BoxSnake: Polygonal Instance Segmentation with Box Supervision

Rui Yang[1*‡] Lin Song[2*†] Yixiao Ge[2] Xiu Li[1†]
[1]Tsinghua Shenzhen International Graduate School, Tsinghua University [2]Tencent AI Lab
rayyang0116@gmail.com {ronnysong, yixiaoge}@tencent.com li.xiu@sz.tsinghua.edu.cn

Abstract

Box-supervised instance segmentation has gained much attention as it requires only simple box annotations instead of costly mask or polygon annotations. However, existing box-supervised instance segmentation models mainly focus on mask-based frameworks. We propose a new end-to-end training technique, termed BoxSnake, to achieve effective polygonal instance segmentation using only box annotations for the first time. Our method consists of two loss functions: (1) a point-based unary loss that constrains the bounding box of predicted polygons to achieve coarse-grained segmentation; and (2) a distance-aware pairwise loss that encourages the predicted polygons to fit the object boundaries. Compared with the mask-based weakly-supervised methods, BoxSnake further reduces the performance gap between the predicted segmentation and the bounding box, and shows significant superiority on the Cityscapes dataset. The source code has been available at https://github.com/Yangr116/BoxSnake.

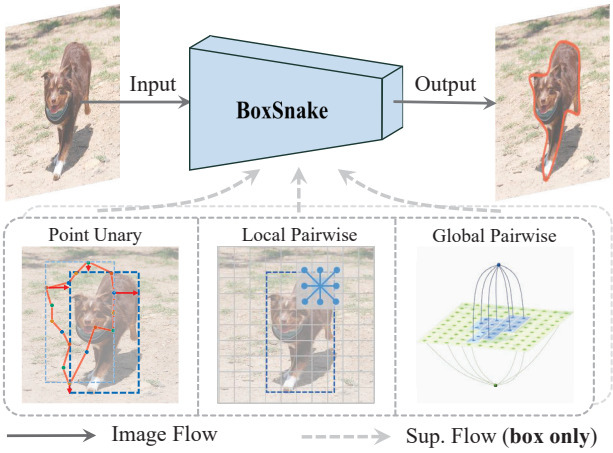


Figure 1. BoxSnake is a box-supervised instance segmentation model that predicts the segmentation of the interested object in the form of polygons. Three terms, involving a point-based unary term and two pairwise terms, are proposed to constrain the predicted polygon to fit the object boundary. The grey dotted line indicates that the proposed losses only work during training.

1. Introduction

Instance segmentation aims to provide precious and fine-grained object localization, which plays a fundamental role in various tasks, such as image understanding, autonomous driving, and robotic grasping. There are two primary paradigms for advanced instance segmentation: mask-based [26, 6, 12, 66, 71, 80, 40] and polygon-based [41, 72, 37, 81, 54]. Mask-based instance segmentation employs pixel-wise masks to represent the objects of interest, while polygon-based instance segmentation utilizes object contours, consisting of a set of vertices along the object boundaries [41, 37, 54] or a center point with a group of rays [72]. Nevertheless, the laborious and costly process of mask or polygon annotation [34, 19, 4] impedes the widespread and universal real-world applications of these methods.

Recent research efforts [15, 34, 28, 67, 39] aim to overcome this obstacle by obtaining instance masks solely through box annotations. For example, BoxSup [15] and Box2Seg [34] employ pseudo mask labels from GrabCut [56] or MCG [3] to train the networks iteratively. BBTP [28] and BoxInst [67] propose an end-to-end mask-based framework utilizing multi-instance learning (MIL) and pairwise affinity modeling. Additionally, BoxLevelSet [39] uses the Chan-Vese level-set energy function [10] to predict instance-aware mask maps as an implicit level-set evolution. However, there is no deep-learning-based method for weakly-supervised polygonal instance segmentation. Therefore, we attempt to explore a new perspective: *Can effective polygon-based instance segmentation be achieved with box annotations only?*

To achieve it, we propose a new end-to-end training technique, termed BoxSnake, with a point-based unary loss and a distance-aware pairwise loss. First, similar to the mask-based methods [15, 67, 39], we argue that all vertices of the expected polygon ought to be tightly enclosed by the bounding box. Thus, we design a point-based unary loss relying on CIoU [83] to constrain the bounding box of the predicted

*Equal contribution. ‡ Work done during an internship at Tencent.
†Corresponding author.

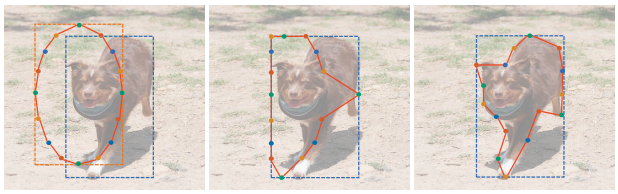
polygon by maximizing its Intersection-over-Union (IoU) with the annotation box. As shown in Figure 2 (b), since the point-based unary loss only optimizes the outermost vertices of the predicted polygon, it can roughly regress to the object of interest but is hard to fit the boundary well.

