ISSN 2464-4617 (print)
ISSN 2464-4625 (online)
Computer Science Research Notes - CSRN
http://www.wscg.eu
WSCG 2025 Proceedings

Balancing Bounding Box and Mask Annotations for Semi-Supervised Instance Segmentation

Daniil Tolstykh
ENGIE Lab CRIGEN
4, rue Josephine Baker
93240, Stains, France
daniil.tolstykh@external.engie.com

Dmitriy Slutskiy
ENGIE Lab CRIGEN
4, rue Josephine Baker
93240, Stains, France
dmitriy.slutskiy@engie.com

ABSTRACT

Instance segmentation models are crucial for precise object detection but often require expensive pixel-wise mask annotations. This paper studies the impact of combining bounding box and mask annotations in semi-supervised segmentation. We propose a method that leverages from both types of labeled data within a unified training framework. Through experiments on YOLO (convolution-based) and DETR (transformer-based) architectures, we demonstrate that balancing these annotation types significantly enhances performance while reducing labeling costs, particularly in terms of manual annotation time. Additionally, we evaluate few-shot and zero-shot scenarios, further highlighting the flexibility and efficiency of our method for budget-constrained segmentation tasks.

Keywords

Instance segmentation, Semi-supervised learning, Labeling cost optimization, YOLOv5, DETR.

1 INTRODUCTION

Neural networks have become a standard tool for image processing, with the main tasks typically categorized into image classification, object detection, and segmentation. Image classification involves tagging an image to indicate whether it contains an object of a particular type. Object detection extends this by identifying and localizing objects of interest and embedding them within bounding rectangles. Segmentation, on the other hand, produces pixel-wise masks, allowing models to draw the exact boundaries of objects in an image. Training these models typically requires annotations that match the type of predictions they are designed to produce. For example, semantic segmentation assigns a class label to each pixel in an image, whereas instance segmentation [15, 16] not only assigns semantic labels, but also distinguishes between individual objects of the same class, assigning unique instance IDs [4, 14].

Segmentation models demand significant annotation effort, with pixel-wise masks being far more time-consuming to create compared to bounding boxes or image-level labels. Weakly and semi-supervised learning methods aim to alleviate this burden by utilizing cheaper forms of annotation. However, an underexplored issue is how changes in the distribution of bounding box and mask annotations affect model performance. Given that data distributions can shift or be imbalanced in real-world scenarios, it is critical to understand the impact of this imbalance on model generalization and segmentation quality.

This paper studies how the ratio of weakly labeled data, namely bounding boxes, impacts the effectiveness of semi-supervised learning in segmentation tasks. To this end, we propose a method for training segmentation models by combining images with pixel-wise masks and bounding box annotations in a single procedure. We demonstrate that a proper balance between mask-labeled and bounding box-labeled training data improves model performance while reducing annotation time. Additionally, we show that the models can achieve similar performance on datasets with varying proportions of the two annotation types.

We tested our idea for the convolution-based YOLO network and for the transformer-based DETR model (refer to Sec. 3.3 for details). We conclude that the optimal mask-to-box labeled data ratio for training segmentation models is approximately 1:4 . . . 1:7.

Finally, we examine two extreme scenarios known as "few-shot" and "zero-shot" learning [31], when we train a YOLO segmentation model on a dataset that has only few mask annotations for each class or even contains bounding box annotations only (refer to Sec. 4.6).

2 RELATED WORK

2.1 Instance segmentation

Fully supervised instance segmentation methods [17, 24, 3, 9, 28] have demonstrated high performance; however, these approaches require substantial annotation efforts. The variety of instance segmentation methods [17, 24, 3, 5, 9, 28] can be categorized into single-stage, two-stage, and multi-stage based on the number of stages involved. Two-stage methods, like Mask R-CNN [17], use a detect-then-segment approach, starting with object detection (e.g, Faster R-CNN [33]) followed by segmentation within each detected box. Single-stage methods, such as YOLACT [3], inspired by YOLOv3 [32], and CenterMask [24], aim to balance speed and accuracy by performing instance segmentation directly. Multi-stage methods, often using the Transformers framework [35, 5, 10], support end-to-end segmentation, eliminating the need for non-maximum suppression and manual anchor box selection. We chose to work with YOLO and DETR since they are based on fundamentally different principles (convolutions and transformers, respectively) while being popular, powerful, and open-source.

2.2 Budget-Efficient Instance Segmentation

The labeling of pixel-wise masks is considerably more time-consuming compared to bounding boxes or image-level labels. For example, as reported in [2], the time required to annotate full masks in the Pascal VOC dataset [13] is approximately 6.3 times greater than that required for bounding box annotations. Weakly-supervised instance segmentation seeks to reduce the labeling burden by relying on weaker forms of supervision, such as image-level labels [22, 29] or bounding boxes [34, 23, 26, 18], instead of pixel-perfect masks. In contrast, semi-supervised methods leverage a combination of labeled and unlabeled data to enhance model performance [36, 30, 8]. By utilizing the segmentations learned from fully labeled data, these methods generate pseudo-instance masks for the unlabeled data. Recently, weakly semi-supervised methods [21] and hybrid supervised approaches [7] have been proposed for budget-efficient instance segmentation. Most papers in weakly and semi-supervised learning follow the existing data splits and settings in the benchmark. In contrast, our work goes beyond these conventional setups by exploring the balance between different types of training data.

3 METHOD

As mentioned in [2], with a fixed annotation budget, the number of annotated images depends on the level of supervision selected. Lower levels of supervision allow for annotating more images because they require less detailed annotation work, enabling annotators to process a larger volume of images within the same time frame. The key question is how to allocate the budget: should it be spent on fewer images with highly detailed supervision, such as pixel-wise masks, which provide precise object boundaries, or on weaker labels, like bounding boxes, which are less detailed but cover a larger number of images? To address this question, we considered different data splits, varying number of pixel-wise and bounding box labeled images in the training dataset. First, we analyze these splits based on two criteria: the cost of the labeling process and the performance of the model trained on each data split (refer to Sec. 4.3). Second, we conduct a visual analysis of the model trained with different data splits (refer to Sec. 4.4).

3.1 Data splits

Following [19], we refer to the complete set of images that includes instance segmentation annotations (i.e., pixel-wise masks and corresponding bounding boxes), as the *full set* $\mathbf{F}$, and the set of images having only bounding box annotations as the *weak set* $\mathbf{W}$.

