse mask prediction, the new architecture predicts parameters for a point head function that can make a point-wise object mask prediction for any point given its position and corresponding image features. The new module has a single point-level mask loss which simplifies its implementation in comparison with PointRend. In what follows we describe the new design in detail and compare it with PointRend using both full mask and point supervision.

### 4.1. Point-wise representation and point head

**PointRend [25]** constructs feature representation for a point concatenating two feature types: (1) a fine-grained point representation extracted from an image-level feature map (e.g. an FPN [32] level feature map) in the exact position of the point via bilinear interpolation and (2) a coarse mask prediction for the point which provides region-specific information. The coarse mask prediction is made by a separate head that takes in RoI features and return a low resolution ($7 \times 7$) mask prediction. Given this point representation, a small MLP network, called point head, makes mask prediction for the exact point location. Note, that in the PointRend model, the parameters of the point head are shared across all points and detected instances.

**Implicit PointRend** uses the point head to make point-wise mask predictions as well. However, instead of relying on coarse mask predictions to distinguish different instances, the new model generates different parameters of the point head for each instance similarly to the dynamic head approach of CondInst [51]. The parameters of the point head implicitly represent the mask of an objects and resembles implicit functions used by 3D community [16, 22, 36]. To make a mask prediction the point head takes in the coordinate of the point relative to the bounding box it belongs to. In addition, we use the same fine-grained point features as in the original PointRend design. Note, that the new Implicit PointRend module does not require a coarse mask prediction and, therefore, can be trained with a single mask loss applied to the output of the point head. We compare overall design of the new module with PointRend in Figure 6.

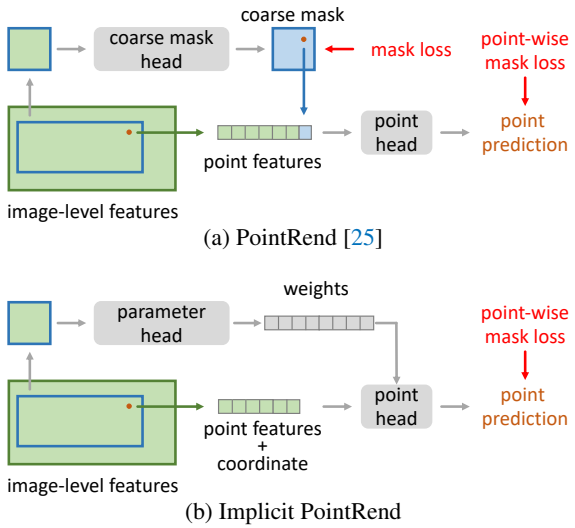


Figure 6. **PointRend [25] *vs*. Implicit PointRend architectures.** Instead of a coarse mask prediction used in PointRend to provide region-level context to distinguish objects, Implicit PointRend generates different parameters of the point head for each detected object. The point head makes prediction at any location independently taking in fine-grained point features and the information about the coordinate of this location relative to the bounding box. The same subdivision mask rendering algorithm [25] is used in both to output high resolution mask with point-wise predictions.

### 4.2. Point selection for inference and training

**Inference.** Implicit PointRend follows the same point selection strategy as PointRend during inference, *i.e*. adaptive subdivision [54]. We start from predictions generated by our point head using a $28 \times 28$ uniform grid of points. Then, we gradually upsample it by a factor of 2 to $224 \times 224$ resolution. In each subdivision step, we select $N$ most uncertain points after interpolation and replace them with the predictions from the point head at these locations. Following the original PointRend implementation, we set $N$ to $28^2$.

**Mask supervision ($\mathcal{M}$) training.** During training, PointRend selects more points from the uncertain regions of the coarse mask prediction via an importance sampling strategy. In our experiments we find that the Implicit PointRend model performs on par with PointRend while using a much simpler uniform point sampling strategy.

**Point supervision ($\mathcal{P}_{10}$) training.** Implicit PointRend is naturally suited for point supervision, where we simply select points with ground truth annotation. Similarly to Mask R-CNN, we ignore points outside predicted boxes.

### 4.3. Implementation details

We set all hyper-parameters following the original PointRend setup [25]. The point head is an MLP with 3 hidden layers of 256 channels, ReLU activations [40] in hidden layers and the sigmoid activation applied to its output. To make a fair comparison, the head that generates parameters for the point head has the same architecture as the coarse mask head of PointRend [25]. We observe however that Implicit PointRend performs as well with a smaller parameter head such as the standard box head architecture [45]. We expect that a better design can bring further improvement.