To address the above issue, we further introduce a pairwise loss based on distance transformation, including a local pairwise term and a global pairwise term. Specifically, as shown in Figure 1, motivated by the weakly-supervised methods based on masks [28, 67, 39], we propose a local-pairwise loss to encourage the predicted polygon not to fall into flat areas. However, compared with mask-based methods, it is difficult to directly optimize the coordinates of polygon vertices. Therefore, we attempt to convert the coordinate regression problem into a classification problem. To approach this, we introduce a hard mapping function based on the curve evolution method [8, 52] to transform the 2D polygon into a 3D plane, which maps the pixels in the interior and exterior of the polygon to two separated level sets. We further use the distance transformation from pixels to predicted polygons to relax the discrete process in the mapping function, enabling end-to-end training of the network. Based on it, the local-pairwise loss encourages the consistency between neighboring pixels in a local window, ensuring that two nearby pixels in the 3D planes are likely to appear on the same level set if they have similar colors. In addition, we further propose a global-pairwise loss to minimize the variance of pixel colors in the same level set, which can better fit the predicted polygon to the object boundary. Besides, it makes the predicted polygon more smooth and more robust to the noise in a local region of the image.

In summary, our contributions lie in the following:

- We design a novel end-to-end training technique to approach polygonal instance segmentation with only box supervision for the first time.
- We introduce a point-based unary loss that regularizes the predicted polygon to objects using box-based IoU.
- We propose a distanced-based pairwise loss involving local and global terms to encourage the predicted polygon to align with object boundaries. More importantly, we devise a method that transforms the polygon regression problem into a classification problem, thereby facilitating the pairwise loss on polygonal segmentation.

We apply the proposed techniques to the state-of-the-art polygon-based framework [37] and achieve competitive performance on COCO [43] and Cityscapes [14] datasets. Compared with the mask-based weakly-supervised counterparts, our method can further narrow the performance gap between the predicted segmentation and the bounding box.With ResNet-50 backbone, our method obtains 3.9% absolute gains over the BoxInst [67] on Cityscapes dataset and shows significant superiority over some fully-supervised methods on COCO dataset, including DeepSnake [54] and PolarMask [72].



(a) image (b) w/ $\mathcal{L}_u$ (c) w/ $\mathcal{L}_u$, $\mathcal{L}_{lp}$ and $\mathcal{L}_{gp}$

Figure 2. Impacts of the different losses. (a) indicates the initial polygon sampled from an ellipse enclosed by the predicted box. (b) denotes the predicted polygon supervised by the point-based unary loss only. (c) is the predicted polygon jointly supervised by the point-based unary loss and the distance-aware pairwise loss.

2. Related Work

Mask-based Instance Segmentation aims to represent individual objects with pixel-level binary masks. The pioneering Mask R-CNN [26] resorted to the foreground-background segmentation within each pre-detected bounding box (object proposal). Follow-ups focused on exploiting cascade structure to find more precise boxes [6, 11, 69, 61, 74] or improving the coarse boundaries [32, 13, 31, 58, 62, 79, 29, 24, 23]. Kernel-based methods [66, 5, 71, 53, 82] generated instance masks from dynamic kernels without dependence on box detection, which achieves sound performance with high efficiency. Inspired by an end-to-end set prediction framework (e.g., DETR [7]), query-based methods [21, 17, 12] tackled instance segmentation via a fixed number of learnable embeddings, where each embedding, the prototype of an instance, can decode a binary mask and its category from feature maps. In summary, the above methods group pixels of each instance by a spatially dense function that performs a pixel-wise classification and binarization (always using a threshold of 0.5).

Polygon-based Instance Segmentation instead represents each object instance with geometrical contours directly. This approach dates back to Snakes or active contours method [30, 73] in the 1980s, which deformed an initial outline to fit the object silhouette. With the rise of deep learning, several approaches have been proposed to trace object boundaries. For instance, Polygon RNN [9, 1] employed a CNN-RNN architecture to sequentially trace object boundaries in a given image patch. Two-stage Deep Snake [54] created initial octagon contours using a detector and then iteratively deformed them through a circular convolution network. PolyTransform [41] generated masks for each object using an off-the-shelf mask-based segmentation pipeline and converted the resulting mask contours into a set of vertices. Subsequently, the Transformer [68, 75] wrapped these vertices to fit the object silhouette better. Curve GCN [44] regarded the initial contour as a graph and used a graph convolutional network to predict vertex-wise

offsets. It employed a differentiable rendering loss to ensure that masks rendered from the predicted points agreed with the ground-truth masks. BoundaryFormer [37], on the other hand, applied a differentiable rasterization method to generate masks from polygons, achieving stunning results. PolarMask [72] and its follow-ups [48, 50] adopted a set of rays in the polar coordinate system to represent object contours, which enables an efficient calculation of Intersection-over-Union. However, the deep learning-based methods mentioned above require expensive ground-truth masks or polygons, which hinders their practical applicability and extension. Alternatively, the proposed BoxSnake can produce the object polygon with only cheap box annotations.