To assess the impact of fixing different annotation budgets, according to [2, 19], we consider different budget scenarios by varying the number of full labels $|\mathbf{F}| \in \{200, 400, 800, 1464\}$ for Pascal VOC 2012 and $\{200, 400, 800, 1475\}$ for Cityscapes [11]. In prior works [19, 7], the authors studied the only case where the weak set $\mathbf{W}$ consists of all the images that were not selected for the full set $\mathbf{F}$, i.e. the number of weak labels is equal to the *total size of the training dataset* minus $|\mathbf{F}|$. In contrast, to examine the impact of the different data splits, we vary the number of weak labels as follows: $\{0, 1000, 3000, 5000, 10731 - |\mathbf{F}|\}$ for Pascal VOC 2012 and $\{0, 1000, 2000, 2975 - |\mathbf{F}|\}$ for Cityscapes.

3.2 Annotation cost

To estimate the labeling cost, we employed the method described in [2]. Tab. 1 presents the average labeling time for a single image for the Pascal VOC 2012 and Cityscapes datasets, respectively.

Label type	Pascal VOC 2012	Cityscapes
Bounding box	38.1s/image	127.5s/image
Pixel-wise	239.7s/image	1,387.5s/image

Table 1: The annotation cost (in seconds) for estimation of heterogeneous labels on the PASCAL VOC 2012 and Cityscapes datasets from [2, 7]

3.3 Models

YOLOv5 segmentation model. YOLOv5 [20] is a widely used computer vision model originally designed

for object detection, known for its state-of-the-art performance on the COCO dataset [27] and its training and deployment efficiency. It was later extended to include instance segmentation, resulting in the YOLACT model [3]. The YOLACT segmentation head is similar to YOLOv5's, featuring a modified detection head with a mask branch and Protonet [3], a small convolution network for generating prototype masks. The prediction head of the YOLOv5 segmentation model extends the detection model's head by including a mask branch that predicts "mask coefficients", which are combined with prototype masks to produce the final output mask for each instance. For more details, see [3].

Based on the implementation of the YOLOv5 architecture, for instance segmentation [20], we modified a one-stage training procedure to be able to mix bounding box and mask labeled data in a single batch. See Sec. 3.4 for more details.

DETR segmentation model: DETR (DEtection TRansformer) is a model developed by Facebook [5, 6] for object detection, using a transformer-based encoder-decoder architecture [35] that bypasses traditional methods like non-maximum suppression and anchor boxes. In [5], DETR was extended for instance segmentation tasks by adding a mask head to the decoder outputs. Training follows a two-step process: first for object detection, then fine-tuning for segmentation, which proved to be faster than a one-step segmentation training.

3.4 Overview of the semi-supervised training strategy

The standard semi-supervised training procedure consists of two stages: first, training the model on object detection using the weak set, followed by fine-tuning on instance segmentation with the full set. This approach is effective for the DETR model due to its architecture, but not for the YOLOv5 model. The primary issue is that the detection head must be partially replaced with the segmentation head, resulting in the removal of some previously trained model weights and a decrease in model quality. Thus, for YOLOv5, we modified the training procedure to allow images with weak and full annotations to be used in a single batch during segmentation model training.

For the YOLOv5 model, we base our approach on the PyTorch implementation from the Ultralytics repository [20]. Next, we describe the original and modified training procedures, as well as the training losses for YOLOv5.

YOLO Loss functions. The overall loss function in YOLOv5 segmentation model (see [20]) is computed as a combination of four individual loss components:

$$Loss = \lambda_1 L_{cls} + \lambda_2 L_{obj} + \lambda_3 L_{loc} + \lambda_3 L_{mask}, \quad (1)$$

This structure allows individual loss components to be deactivated while keeping the others. Moreover, it is possible to disable a single loss component for particular batch elements during training. This feature is utilized below.

One-stage semi-supervised setup. The original YOLOv5 training procedure for instance segmentation consists of the following steps:

- create a dataset that includes both images and labels, and a dataloader to iterate over it.
- take a batch of images from the dataloader and propagate it through the model
- calculate the loss for the batch and backpropagate the gradients to adjust the model's weights.

Since we intend to train a model with mixed full/weak batch, the initial training procedure was modified:

- as there is no segmentation annotation for the weak set, we create rectangular masks from the bounding boxes to preserve dataloader conventions,
- we create an additional dataloader to iterate over the weak dataset,
- during training, the algorithm iterates over two dataloaders simultaneously and fuses batches from different dataloaders into a single one
- the loss function does not take weak set masks into account when calculating the Binary Cross-Entropy loss L_{mask}.

4 EXPERIMENTS

In this section, we conduct quantitative evaluation of the instance segmentation models trained with different number of pixel-wise labels and bounding box labels. We compare our results to the state-of-the-art approaches. We also perform a visual analysis of the model quality for different data splits used for training.

4.1 Datasets

We used Pascal VOC 2012 [13] as the primary dataset for training our YOLOv5 and DETR segmentation models. Also, we used Cityscapes [11] only for the YOLOv5 model since DETR, being transformer-based, requires a lot of data.

Pascal VOC 2012. Pascal VOC 2012 (hereinafter referred to as VOC2012) [13] consists of 1464 training, 1449 validation, and 1456 test images including 20 object categories for instance segmentation. Additionally, it includes 9,267 images with only bounding box annotations. We report instance segmentation results with $AP50$, $AP70$, $AP75$ on the validation set.

Cityscapes. Cityscapes [11] includes 2975 training, 500 validation, and 1525 test images of urban scenes with pixel-level annotations for 30 object classes (e.g, cars, pedestrians). It provides fine and coarse mask annotations. We use only the *fine* data and 8 classes (person, rider, car, truck, bus, train, motorcycle, and bicycle), since instance segmentation isn't available for the other classes. We measure the performance by using *AP*50 and *AP*50:95 on the validation set.