As the input for the point head we use fine-grained features extracted from `p2` level of FPN [32] that has 256 channel dimension. We use the random Fourier positional encoding [50] to represent the point $(x, y)$ location relative to the center of the bounding box. We found that the model that uses positional encoding performs better than the one that uses $(x, y)$ coordinates directly. Please see the supplement for more detail and ablation experiments.

Implicit PointRend uses a single per-point mask loss (binary cross entropy) applied to the outputs of the point head. In addition, following the common setup for implicit function models [50], we add $l_2$ loss on predicted parameters of the point head to avoid predicted parameters become unbounded. This loss plays the role of weight decay which is otherwise absent for the dynamic parameters. In our experiments, we set the $l_2$ loss weight to 1e-5.

### 4.4. Experimental evaluation

To evaluate Implicit PointRend, we use COCO [33] and 3× schedule [55] to avoid underfitting. All hyper-parameters for the Implicit PointRend module are the same as in PointRend [25]. In all experiments we attach the modules as a mask head to Faster R-CNN [45]. Unless specified, we use a ResNet-50-FPN [20, 32] backbone.

In our experiments, we observe that Implicit PointRend performance significantly increases even for smaller models if the point sub-sampling augmentation described in section 3.2 is used (see the supplement for an ablation). We did not observe the same behavior with the standard PointRend module and hypothesize, that the implicit representation of an object mask is more prone to overfitting than the coarse mask-based representation due to its higher capacity. To make a fair comparison, in this section we use the point-based augmentation for all models supervised with points.

**Main results.** We compare the new Implicit PointRend and PointRend in Table 5 with both full mask and point supervision. Unlike PointRend, our new module does not use any importance point sampling strategy and has only one mask loss. The more straightforward Implicit PointRend module performs on par with PointRend trained with full mask supervision (AP ($\mathcal{M}$)). Moreover, the new module significantly outperforms the baseline in case of point supervision (AP ($\mathcal{P}_{10}$)). This observation underscores unique challenges of the point supervision and suggests that new model designs may be needed to fully uncover the potential of the point-based annotation form.

| method | AP ($\mathcal{M}$) | AP ($\mathcal{P}_{10}$) |
|---|---|---|
| PointRend [25] | 38.3 | 35.7 |
| Implicit PointRend | **38.5** (+0.2) | **36.9** (+1.2) |

Table 5. **PointRend *vs*. Implicit PointRend** with mask ($\mathcal{M}$) and our point supervision ($\mathcal{P}_{10}$) on COCO `val2017`. A ResNet-50-FPN [20, 32] backbone is used for both models. While Implicit PointRend performs on par with the standard PointRend with the full mask supervision (AP ($\mathcal{M}$)), it significantly outperforms the baseline in case of the point supervision (AP ($\mathcal{P}_{10}$)).

| method | supervision | R50 AP | R101 AP | X101 AP |
|---|---|---|---|---|
| Mask R-CNN | $\mathcal{M}$ | **37.2** | **38.6** | 39.5 |
| Mask R-CNN | $\mathcal{P}_{10}$ | 36.0 (-1.2) | 37.8 (-0.8) | 38.5 (-1.0) |
| Implicit PointRend | $\mathcal{P}_{10}$ | 36.9 (-0.3) | 38.5 (-0.1) | **39.7** (+0.2) |

Table 6. Implicit PointRend performance with 10 points supervision achieves the performance of the standard Mask R-CNN model with full mask supervision on COCO `train2017`. We use R50, R101, and X101 backbones [20, 57] with FPN [32].

Similar to the other methods studied in section 3.3, Implicit PointRend-based model supervised with points achieves 96% of its full mask supervision performance. In Table 6 we compare Implicit PointRend trained with points and the standard Mask R-CNN using the backbones of different capacities. As expected from a stronger method, Implicit PointRend significantly outperforms the standard Mask R-CNN when both trained with points. Moreover, we find that Implicit PointRend trained with only 10 labeled points per objects achieves the performance of the fully-supervised Mask R-CNN. Given that Mask R-CNN is still actively used in many real-world applications, this result suggests broad applicability of the new annotation form in scenarios with a constrained annotation budget.