Box-supervised Instance Segmentation is a workaround for fully-supervised methods, which has been explored in traditional interactive segmentation [56, 64, 38]. In the context of deep learning, many arts [28, 67, 36, 39, 78, 25] tried to perform mask-based instance segmentation with just bounding-box annotations. BBTP [28] converted the box tightness prior [38] as the latent ground truth via multiple instance learning (MIL) and employed the structural constraint to maintain the piece-wise smoothness in predicted masks. BoxInst [67] achieved stunning instance segmentation results by substituting the mask loss with projection and pairwise losses in CondInst [66]. DiscoBox [36] further leveraged cross-image correspondence to enhance pairwise affinity, thus improving segmentation performance. The above methods can be summarized as a CRF energy model [33], where the unary potential is responsible for finding the initial instance mask (seeds) and the pairwise potential for label propagation. Similar appearance models [35, 20] are also applied in the partially supervised instance segmentation. Moreover, based on the Chan-Vese level set energy function [10], BoxLevelSet [39] evolved the instance mask through low-level image features and tree-filter [60, 59] refined high-level features within the object bounding box. By contrast, we in this paper formulate a method to train the polygon-based instance segmentation frameworks with only box annotations.

3. BoxSnake

Traditional Snakes or active contours method [30, 73, 8] can obtain object boundaries by coarsely annotating the object region and numerically minimizing a hand-crafted energy function. However, there is no deep-learning-based method for polygon-based instance segmentation with just box annotations. We in this paper propose the BoxSnake, a novel deep learning-based framework that aims to solve polygonal instance segmentation with only bounding-box supervision. To supervise BoxSnake using boxes, we formulate two loss functions, namely the point-based unary loss and the distance-aware pairwise loss, to guide the predicted polygon to fit the object boundaries accurately.

3.1. Definition

Given an input image $\mathcal{I} \in R^{H\times W\times 3}$ with the resolution of $H \times W$ and N interested objects, the set of pixels in the image is denoted by Ω. The BoxSnake predicts a polygon for each object, where each polygon contains K ordered vertices, sorted counterclockwise according to their initial angles. It represents the outline of an object, where each pair of adjacent vertices can be linked as a segment. For the n-th interested object, the predicted polygon is denoted as $\mathcal{C}^n = \{(x_i^n, y_i^n)\}_{i=1}^K$ and its bounding-box annotation is b^n. For simplicity, we will omit n in the following.

3.2. Point-based Unary Loss

The point unary loss is designed to ensure that all the vertices of the predicted polygon are enclosed within the ground-truth bounding box. Given a predicted polygon $\mathcal{C}$ and its ground-truth bounding box b, we can easily calculate the bounding box of $\mathcal{C}$ using the $\max$ and $\min$ operation along with the x- and y-axis:

$$(x_1, y_1) = \min(\mathcal{C}), \quad (x_2, y_2) = \max(\mathcal{C}), \tag{1}$$

where (x_1, y_1) and (x_2, y_2) are the top left and bottom right coordinates of the bounding box b_c, respectively. Then, the discrepancy between b_c and b is minimized by the point-based unary loss:

$$\mathcal{L}_u = 1 - CIoU(b_c, b), \tag{2}$$

where $CIoU(\cdot,\cdot)$ represents the complete intersection over union [83]. This loss term encourages the tightest box covering the predicted polygon matches its ground-truth bounding box exactly. As reported in the experiments (Table 7), with the unary loss only, BoxSnake demonstrates reasonable instance segmentation performance.

3.3. Distance-aware Pairwise Loss

Nevertheless, as illustrated in Figure 2 (b) and Figure 5 (b), only the point-based unary loss fails to fit the object boundary well. Therefore, we propose a distance-aware pairwise loss involving local and global pairwise terms.

Local Pairwise Term. Object boundaries are typically located in regions with local color variation in the image [22]. According to this hypothesis, we propose a local pairwise loss based on windows to encourage predicted polygons to be locally consistent with the positions of the image edges. However, compared with mask-based methods [67], it is difficult to directly optimize the coordinates of polygon vertices. Therefore, we attempt to convert the coordinate regression problem into a classification problem.

As shown in Figure 3 (b), we introduce the curve evolution [8, 52] method to reformulate the predicted polygon $\mathcal{C}$ to a 3D plane, which maps the pixels inside and outside the polygon into two separate level sets. Specifically, given a

pixel at location (x, y) in a 2D image, we define the curve evolution process as a discrete function $\mathcal{U}_{\mathcal{C}}(x, y) \in \{0, 1\}$. The $\mathcal{U}_{\mathcal{C}}(x, y) = 1$ if the pixel is inside the polygon, and $\mathcal{U}_{\mathcal{C}}(x, y) = 0$ if it is outside the polygon. The curve evolution function can be easily implemented by the point-in-polygon (PIP) algorithm [57, 63]. With the above techniques, the constraint of consistency between the polygon and the image is transformed into similarly colored pixel points located in the same level set. This process can be formulated as minimizing the local-pairwise energy:

$$E = \sum_{(p,q)\in\mathring{\Omega}_k(i,j)} w_{[(i,j),(p,q)]} \mid \mathcal{U}_{\mathcal{C}}(i, j) - \mathcal{U}_{\mathcal{C}}(p, q) \mid, \quad (3)$$

where $\mathring{\Omega}_k(i, j)$ means the adjacent pixels within a $k \times k$ window at the position (i, j). $w_{[(i,j),(p,q)]}$ measures the affinity of two pixels by color distance:

$$w_{[(i,j),(p,q)]} = exp\left(-\frac{\parallel I(i, j) - I(p, q) \parallel_2}{2\sigma_I^2}\right), \quad (4)$$

where $I(\cdot, \cdot)$ indicates the color value at the input coordinate, $\parallel \cdot \parallel_2$ is Euclidean distance, and σ_I is a hyper-parameter for temperature. Eq. 4 tends to be zeros at the edges. If two adjacent pixels have a high color similarity but are assigned to different level sets, Eq. 3 will give them a high penalty, and vice visa.