Method	|**F**|	|**W**|	Labeling cost (h)	*AP*50	*AP*70	*AP*75
Semi and Weakly supervision						
LACI[1]	-	10582	112	57.7	33.5	31.2
BoxInst[34]	-	10582	112	61.4	-	37.0
BBTP w/CRF[18]	-	10582	112	59.1	-	21.9
BBAM[23]	-	10582	112	63.7	39.5	31.8
Box2Mask-T[26]	-	10582	112	70.8	50.8	44.4
Budget-aware[2]	100	-	6.7	14.9	-	-
	200	-	13.3	23.7	-	-
	400	-	26.6	35.5	-	-
	800	-	53.3	42.9	-	-
	1464	-	97.5	46.8	-	-
Full supervision						
Budget-aware[2]	10582	-	704.6	56.4	-	-
Mask R-CNN[17]	10582	-	704.6	67.9	50.7	43.5
Center-mask[24]	10582	-	704.6	68.8	53.1	45.9
YOLACT[3]	10582	-	704.6	72.3	56.2	-
Hybrid supervision [7]						
Mask-R-CNN	105	10477	117.9	61.8	42.7	35.2
	317	10265	129.7	63.5	45.0	38.5
	529	10053	141.6	63.9	46.3	39.7
	1058	9524	171.2	65.4	48.0	41.6
	2116	8446	230.3	66.2	49.4	42.2
	5291	5291	408.3	66.9	50.0	43.5
CenterMask	105	10477	117.9	62.2	45.1	38.2
	317	10265	129.7	64.2	47.9	41.6
	529	10053	141.6	65.3	49.2	42.5
	1058	9524	171.2	66.1	50.8	44.6
	2116	8446	230.3	66.5	51.6	45.4
	5291	5291	408.3	67.2	52.1	45.9
Ours						
YOLOv5	200	-	13.3	49.3	34.5	29.6
	200	1000	23.9	56.5	40.6	35.0
	200	3000	45.1	62.9	44.9	38.8
	200	5000	66.2	65.3	46.8	40.2
	200	10531	122.7	67.4	50.3	44.4
	400	-	26.6	52.9	38.6	34.4
	400	1000	37.2	58.1	43.0	37.8
	400	3000	58.4	63.8	47.6	41.7
	400	5000	79.5	65.9	49.9	44.1
	400	10331	136	70.2	52.8	46.5
	800	-	53.3	58.1	44.3	39.8
	800	1000	63.9	61.7	46.6	41.6
	800	3000	85.0	66.5	50.2	44.4
	800	5000	106.2	69.0	53.5	47.4
	800	9931	158.4	71.6	54.9	48.9
	1464	-	97.5	63.6	49.9	44.0
	1464	1000	108.1	65.7	51.0	46.1
	1464	3000	129.2	68.7	54.0	48.3
	1464	5000	150.4	70.0	55.5	50.8
	1464	9267	195.6	74.7	58.8	53.3
DETR	400	1000	37.2	37.4	21.0	15.5
	400	3000	58.4	46.8	25.8	20.3
	400	5000	79.5	50.4	29.5	22.8
	400	10331	136	55.9	33.7	36.8
	800	1000	63.9	43.6	26.7	21.6
	800	3000	85.0	52.8	34.4	27.0
	800	5000	106.2	55.5	36.3	30.0
	800	9931	158.4	60.9	40.6	34.1
	1464	1000	108.1	49.9	33.4	27.8
	1464	3000	129.2	56.9	39.5	33.0
	1464	5000	150.4	59.4	40.9	33.9
	1464	9267	195.6	62.8	44.9	37.8

Table 2: The instance segmentation results for the VOC2012 validation set. The underlined models illustrate the trade-offs between using the full and weak datasets. They are visualized on Fig. 2 and Fig. 4.

Method	|**F**|	|**W**|	Labeling cost (h)	*AP*50	*AP*50:95
Semi and Weakly supervision					
WSSPS[25]	-	2975	105.4	-	17.0
Ubteacher[30]	148	-	57.0	-	16.0
	297	-	114.5	-	20.0
	595	-	229.3	-	27.1
	893	-	344.2	-	28.0
Noisy Boundaries[36]	148	-	57.0	-	17.1
	297	-	114.5	-	22.1
	595	-	229.3	-	29.0
	893	-	344.2	-	32.4
	1190	-	458.6	-	33.0
Full supervision					
Mask-R-CNN[17]	2975	-	1146.6	59.5	31.5
PANet[28]	2975	-	1146.6	57.1	31.8
Center-mask[24]	2975	-	1146.6	61.7	34.7
Mask2Former[9]	2975	-	1146.6	63.9	38.5
Hybrid supervision[7]					
Mask-R-CNN	29	2946	115.5	40.2	19.4
	89	2886	136.5	49.6	24.2
	148	2827	157.2	51.8	25.8
	297	2678	209.3	55.0	28.5
	595	2380	313.6	56.8	29.5
	1486	1488	625.4	59.2	31.1
CenterMask	29	2946	115.5	39.9	18.7
	89	2886	136.5	51.8	26.7
	148	2827	157.2	54.0	27.4
	297	2678	209.3	56.1	30.5
	595	2380	313.6	57.9	32.1
	1486	1488	625.4	61.4	34.0
Ours					
YOLOv5	200	-	77.1	35.3	19.4
	200	1000	112.5	49.8	27.3
	200	2000	147.9	53.2	29.6
	200	2775	175.4	54.1	30.0
	400	-	154.2	43.6	24.0
	400	1000	189.6	52.0	29.6
	400	2000	225	55.2	31.8
	400	2575	245.4	55.3	31.6
	800	-	308.3	47.6	27.5
	800	1000	343.8	54.3	31.7
	800	2000	379.2	55.7	32.7
	1475	-	568.5	53.5	31.4
	1475	1000	603.9	55.0	33.0
	1475	1500	621.6	56.0	33.3
	2975	-	1146.6	56.5	33.8

Table 3: The instance segmentation results for the Cityscapes validation set. The underlined models illustrate the trade-offs between using the full and weak datasets. They are visualized on Fig. 3.

4.2 Training procedure

We use both YOLOv5 and DETR segmentation models with backbones pretrained on the ImageNet dataset [12] as is common practice. Model performance is measured using the mean average precision (AP) across the available classes. Our source code is available in `https://github.com/OneMagicKey/optimal-annotation-mix`.

YOLO. The YOLOv5 segmentation model was trained using a batch size of 16 and the SGD optimizer. We used an image size of 512 for VOC2012 and 1024 for Cityscapes, default augmentations, and a scheduler provided by [20]. The underlining is explained in Sec. 4.4.