In addition, in the supplement we report Implicit PointRend performance on COCO with mask supervision.

## 5. Conclusion

We present the new point annotation scheme for instance segmentation which labels only a bounding box and several random points annotated per instance. Unlike many other weak annotation forms, the resulting point-based supervision can be seamlessly applied to existing instance segmentation models without any modification to their architecture or training algorithm. The new annotation scheme provides the best trade-off between annotation time and accuracy among other schemes for instance segmentation. We further propose a simple but effective module named Implicit PointRend which tackles the unique challenges of point supervision with implicit mask representation.

## References


[1] Jiwoon Ahn, Sunghyun Cho, and Suha Kwak. Weakly supervised learning of instance segmentation with inter-pixel relations. In *CVPR*, 2019.

[2] Aditya Arun, CV Jawahar, and M Pawan Kumar. Weakly supervised instance segmentation by learning annotation consistent instances. In *ECCV*, 2020.

[3] Min Bai and Raquel Urtasun. Deep watershed transform for instance segmentation. In *CVPR*, 2017.

[4] Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. What's the point: Semantic segmentation with point supervision. In *ECCV*, 2016.

[5] Rodrigo Benenson, Stefan Popov, and Vittorio Ferrari. Large-scale interactive object segmentation with human annotators. In *CVPR*, 2019.

[6] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. YOLACT++: Better real-time instance segmentation. *PAMI*, 2020.

[7] Liang-Chieh Chen, Raphael Gontijo Lopes, Bowen Cheng, Maxwell D Collins, Ekin D Cubuk, Barret Zoph, Hartwig Adam, and Jonathon Shlens. Naive-student: Leveraging semi-supervised learning in video sequences for urban scene segmentation. In *ECCV*, 2020.

[8] Bowen Cheng, Maxwell D Collins, Yukun Zhu, Ting Liu, Thomas S Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-DeepLab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In *CVPR*, 2020.

[9] Hisham Cholakkal, Guolei Sun, Fahad Shahbaz Khan, and Ling Shao. Object counting and instance segmentation with image-level supervision. In *CVPR*, 2019.

[10] Herbert H Clark. Coordinating with each other in a material world. *Discourse studies*, 2005.

[11] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

[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*, 2015.

[13] Ruochen Fan, Qibin Hou, Ming-Ming Cheng, Gang Yu, Ralph R Martin, and Shi-Min Hu. Associating inter-image salient instances for weakly supervised semantic segmentation. In *ECCV*, 2018.

[14] Chaz Firestone and Brian J Scholl. "please tap the shape, anywhere you like" shape skeletons in human vision revealed by an exceedingly simple measure. *Psychological science*, 2014.

[15] 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.

[16] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas Funkhouser. Local deep implicit functions for 3D shape. In *CVPR*, 2020.

[17] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *ICCV*, 2019.

[18] Michael Gygli and Vittorio Ferrari. Efficient object annotation via speaking and pointing. *IJCV*, 2019.

[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. Deep residual learning for image recognition. In *CVPR*, 2016.

[21] Cheng-Chun Hsu, Kuang-Jui Hsu, Chung-Chi Tsai, Yen-Yu Lin, and Yung-Yu Chuang. Weakly supervised instance segmentation using the bounding box tightness prior. In *NIPS*, 2019.

[22] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, and Thomas Funkhouser. Local implicit grid representations for 3D scenes. In *CVPR*, 2020.

[23] Anna Khoreva, Rodrigo Benenson, Jan Hosang, Matthias Hein, and Bernt Schiele. Simple does it: Weakly supervised instance and semantic segmentation. In *CVPR*, 2017.

[24] Alexander Kirillov, Evgeny Levinkov, Bjoern Andres, Bogdan Savchynskyy, and Carsten Rother. InstanceCut: from edges to instances with multicut. In *CVPR*, 2017.

[25] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. PointRend: Image segmentation as rendering. In *CVPR*, 2020.

[26] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *IJCV*, 2020.

[27] Issam H Laradji, Negar Rostamzadeh, Pedro O Pinheiro, David Vazquez, and Mark Schmidt. Where are the blobs: Counting by localization with point supervision. In *ECCV*, 2018.