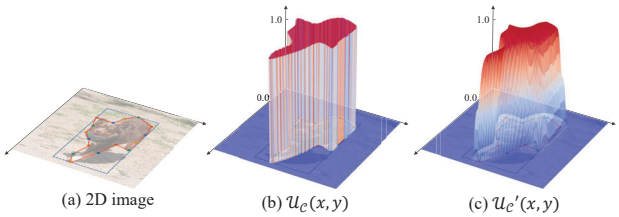
However, the mapping function $\mathcal{U}_{\mathcal{C}}(\cdot, \cdot)$ in Eq. 3 is a discrete and non-differentiable function, making the energy can not be trained in an end-to-end manner for deep neural networks. To solve this issue, we introduce a distance transformation process to relax the mapping function into a continuous and differentiable one. Specifically, we calculate the minimum vertical distance from a pixel (x, y) to the predicted polygon as $D_{\mathcal{C}}(x, y)$, which reflects the distance from the exported object boundary. We further apply the Sigmoid function to normalize the distance to $(0, 1)$. The approximate mapping function can be formulated as:

$$\mathcal{U}'_{\mathcal{C}}(x, y) = \sigma\left(\frac{2 \cdot (\mathcal{U}_{\mathcal{C}}(x, y) - 0.5) \cdot D_{\mathcal{C}}(x, y)}{\tau}\right), \quad (5)$$

where τ denotes the temperature hyper-parameter for Sigmoid operation $\sigma(\cdot)$. As illustrated in Figure 3 (c), the approximate mapping function is continuous at the polygon boundaries and is differentiable w.r.t. the coordinates of the vertices. To this end, we propose a local-pairwise loss:

$$\mathcal{L}_{lp} = \sum_{(p,q)\in\mathring{\Omega}_k(i,j)} w_{[(i,j),(p,q)]} \mid \mathcal{U}'_{\mathcal{C}}(i, j) - \mathcal{U}'_{\mathcal{C}}(p, q) \mid, \quad (6)$$

which encourages similar-colored pixels within a local region to be located on the same level set and have consistent



(a) 2D image (b) $\mathcal{U}_{\mathcal{C}}(x, y)$ (c) $\mathcal{U}_{\mathcal{C}}{}'(x, y)$

Figure 3. The diagram of distance relaxation. (a) is a predicted polygon on a 2D image. (b) is the hard mapping function to transform the polygon to a 3D plane with two separate level sets. (c) is the approximate mapping function.

distances to the object boundary. At the first glance, the local pairwise loss could potentially lead the network to have two trivial results, i.e., the predicted polygon may expand to the entire image or collapse to a single point. However, these trivial results can be avoided by integrating the proposed point-based unary loss. The unary loss ensures the polygon is inside the ground-truth box, thus preventing the polygon from expanding to the whole image. Additionally, it encourages the area of the bounding box of the polygon to match the object annotation box, preventing it from collapsing into a single point.

Global Pairwise Term. Since color variations in a local region of the image may be noise, training with a local-pairwise loss may lead to unexpected segmentation boundaries. Therefore, we further propose a global pairwise loss to reduce the influence of local noise. It is designed based on a hypothesis, i.e., internal or external regions of the object should be nearly homogeneous [10], which is formulated as:

$$\begin{aligned}\mathcal{L}_{gp} = &\sum_{(x,y)\in\Omega} ||I(x, y) - u_{in}||_2 \cdot \mathcal{U}'_{\mathcal{C}}(x, y) \\ &+ \sum_{(x,y)\in\Omega} ||I(x, y) - u_{out}||_2 \cdot (1 - \mathcal{U}'_{\mathcal{C}}(x, y)),\end{aligned} \quad (7)$$

where u_{in} and u_{out} indicate the average image color inside and outside the predicted polygon, respectively. The u_{in} and u_{out} are defined as:

$$\begin{aligned}u_{in} &= \frac{\sum_{(x,y)\in\Omega} \mathcal{I}(x, y) \cdot \mathcal{U}'_{\mathcal{C}}(x, y)}{\sum_{(x,y)\in\Omega} \mathcal{U}'_{\mathcal{C}}(x, y)}, \\ u_{out} &= \frac{\sum_{(x,y)\in\Omega} \mathcal{I}(x, y) \cdot (1 - \mathcal{U}'_{\mathcal{C}}(x, y))}{\sum_{(x,y)\in\Omega} (1 - \mathcal{U}'_{\mathcal{C}}(x, y))},\end{aligned} \quad (8)$$

which is modulated by the approximate mapping function. As shown in Figure 2 (c) and Figure 5 (d), the global-pairwise loss typically makes the predicted polygon more smooth and better fit the object boundary.