DETR. We use DETR with ResNet-50 backbone for our experiments. The default image size was set to 800 pixels, with a maximum size of 1333 pixels. We used the AdamW optimizer and augmentations from the original repository [6]. We train DETR according to the original procedure described in [5]. First,

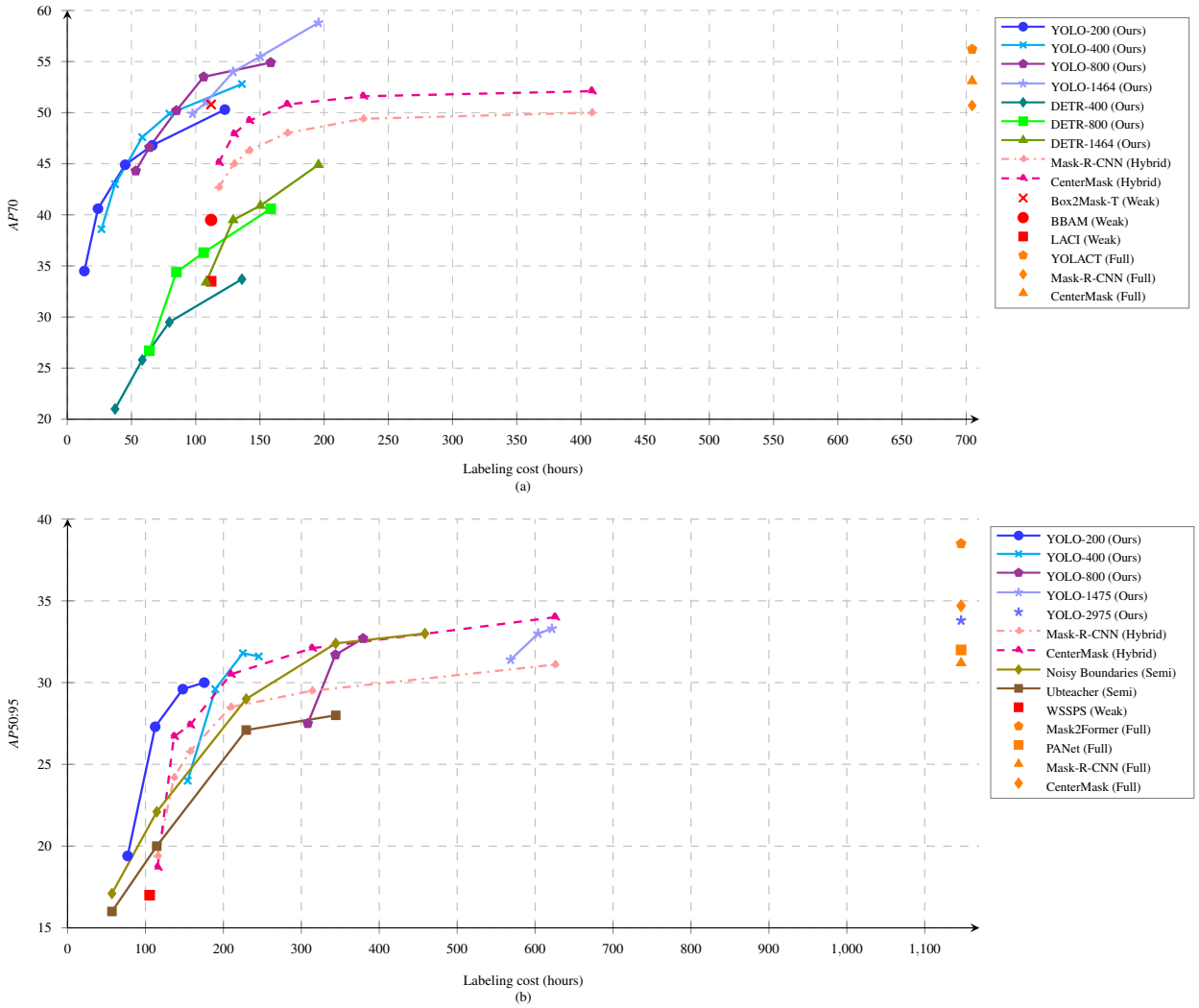


Figure 1: Evaluation of instance segmentation results from different methods. (a) is results on the VOC2012 dataset, (b) is results on the Cityscapes dataset. The number in the model name indicates the size of **F** dataset used for training

we train the model using only boxes. Then, we freeze all weights and train only the mask head. During the bounding box phase, the model was trained with a batch size of 10, while the mask phase uses a batch size of 3. We use the publicly available PyTorch implementation from [6].

4.3 Experimental results

YOLOv5 and DETR models are evaluated according to two criteria: their performance (the higher, the better), and the cost of the labeling process (the lower, the better). The labeling cost is calculated based on the statistics presented in Tab. 1. We report our results against weakly-supervised [25, 1, 34, 18, 23, 26], semi-supervised [30, 36], hybrid-supervised [7], and fully supervised [17, 28, 24, 9, 3] methods. The results are listed in Tabs. 2 and 3. The columns represent the method name, the number of fully labeled images ($|\mathbf{F}|$), the number of weakly labeled images ($|\mathbf{W}|$), the labeling cost (in hours), and the instance segmentation performance metrics. We show that our semi-supervised setup, which balances fully labeled and weakly labeled images, outperforms state-of-the-art approaches at a comparable or reduced labeling cost.

The overall results are shown in Fig. 1 (a) and Fig. 1 (b). For clarity, we grouped our models trained on the same size of the **F** dataset in Fig. 1. To differentiate models trained on different data splits, we include the number of full (**F**) and weak (**W**) images in the model name. If the model name ends with the number of full images only, it represents a group of models with the same **F** set but varying **W** sets. For example, YOLO-400/5000 refers to a model trained with $|\mathbf{F}| = 400$ and $|\mathbf{W}| = 5000$, while YOLO-400 represents a group of models such as YOLO-400/0, YOLO-400/1000, ... , YOLO-400/10331.