[28] Issam H Laradji, Negar Rostamzadeh, Pedro O Pinheiro, David Vazquez, and Mark Schmidt. Proposal-based instance segmentation with point supervision. In *ICIP*, 2020.

[29] Zhuwen Li, Qifeng Chen, and Vladlen Koltun. Interactive image segmentation with latent diversity. In *CVPR*, 2018.

[30] JunHao Liew, Yunchao Wei, Wei Xiong, Sim-Heng Ong, and Jiashi Feng. Regional interactive image segmentation networks. In *ICCV*, 2017.

[31] Di Lin, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun. ScribbleSup: Scribble-supervised convolutional networks for semantic segmentation. In *CVPR*, 2016.

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

[33] 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.

[34] Yun Liu, Yu-Huan Wu, Pei-Song Wen, Yu-Jun Shi, Yu Qiu, and Ming-Ming Cheng. Leveraging instance-, image- and dataset-level information for weakly supervised instance segmentation. *PAMI*, 2020.

[35] Kevis-Kokitsi Maninis, Sergi Caelles, Jordi Pont-Tuset, and Luc Van Gool. Deep extreme cut: From extreme points to object segmentation. In *CVPR*, 2018.

[36] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3D reconstruction in function space. In *CVPR*, 2019.

[37] Pascal Mettes and Cees GM Snoek. Pointly-supervised action localization. *IJCV*, 2019.

[38] Pascal Mettes, Jan C Van Gemert, and Cees GM Snoek. Spot on: Action localization from pointly-supervised proposals. In *ECCV*, 2016.

[39] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-Net: Fully convolutional neural networks for volumetric medical image segmentation. In *International conference on 3D vision (3DV)*, 2016.

[40] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010.

[41] Dim P Papadopoulos, Jasper RR Uijlings, Frank Keller, and Vittorio Ferrari. Extreme clicking for efficient object annotation. In *ICCV*, 2017.

[42] Dim P Papadopoulos, Jasper RR Uijlings, Frank Keller, and Vittorio Ferrari. Training object class detectors with click supervision. In *CVPR*, 2017.

[43] Jordi Pont-Tuset, Pablo Arbelaez, Jonathan T Barron, Ferran Marques, and Jitendra Malik. Multiscale combinatorial grouping for image segmentation and object proposal generation. *PAMI*, 2016.

[44] Rui Qian, Yunchao Wei, Honghui Shi, Jiachen Li, Jiaying Liu, and Thomas Huang. Weakly supervised scene parsing with point-based distance metric learning. In *AAAI*, 2019.

[45] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.

[46] Zhongzheng Ren, Zhiding Yu, Xiaodong Yang, Ming-Yu Liu, Alexander G. Schwing, and Jan Kautz. UFO$^2$: A unified framework towards omni-supervised object detection. In *ECCV*, 2020.

[47] Ellen Riloff and Janyce Wiebe. Learning extraction patterns for subjective expressions. In *EMNLP*, 2003.

[48] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "GrabCut" interactive foreground extraction using iterated graph cuts. *ACM transactions on graphics (TOG)*, 2004.

[49] H Scudder. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 1965.

[50] Matthew Tancik, Pratul P Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *NIPS*, 2020.

[51] Zhi Tian, Chunhua Shen, and Hao Chen. Conditional convolutions for instance segmentation. In *ECCV*, 2020.

[52] Zhi Tian, Chunhua Shen, Xinlong Wang, and Hao Chen. BoxInst: High-performance instance segmentation with box annotations. In *CVPR*, 2021.

[53] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *IJCV*, 2013.

[54] Turner Whitted. An improved illumination model for shaded display. In *ACM Siggraph 2005 Courses*, 2005.

[55] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019.

[56] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *CVPR*, 2020.

[57] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017.

[58] Ning Xu, Brian Price, Scott Cohen, Jimei Yang, and Thomas S Huang. Deep interactive object selection. In *CVPR*, 2016.

[59] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *ACL*, 1995.

[60] Yanzhao Zhou, Yi Zhu, Qixiang Ye, Qiang Qiu, and Jianbin Jiao. Weakly supervised instance segmentation using class peak response. In *CVPR*, 2018.

[61] 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.

[62] Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin D Cubuk, and Quoc V Le. Rethinking pre-training and self-training. In *NIPS*, 2020.