Clipping Strategy. The $\mathcal{L}_{lp}$ and $\mathcal{L}_{gp}$ need to involve the background information. However, calculating these loss terms on all the background pixels directly may not be practical due to potential memory constraints. To address this

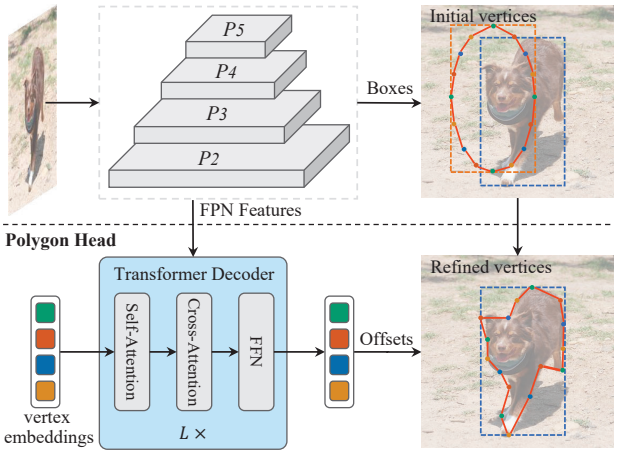


Figure 4. The network architecture of BoxSnake. The multi-scale features are extracted from the input image by a backbone network. The box predictor is attached to these features to obtain bounding boxes. The polygon head predicts the polygon for each box, which is trained with box annotation only.

issue, we resize the predicted polygon to make the size of its bounding box to be $S \times S$ by using a bilinear interpolation. We further employ the RoIAlign [26] operation to crop and resize the image to the size of $S \times S$, according to the coordinates of the ground-truth box. Accordingly, we use the cropped image as guidance for the pairwise losses. This strategy reduces the memory requirements during the training phase, making BoxSnake more practical for users with limited computational resources.

So far, we have integrated the proposed losses to jointly supervise the network to predict accurate object polygons with box supervision only:

$$\mathcal{L}_{polygon} = \alpha\mathcal{L}_u + \beta\mathcal{L}_{lp} + \gamma\mathcal{L}_{gp}, \tag{9}$$

where α, β, and γ are the modulated weights for each loss term. During training, $\mathcal{L}_u$ ensures the polygon is tightly enclosed by the ground-truth box, while $\mathcal{L}_{lp}$ and $\mathcal{L}_{gp}$ further fit the predicted polygon to the object boundary.

3.4. Network Architecture

The proposed training technique is flexible and easy to use as a plug-and-play training module. As same as the BoundaryFormer [37], we apply our method to the Mask R-CNN [26] framework and use a Transformer as the polygon head, which is shown in Figure 4. A backbone network and feature pyramid network [42] are used to extract multi-scale feature maps from the input image. The box regression and classification head generate object bounding boxes and corresponding categories from each scale. Different from the BoundaryFormer, we replace the mask-supervised loss function with the proposed weakly-supervised losses. Besides, the polygon head predicts the polygon by regressing the relative coordinates of polygon vertices. It is made up of L Transformer decoders and each Transformer decoder is consisting of vanilla self-attention [68], deformable cross-attention [84], and feed-forward modules. Following the previous literature [41, 54, 37], the vertices of initialized polygons are sampled from an ellipse enclosed by the bounding box. They are further refined by Transformer decoders iteratively and generate the final polygon prediction.

4. Experiments

To prove effectiveness of BoxSnake, we conduct experiments on COCO [43] and Cityscapes [14] datasets. For COCO, the models are trained on train2017 set with 115K images. The ablation experiments are evaluated on val2017 set with 5K images, and the large-backbone results are reported on test-dev set with 20K images. For Cityscapes, we train and evaluate the models on the fine part, consisting of $2,975$ train and 500 validation images with a high resolution and annotation quality. Notably, just bounding-box annotations are enabled during training.

4.1. Implementation Details

We employ Mask R-CNN [26] as the underlying detector whose FPN [42] features attach the polygon head. We represent each polygon using 64 vertices and employ 4 Transformer decoders to refine the initial vertices. Different from BoundaryFormer [37], we predict the polygon in the entire scope instead of within the predicted bounding box. This eliminates the need for an additional alignment strategy, and the predicted polygon is not constrained to the predicted box. To balance the different loss terms, we set the weights $\alpha = 1.0$, $\beta = 0.5$, and $\gamma = 0.03$ in Eq. 9. Regarding the distance-aware pairwise loss, we use a clipping size of 72×72, including a 64×64 grid map with 4 zero padding on each side and a temperature of 0.1 in Eq 5. For the local pairwise term (Eq. 6), we compute the pairwise relationship in 3×3 windows with a dilation rate of 2 and set σ_I to 1.0. In addition, the bounding box classification and regression losses are the same as those in Mask R-CNN.