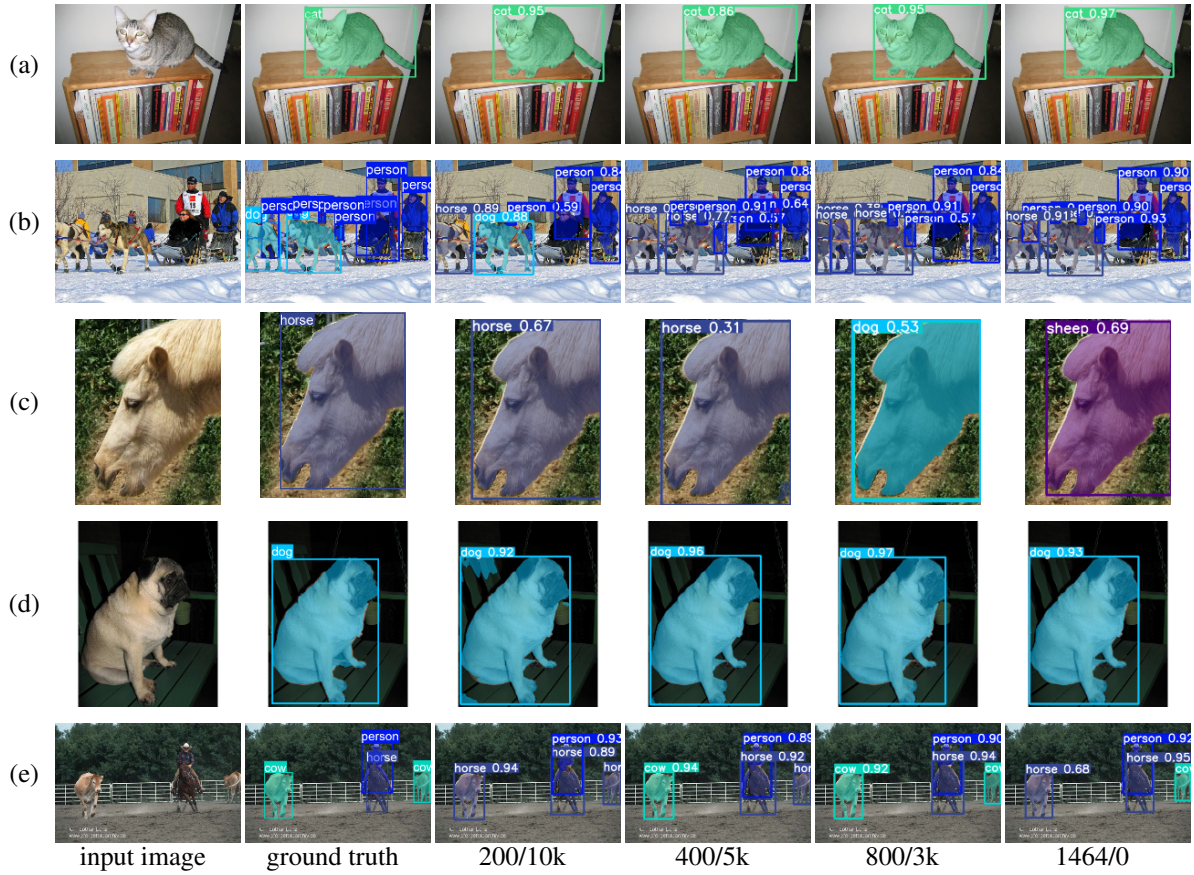


Figure 2: Visualizations of instance segmentation results on the VOC2012 validation set for different $|\mathbf{F}|/|\mathbf{W}|$ data splits using the YOLO model. The values in the column headers represent the sizes of the $\mathbf{F}$ and $\mathbf{W}$ datasets used for model training. The numbers within the images represent the confidence scores for each segmented instance.

Comparison with state-of-the-art. For VOC2012, our semi-supervised model YOLO-1464/9267 surpasses the fully supervised methods Mask R-CNN [17], CenterMask [24], and YOLACT [3] by 8.1, 5.7, and 2.6 in AP70, respectively, while reducing labeling costs by 3.5 times. Similarly, under the same labeling cost, YOLO-800/5000 outperforms the weakly supervised methods LACI [1], BBAM [23], and Box2Mask_T [26] by 20, 14, and 2.7 in AP70, respectively. Furthermore, it also outperforms Hybrid Mask R-CNN [7] and CenterMask [7] across all data splits, while being cheaper to label.

In the Cityscapes dataset, our YOLO-200/1000 model outperforms WSSPS [25], Ubteacher-297/0 [30], and Noisy Boundaries-297/0 [36] by 10.3, 7.3, and 5.2 in AP50:95, while maintaining the same labeling cost. Additionally, our YOLO-1475/1500 model achieves 33.3 AP50:95, nearly matching the performance of the fully supervised CenterMask [24] (34.7) and outperforming both PANet [28] (31.8) and Mask R-CNN [17] (31.5), while reducing labeling costs by half.

Notably, DETR performs significantly worse than YOLO. We attribute this to the nature of DETR as a transformer-based model, which generally requires more training data compared to convolution-based models like YOLO. In addition, DETR struggles to detect small objects and requires higher training resolutions (800 pixels versus YOLO's 512 pixels).

Search for optimal training data split. We begin by analyzing the performance of various YOLO models on the VOC2012 dataset. As shown in Fig. 1 (a), splits with $|\mathbf{W}| = 0$ (the leftmost points on all YOLO graphs) are suboptimal as there are YOLO models with the same annotation cost but better *AP*70 scores. A similar observation applies to models where $|\mathbf{W}| = 10731 - |\mathbf{F}|$ (the rightmost points on all YOLO graphs). Edge cases like YOLO-200/0 and YOLO-1464/9267 (the leftmost and rightmost points on their respective graphs) are excluded, as they lack meaningful comparability. The best *AP*70 results, for fixed annotation costs, are achieved with the models YOLO-400/5000 (the 4th point from the left on the YOLO-400 graph) and YOLO-800/5000 (the 4th point from the left on the YOLO-800 graph). From this analysis, we conclude that

Observation 1. *Extreme training data splits, where the absolute majority is either* $\mathbf{F}$ *or* $\mathbf{W}$*, are not optimal in terms of labeling cost and performance.*

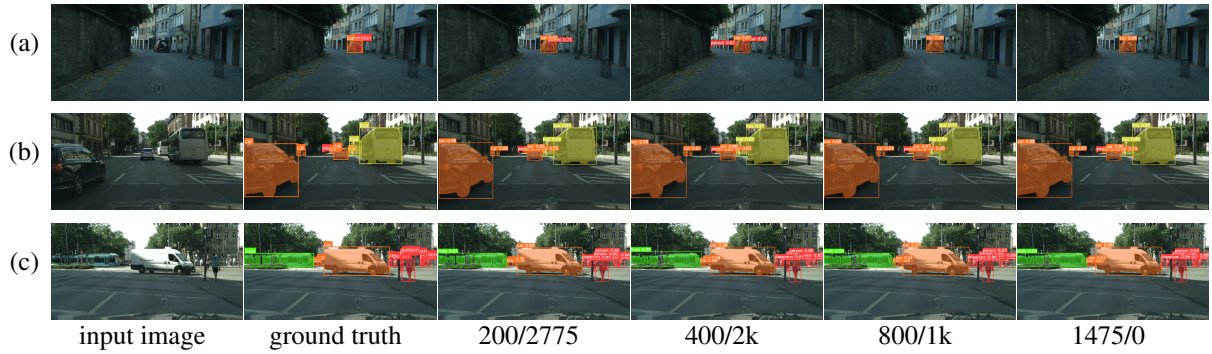


Figure 3: Visualizations of instance segmentation results on the Cityscapes validation set for different $|\mathbf{F}|/|\mathbf{W}|$ data splits using the YOLO model. The numbers in the column headers represent the sizes of the **F** and **W** datasets used for model training. The numbers within the images represent the confidence scores for each segmented instance.

For the DETR model, the same principle holds true. The highest *AP*70 scores for fixed annotation costs are achieved by DETR-800/3000 and DETR-800/5000 models, corresponding to the second and third points from the left on the DETR-800 graph, respectively. This indicates that the most effective $|\mathbf{F}|/|\mathbf{W}|$ splits in terms of the annotation cost are the "middle" ones, thereby confirming Observation 1.

Similar observations apply to the YOLO model trained on Cityscapes dataset. In Fig. 1 (b), for every model with $|\mathbf{W}| = 0$ (except YOLO-200/0), alternative models achieve the same *AP*50:95 scores at lower annotation costs. Additionally, the YOLO-2975/0 model has twice the annotation cost of YOLO-1475/1500, despite achieving an equal *AP*50:95 score. Notably, the YOLO-800/2000 model achieves almost the same *AP*50:95 score as YOLO-1475/1500 at about half the annotation cost.

We complete this subsection with two natural remarks, that follow from Tabs. 2 and 3:

Observation 2. *Even with a small* $|\mathbf{F}|$, *the model achieves competitive results, provided that* $|\mathbf{W}|$ *is large enough.* Refer to Sec. 4.6 for a discussion of extreme cases.

Observation 3. *Larger sizes of the set* **W** *result in better performance for any given size of* **F**; *in other words, increasing the weakly labeled set improves box quality and enhances AP for the masks. Similarly, larger sizes of the set* **F** *lead to better performance regardless of the size of* **W**.

4.4 Visual data split analysis

This section focuses on the visual analysis of the model output. By examining how models trained on different splits of the training data perform on specific images, we can identify trade-offs in mask quality, detection accuracy, and class assignment.

To facilitate this analysis, we selected models trained on different data splits that achieved identical *AP* scores. These models, underlined in Tabs. 2 and 3, illustrate the trade-offs between using the full and weak datasets.

Fig. 2 showcases instance segmentation results for five representative images from the VOC2012 dataset, produced by the YOLO model. The results are displayed for four selected models with similar *AP*70 scores. We highlight the following observations:

Observation 4. *Models with identical AP70 scores produce visually similar segmentations, especially for large, easily detected objects* (see Fig. 2 (a)).

Observation 5. *Training with more weak examples improves a model's ability to locate and classify objects* (see Fig. 2 (b, c)) For instance, if there are classes that are visually or semantically similar, such as horses and sled dogs, or sheep and horse heads, a model trained on a dataset with a larger **W** will make fewer errors in determining the class of an object.

Observation 6. *A larger size of the* **F** *set is expected to result in better mask quality* (see Fig. 2 (d)).

Observation 7. *The "middle" 800/3K* $|\mathbf{F}|/|\mathbf{W}|$ *split appears to be the optimal choice visually, as it strikes a balance between mask quality* (Observation 6) *and class identification* (Observation 5). See Fig. 2 (e).

Fig. 3 presents the instance segmentation masks produced by the YOLO model trained on the Cityscapes dataset. Compared to VOC2012, Cityscapes is a more homogeneous dataset than VOC2012, with high-quality images and only 7 instance categories, which implies that the produced masks are visually similar for the $|\mathbf{F}|/|\mathbf{W}|$ splits with close *AP* scores (for instance, see Fig. 3 (b, c)). This confirms Observation 4. For the same reason, examples in the Cityscapes validation dataset that could illustrate Observations 5-7 are rare. In particular, see Fig. 3 (a) which illustrates Observation 5.

Finally, Fig. 4 shows the instance segmentation masks obtained by the DETR model trained on the VOC2012 dataset. The results of our experiments verified most of the observations. While limited training data prevents

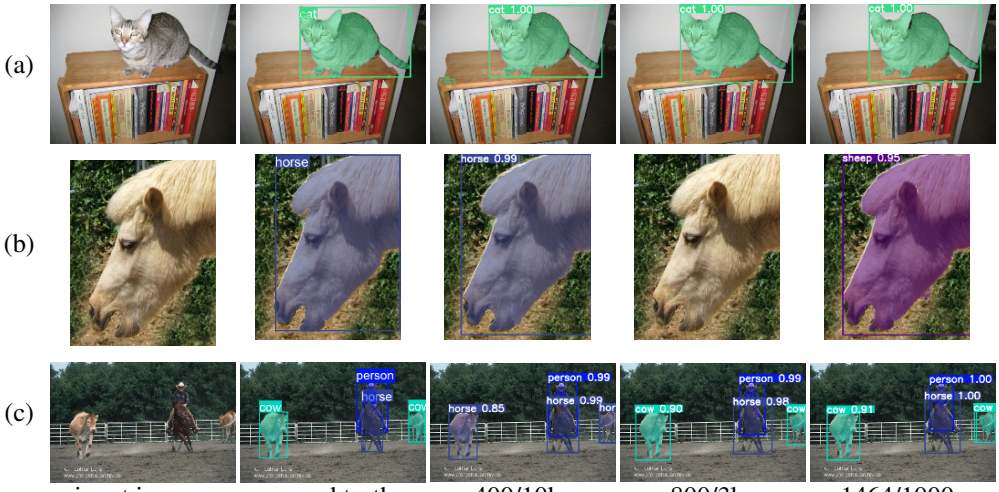


Figure 4: Visualizations of instance segmentation results on the VOC2012 validation set for different $|\mathbf{F}|/|\mathbf{W}|$ data splits using the DETR model. The numbers in the column headers represent the sizes of the $\mathbf{F}$ and $\mathbf{W}$ datasets used for model training. The numbers within the images represent the confidence scores for each segmented instance.

confirming all observations, no contradictions were noted. In particular, Fig. 4 (a) illustrates that the produced instance segmentation masks are similar, aligning with Observation 4, while Fig. 4 (b) supports Observation 5.

4.5 Heuristic formula for an |F|/|W| ratio to maintain the AP score

In this section, we present a heuristic formula to determine the $|\mathbf{F}|/|\mathbf{W}|$ training data ratio curve for the models with the same *AP* score. This formula is derived from the underlined experiments in Tabs. 2 and 3.