Unless otherwise specified, we train and infer models similar to Mask R-CNN. ResNet [27] and Swin Transformer [46] are employed as the backbone, which is initialized with weights pre-trained on ImageNet [16]. The polygon head is initialized as [84], and other new layers are initialized as in Mask R-CNN. We optimize all models using AdamW [47]. On COCO, we train the models for 90K ($1\times$) and 180K ($2\times$) iterations with a batch size of 16 on 8 GPUs. The initial learning rate is 1×10^{-4}, and the weight decay is 0.1. For the 90K schedule, the learning rate is decreased by a factor of 10 at steps 60K and 80K, while for the 180K schedule, it is decreased at steps 120K and 160K. Moreover, we apply random flipping and scale jittering augmentation. For the ResNet and Swin Transformer backbones, we randomly sample the short side of training images from

method	backbone	out	AP ↑	AP_{50} ↑	ΔAP_b ↓
fully-supervised methods					
Mask R-CNN [26]	R50-FPN	$\mathcal{M}$	35.2	56.3	-
CondInst [66]	R50-FPN	$\mathcal{M}$	35.6	56.4	-
PolarMask [72]	R50-FPN	$\mathcal{C}$	29.1	49.5	-
Deep Snake [54]	DLA-34	$\mathcal{C}$	30.5	–	-
DANCE [45]	R50-FPN	$\mathcal{C}$	34.5	55.3	-
BoundaryFormer [37]	R50-FPN	$\mathcal{C}$	36.1	56.7	-
box-supervised methods					
DiscoBox [36]	R50-FPN	$\mathcal{M}$	30.7	52.6	10.7
BoxInst [67]	R50-FPN	$\mathcal{M}$	30.7	52.2	8.7
BoxSnake	R50-FPN	$\mathcal{C}$	**31.1**	**53.4**	**7.8**
BBTP [28]	R101-FPN	$\mathcal{M}$	21.1	45.5	19.3
BoxCaseg [70]	R101-FPN	$\mathcal{M}$	30.9	53.7	9.1
BoxInst [67]	R101-FPN	$\mathcal{M}$	31.6	54.0	9.8
BoxSnake	R101-FPN	$\mathcal{C}$	**31.6**	**54.0**	**8.3**

Table 1. Comparisons with classical instance segmentation methods on COCO val2017 set. All models are trained with the $1\times$ schedule. ΔAP_b indicates the accuracy gap between the predicted bounding box and segmentation. $\mathcal{M}$ and $\mathcal{C}$ denote the segmentation formats being mask and polygon, respectively.

method	backbone	out	AP	AP_{50}
fully-supervised methods				
Mask R-CNN [26]	R50-FPN	$\mathcal{M}$	31.5	–
CondInst [66]	R50-FPN	$\mathcal{M}$	33.1	–
E2EC [81]	DLA-34	$\mathcal{C}$	34.0	–
BoundaryFormer [37]	R50-FPN	$\mathcal{C}$	34.7	60.8
box-supervised methods				
BoxInst [67]	R50-FPN	$\mathcal{M}$	22.4	49.0
AsyInst [77]	R50-FPN	$\mathcal{M}$	24.7	53.0
BoxSnake	R50-FPN	$\mathcal{C}$	**26.3**	**54.2**

Table 2. Results on Cityscapes validation set. $\mathcal{M}$ and $\mathcal{C}$ denote the segmentation formats being mask and polygon, respectively. DLA-34 refers to the backbone used in [18]. The reported results of BoxInst are obtained from the official repository [65].

$[640, 800]$ and $[480, 800]$, respectively. During inference, the short side is set to 800 pixels. On Cityscapes, our models are trained for 24K iterations using a batch size of 8 on 8 GPUs. The initial learning rate is set to 1×10^{-4} and is subsequently reduced to 1×10^{-5} at 18K iterations. The short size of the training images is randomly resized within the range of $[800, 1024]$, while the long size is kept at most 2048. During inference, the short size is set to 1024 pixels. The performance is evaluated using the COCO-format mask AP on two benchmarks.

4.2. Main Results

To demonstrate the effectiveness of our BoxSnake, we compare our BoxSnake with fully-supervised and box-supervised instance segmentation approaches on COCO val2017 set and Cityscapes validation set.

Results on COCO. As reported in Table 1, BoxSnake achieves results better than or comparable to those of mask-based instance segmentation methods using only box annotations. Specifically, BoxSnake attains 31.1% mask AP with the ResNet-50 backbone and $1\times$ schedule, outperforming both BoxInst [67] and DiscoBox [36] by 0.4% mask AP. When combined with the ResNet-101 backbone, our BoxSnake achieves 31.6% AP, which significantly surpasses BBTP [28] by 10.5% mask AP. Notably, BoxSnake greatly reduces the accuracy gap between the predicted box and polygon. This gap is $\sim$ 8% AP for our method but $\sim$ 10% AP for BoxInst and DiscoBox. Additionally, without mask or polygon annotations, BoxSnake even achieves better performance than a few fully supervised polygonal instance segmentation methods. For example, when using ResNet-50, BoxSnake surpasses PolarMask [72] and Deep Snake [54] by 2.0% and 0.6% mask AP, respectively. Some qualitative results are shown in Figure 5 (e), where the polygon is aligned with the object boundaries well. This result demonstrates the great potential of the polygonal instance segmentation with box annotations.