Remark 1. *As $|\mathbf{F}|$ decreases and $|\mathbf{W}|$ increases, the values $|\mathbf{W}|$ and $|\mathbf{F}+\mathbf{W}|$ differ slightly.*

Using this observation, we propose the following:

Assertion 1. *Let ϕ_0 represent the maximum number of $\mathbf{F}$ images used in the experiments, and let $\tau \in (0,1)$. Models trained with the following quantities of $|\mathbf{F}|/|\mathbf{W}|/|\mathbf{F}+\mathbf{W}|$ annotations achieve approximately equal AP scores*:

$$|\mathbf{F}|(\tau) = \tau\phi_0,\ |\mathbf{W}|(\tau) = \frac{c(\tau)}{\tau}\phi_0, \qquad (|\mathbf{F}+\mathbf{W}|)(\tau) = (\tau + \frac{c(\tau)}{\tau})\phi_0, \tag{2}$$

$$\textit{where}\ \ \frac{c(\tau)}{\tau} \xrightarrow[\tau\to 1]{} 0 \tag{3}$$

$$\textit{and}\ \ c(\tau) \approx k \tag{4}$$

otherwise. Here k is a positive constant.

Remark 2. *The validity of this formula near the extreme values of 0 and 1 of the parameter τ is not relevant for practical tasks.*

For instance, using Tab. 2, the constants from Assertion 1 are $\phi_0 = 1464, k = 1$, the $|\mathbf{F}|/|\mathbf{W}|$ equiscore law can be approximated by the hyperbola

$$|\mathbf{W}|(|\mathbf{F}|) = \frac{\phi_0^2}{|\mathbf{F}|} \tag{5}$$

This equation holds as long as $|\mathbf{F}|$ is not close to 0 or ϕ_0, according to Remark 2 (see Fig. 5a). From Eq. (3), hyperbola (5) fails to converge to the experimental point $(1464, 0)$ at $\tau = 1$, as $c(\tau)$ is not constant near $\tau = 1$.

Similarly, for Tab. 3, the constants are $\phi_0 = 1475, k = 1/3$. In this case, the $|\mathbf{F}|/|\mathbf{W}|$ equiscore law approximates the hyperbola

$$|\mathbf{W}|(|\mathbf{F}|) = \frac{\phi_0^2}{3|\mathbf{F}|} \tag{6}$$

This remains valid as long as $|\mathbf{F}|$ is not near 0 and ϕ_0, per Remark 2 (see Fig. 5b). The experimental point $(200, 2775)$ aligns with an *AP* score of 30.0, which is lower than the equiscore parameter of 31.5. Therefore, the experimental point with $|\mathbf{F}| = 200$ and score 31.5 should be closer to the hyperbola (6).

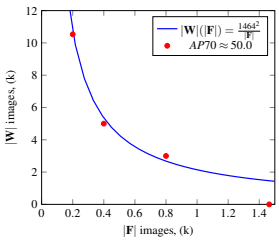
Assertion 1 also applies to the DETR model, with the constants being $\phi_0 = 1464$ and $k = 3/2$, which aligns with the data in Tab. 2. This means that the $|\mathbf{F}|/|\mathbf{W}|$ equiscore law is approximately described by the hyperbola

$$|\mathbf{W}|(|\mathbf{F}|) = \frac{3}{2}\frac{\phi_0^2}{|\mathbf{F}|} \tag{7}$$

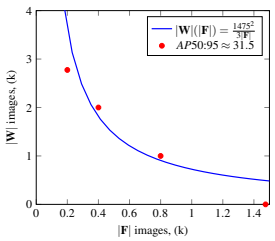
as long as $|\mathbf{F}|$ is not close to 0 and ϕ_0, consistent with Remark 2 (see Fig. 5c).

4.6 Few-shot and zero-shot

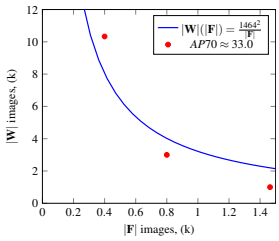
In this section, we investigated few-shot and zero-shot setups for the YOLO model. We did not test these



(a) YOLO on VOC2012 dataset.



(b) YOLO on Cityscapes dataset.



(c) DETR on VOC2012 dataset

Figure 5: The equiscore hyperbolas for different $|\mathbf{F}|/|\mathbf{W}|$ splits which illustrates Assertion 1.

approaches on DETR because of its poor performance with small **F** datasets.

Few-shot instance segmentation aims to accurately segment instances of a given object category using only a few annotated examples for model training, which is useful when collecting a large annotated dataset is impractical. In contrast, *zero-shot* instance segmentation targets segmenting object categories the model hasn't seen during training, relying on auxiliary information about unseen classes for accurate segmentation.

Pretrain	aeroplane	cow	bird	boat	sheep	dog	horse	cat	all
ImageNet	49.1	55.4	47.0	29.1	42.8	67.5	16.8	62.8	46.3
Cityscapes-seg	56.1	58.4	62.2	29.7	48.1	72.4	37.0	76.4	55.0
YOLO-1464/9267	64.6	71.5	75.4	51.7	61.4	84.1	69.2	87.6	70.6

Table 4: The *AP*70 results across 8 classes of the VOC2012 validation set for different few-shot training setups. YOLO models achieve decent results, demonstrating the potential of few-shot learning when extensive training data is unavailable

Few-shot learning. To evaluate the few-shot capabilities of the YOLO model, we implemented the following pipeline. First, we take the best model trained on the Cityscapes dataset. Second, we select images from the VOC2012 dataset that include only specific classes (aeroplane, cow, bird, boat, sheep, dog, horse, and cat), which are not present in the Cityscapes. Finally, we used 17 images containing 24 object masks (3 per class) as the **F** set, with the remaining images assigned to the **W**. We fine-tune the model for 20k iterations using the semi-supervised training procedure discussed in Sec. 3.4 (see Sec. 4.2 for training details). For comparison, the pipeline was also tested on YOLO with an ImageNet pretrained backbone, omitting the Cityscapes training step. We take the best YOLO model we trained in Sec. 4.2 as the baseline. We present our results in Tab. 4 and some visualizations in Fig. 6. The first two lines in Tab. 4 correspond to our few-shot experiments. Although the baseline YOLO model significantly outperforms both few-shot models in overall *AP*70 (46.3 for ImageNet-based and 55 for Cityscapes-seg-based versus 70.6 for the baseline), they still achieve decent results, demonstrating the potential of few-shot learning when extensive training data is unavailable.

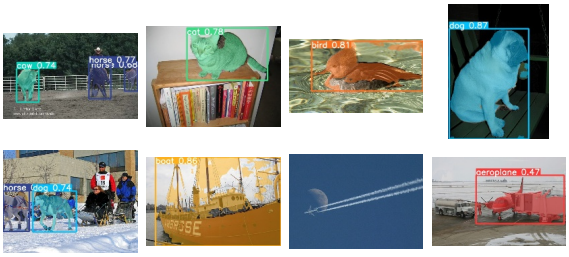


Figure 6: Visualization of few-shot results on the VOC2012 validation set, Cityscapes-seg pretrain.