Results on Cityscapes. To demonstrate our BoxSnake can generalize beyond the COCO dataset, we conduct experiments on Cityscapes benchmark [14]. As presented in Table 2, our BoxSnake outperforms BoxInst [67] and AsyInst [77] by a significant margin. Specifically, BoxSnake achieves 26.3% mask AP, which surpasses the BoxInst and AsyInst by 3.9% and 1.6% mask AP, respectively. This superiority could be derived from a fact, i.e., Cityscapes dataset has more vehicle instances without holes. As shown in Figure 6, BoxInst has an ambiguous boundary at the shadow. By contrast, our BoxSnake presents a fine and clear boundary between the vehicle and the road since the polygon-based framework could learn some shape priors [41]. This excellent performance reveals the tremendous potential of the box-based polygonal instance segmentation.

4.3. Ablation Studies

We conduct ablation experiments on COCO val2017 set to verify the effectiveness of BoxSnake. All models use the ResNet-50 backbone and $1\times$ schedule in default, except exploring the upper bound with the large backbone.

Different unary loss. As mentation before, the unary loss plays a crucial role in ensuring that all vertices of the predicted polygon lie within the ground-truth box, thereby avoiding potential trivial solutions from the distance-aware pairwise loss. We conduct experiments to investigate the efficacy of different unary losses, as presented in Table 3. 'Dice on P_3' represents the approach as BoxInst [67] that minimizes the discrepancy between the projected level-set map $\mathcal{U}'_{\mathcal{C}}(x, y)$ and projected box mask using Dice loss [49], where the size of the level-set map is same as P_3. This method obtains 19.3% mask AP since the max projection on the level-set map selects the points that fall in the saturated zone of the Sigmoid operation (the gradient could vanish). By contrast, GIoU [55] and CIoU [83] loss works for vertices directly by maximizing the IoU between the cir-

(a) ground truth (b) w/ $\mathcal{L}_u$ (c) w/ $\mathcal{L}_u$ and $\mathcal{L}_{lp}$ (d) w/ $\mathcal{L}_u$ and $\mathcal{L}_{gp}$ (e) w/ $\mathcal{L}_u$, $\mathcal{L}_{lp}$ and $\mathcal{L}_{gp}$

Figure 5. Qualitative results of different loss terms on COCO val2017 set. $\mathcal{L}_u$, $\mathcal{L}_{lp}$ and $\mathcal{L}_{gp}$ refer to the unary loss (Eq. 2), the local pairwise loss (Eq. 6) and the global pairwise loss (Eq.7), respectively. The pairwise losses can enable predictions to align with boundaries.

cumscribed boxes of polygons and their ground-truth boxes. As a result, they yield $\sim 4\%$ AP gains over 'Dice on P_3'.

Varying the window size. Local pairwise term encourages two nearby pixels with a similar color to lie in the same level set. The window size determines the number of neighboring pixels to compute the local pairwise loss with each pixel. Inspired by [2], the receptive field of the kernel can be expanded by the dilation trick. As reported in Table 6, varying the window size brings minor fluctuations in performance ($\sim 0.4\%$ mask AP).

Effectiveness of clipping strategy. The resolution of $\mathcal{U}'_{\mathcal{C}}(x, y)$ influences distance-aware pairwise loss since this loss builds the relationship between each pixel and its neighboring pixels. As shown in Table 4, increasing the resolution from P_3's size to P_2's size, the performance is boosted from 29.6% to 29.8% mask AP. Nevertheless, the distance-aware pairwise loss is mainly contingent on the background pixels surrounding the ground-truth box because background pixels can propagate zero-level set signals into the box. In light of this, a clipping strategy is employed, which brings considerable improvement by 1.5% mask AP. Notably, this strategy is greatly beneficial for small instances, as presented in the fifth column.

Different initial methods. The polygon head evolves a set of initial vertices by predicting 2D offsets for each vertex (§ 3.4). An appropriate initial status could impact the evolutionary process, as demonstrated by [45, 41]. As detailed in Table 5, we initialize the polygon with the square or elliptical format, where the latter outperforms the former 0.4% mask AP. Additionally, as reported in Table 7, taking the inscribed ellipse as the prediction can obtain 15.5 mask AP.

The effect of each loss term. We ablate the effect of each loss in Table 7. By using the point-based unary loss alone, BoxSnake is capable of obtaining a basic result (23.9% mask AP), demonstrating a much finer location than boxes (10.6% mask AP) and ellipses (15.5% mask AP). As shown in Figure 5 (b), the predicted polygon fits the object boundaries coarsely. Integration of the pairwise loss can further enhance the quality of predicted polygons, indicating that the pairwise loss indeed attracts the predicted polygon to the object boundaries. Specifically, the local and global pairwise terms bring 9.6% and 8.6% mask AP_{75} gains. Their related qualitative results are shown in Figure 5 (c) and (d), respectively, where the predicted polygons are attracted to the object boundaries. The integration of point-based unary loss and distance-aware pairwise loss elevates the performance of BoxSnake to 31.1% mask AP.