Failure cases: zero-shot learning. The model showed poor performance in zero-shot learning. Although Protonet (see Sec. 3.3 and [3]) is able to learn some image features, fails to combine them effectively without training.

5 CONCLUSION

In this study, we explored the effectiveness of combining mask and bounding box annotations in training instance segmentation models using YOLOv5 and DETR architectures. Our experiments confirmed that a balanced mix of these annotation types can yield high-quality segmentation results while reducing the overall annotation cost. By leveraging semi-supervised learning strategies, we can achieve high-quality results with fewer annotated samples, making this approach highly efficient for practical applications. Additionally, our investigation into few-shot and zero-shot learning revealed that a minimal number of segmentation annotations can train a model with reasonable quality, but relying only on bounding boxes is insufficient.

6 ACKNOWLEDGMENTS

The authors express their gratitude to Irene De Teresa Trueba and the anonymous referees for their valuable comments, which significantly enhanced the readability of this text. Additionally, the authors extend their thanks to Fabrice Boudaud and the CSAI Lab of ENGIE CRIGEN for their support in this work.

7 REFERENCES

[1] Aditya Arun, C. V. Jawahar, and M. Pawan Kumar. Weakly supervised instance segmentation by learning annotation consistent instances. In *European Conference on Computer Vision*, 2020.

[2] Miriam Bellver, Amaia Salvador, Jordi Torres, and Xavier Giro i Nieto. Budget-aware semi-supervised semantic and instance segmentation. In *CVPRW*, 2019.

[3] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. Yolact: Real-time instance segmentation. In *ICCV*, 2019.

[4] Bert De Brabandere, Davy Neven, and Luc Van Gool. Semantic instance segmentation with a discriminative loss function. *arXiv preprint arXiv:1708.02551*, 2017.

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *arXiv preprint arXiv:2005.12872*, 2020.

[6] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. `https://github.com/facebookresearch/detr`, June 2020.

[7] Linwei Chen, Ying Fu, Shaodi You, and Hongzhe Liu. Hybrid supervised instance segmentation by learning label noise suppression. *Neurocomputing*, 496:131–146, 2022.

[8] Xin Chen, Jie Hu, Xiawu Zheng, Jianghang Lin, Liujuan Cao, and Rongrong Ji. Depth-guided semi-supervised instance segmentation. *ArXiv*, abs/2406.17413, 2024.

[9] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1280–1289, 2021.

[10] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2022.

[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 *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[13] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

[14] Alireza Fathi, Zbigniew Wojna, Vivek Rathod, Peng Wang, Hyun Oh Song, Sergio Guadarrama, and Kevin P. Murphy. Semantic instance segmentation via deep metric learning. *arXiv preprint arXiv:1703.10277*, 2017.

[15] Wenchao Gu, Shuang Bai, and Lingxing Kong. A review on 2d instance segmentation based on deep neural networks. *Image and Vision Computing*, 120:104401, 2022.

[16] Bharath Hariharan, Pablo Arbelaez, Ross Girshick, and Jitendra Malik. *Simultaneous Detection and Segmentation*, pages 297–312. Springer International Publishing, 2014.

[17] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017.

[18] 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 H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[19] Moustafa S. Ibrahim, Arash Vahdat, Mani Ranjbar, and William G. Macready. Semi-supervised semantic image segmentation with self-correcting networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[20] Glenn Jocher, Ayush Chaurasia, Alex Stoken, Jirka Borovec, NanoCode012, Yonghye Kwon, Kalen Michael, TaoXie, Jiacong Fang, imyhxy, Lorna, Zeng Yifu, Colin Wong, Abhiram V, Diego Montes, Zhiqiang Wang, Cristi Fati, Jebastin Nadar, Laughing, UnglvKitDe, Victor Sonck, tkianai, YxNONG, Piotr Skalski, Adam Hogan, Dhruv Nair, Max Strobel, and Mrinal Jain. ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation, Nov. 2022.

[21] Beomyoung Kim, Joonhyun Jeong, Dongyoon Han, and Sung Ju Hwang. The devil is in the points: Weakly semi-supervised instance segmentation via point-guided mask representation. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11360–11370, 2023.

[22] Beomyoung Kim, Young Joon Yoo, Chae-Eun Rhee, and Junmo Kim. Beyond semantic to instance segmentation: Weakly-supervised instance segmentation via semantic knowledge transfer and self-refinement. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4268–4277, 2021.

[23] Jungbeom Lee, Jihun Yi, Chaehun Shin, and Sungroh Yoon. Bbam: Bounding box attribution map for weakly supervised semantic and instance segmentation. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2643–2651, 2021.

[24] Youngwan Lee and Jongyoul Park. Centermask: Real-time anchor-free instance segmentation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13903–13912, 2020.

[25] Qizhu Li, Anurag Arnab, and Philip H. S. Torr. Weakly- and semi-supervised panoptic segmentation. In *Computer Vision - ECCV 2018: 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XV*, pages 106–124, Berlin, Heidelberg, 2018. Springer-Verlag.

[26] Wentong Li, Wenyu Liu, Jianke Zhu, Miaomiao Cui, Risheng Yu, Xiansheng Hua, and Lei Zhang. Box2Mask: Box-Supervised Instance Segmentation via Level-Set Evolution . *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 46(07):5157–5173, July 2024.

[27] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollar. Microsoft coco: Common objects in context, 2015.

[28] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8759–8768, 2018.

[29] Yun Liu, Yu-Huan Wu, Peisong Wen, Yujun Shi, Yu Qiu, and Ming-Ming Cheng. Leveraging instance-, image-and dataset-level information for weakly supervised instance segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3):1415–1428, 2020.

[30] Yen-Cheng Liu, Chih-Yao Ma, Zijian He, Chia-Wen Kuo, Kan Chen, Peizhao Zhang, Bichen Wu, Zsolt Kira, and Peter Vajda. Unbiased teacher for semi-supervised object detection. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[31] Mark Palatucci, Dean Pomerleau, Geoffrey E Hinton, and Tom M Mitchell. Zero-shot learning with semantic output codes. In *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc., 2009.

[32] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[33] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[34] Zhi Tian, Chunhua Shen, Xinlong Wang, and Hao Chen. BoxInst: High-performance instance segmentation with box annotations. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2021.

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[36] Zhenyu Wang, Yali Li, and Shengjin Wang. Noisy boundaries: Lemon or lemonade for semi-supervised instance segmentation? *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16805–16814, 2022.