Large Backbone. To explore the upper bound of BoxSnake, we adopt larger backbones and evaluate their results on COCO test-dev set. BoxSnake attains 32.2% mask AP with ResNet-101 and $2\times$ training schedule. When equipped with Swin-B [46], the performance can be promoted to

(a) BoxInst (b) BoxSnake (**ours**)

Figure 6. Qualitative comparisons on Cityscapes validation set. The major difference is marked by the green rectangle.

unary loss	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Dice on P_3	19.3	43.6	14.5	7.3	19.9	30.6
GIoU [55]	23.7	48.6	20.7	11.9	24.4	33.7
CIoU [83]	23.9	48.8	21.3	12.4	24.6	34.4

Table 3. Ablation study for different unary losses on COCO val2017 set. Only the unary loss is employed for training. 'Dice on P_3' refers to the method proposed by BoxInst [67], which uses the Dice loss [49] to minimize the discrepancy between the projected level-set map and annotation box.

method	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
full-supervised methods						
P_3	33.2	53.6	34.8	12.1	36.2	52.8
P_2	34.8	54.9	36.7	14.3	37.4	53.0
Clipping Strategy	36.4	57.2	39.0	19.6	38.6	47.9
box-supervised methods						
P_3	29.6	52.3	29.4	13.2	31.5	44.6
P_2	29.8	52.7	29.2	13.6	31.8	44.7
Clipping Strategy	31.1	53.4	31.3	14.6	33.5	46.7

Table 4. Ablation study for clipping strategy (§ 3.3) on COCO val2017 set. 'P_3' and 'P_2' denote that the predicted polygon is scaled to the size of P_3 and P_2, respectively.

initial method	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
square	30.7	53.5	30.6	14.1	32.9	46.2
ellipse	31.1	53.4	31.3	14.6	33.5	46.7

Table 5. Ablation study for initial polygon on COCO val2017 set.

38.5% mask AP. Moreover, with Swin-L [46], the upper bound can be pushed further to 39.5% mask AP. This result demonstrates a bright prospect of the polygon-based instance segmentation using just box supervision.

size	dilation	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
3×3	1	30.8	53.3	30.8	13.4	33.0	46.5
3×3	2	31.1	53.4	31.3	14.6	33.5	46.7
5×5	1	30.9	53.2	30.9	14.4	33.0	46.3

Table 6. Ablation study for the window size in Eq. 6 on COCO val2017 set. The different window size in the local-pairwise loss brings marginal fluctuations.

$\mathcal{L}_u$	$\mathcal{L}_{lp}$	$\mathcal{L}_{gp}$	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
box mask			10.6	32.2	4.6	5.7	11.3	15.6
ellipse mask			15.5	39.4	10.1	9.5	16.3	21.5
✓			23.9	48.8	21.3	12.4	24.6	34.4
✓	✓		30.8	52.8	30.7	13.7	33.1	46.3
✓		✓	29.8	53.2	29.9	13.9	31.5	44.8
✓	✓	✓	31.1	53.4	31.3	14.6	33.5	46.7

Table 7. Ablation study for different loss terms on COCO val2017 set. 'box mask' and 'ellipse mask' denote the results from square and ellipse initialization, respectively. The unary loss improves the recognition of objects, and the pairwise losses greatly improve the boundary accuracy.

method	backbone	architecture	out	AP	AP_{50}	AP_{75}
BoxInst [67]	R50	CondInst [66]	$\mathcal{M}$	32.1	55.1	32.4
DiscoBox [36]	R50	SOLOv2 [71]	$\mathcal{M}$	32.0	53.3	32.6
BoxInst [67]	R101	CondInst [66]	$\mathcal{M}$	32.5	55.3	33.0
BoxLevelSet [39]	R101	SOLOv2 [71]	$\mathcal{M}$	33.4	56.8	34.1
BoxCaseg [70]	R101	M-RCNN [26]	$\mathcal{M}$	30.9	54.3	30.8
BoxSnake	R50	M-RCNN [26]	$\mathcal{C}$	31.6	54.8	31.5
BoxSnake	R101	M-RCNN [26]	$\mathcal{C}$	32.2	55.8	32.1
BoxSnake	Swin-B	M-RCNN [26]	$\mathcal{C}$	38.5	65.3	38.9
BoxSnake	Swin-L	M-RCNN [26]	$\mathcal{C}$	**39.5**	**66.8**	**39.9**

Table 8. Comparisons with state-of-the-art methods on COCO test-dev set. $\mathcal{M}$ and $\mathcal{C}$ denote the formats being mask and polygon, respectively. BoxSnake predicts polygon with box supervision, achieving comparable performance to mask-based methods.

5. Conclusion

This paper introduces a new end-to-end training technique for weakly-supervised instance segmentation based on polygons, utilizing only box annotations. Our method integrates a point-based unary loss and a distance-aware pairwise loss. The former maximizes the Intersection-over-Union between the circumscribed box of the predicted polygon and its ground-truth box, thereby making the predicted polygons around the target objects. The latter one, leveraging pixel affinities, encourages that the predicted polygons are better to fit the object boundary and are robust to the local noise. The proposed BoxSnake achieves competitive performance on both COCO and Cityscapes datasets, making an effective polygon-based instance segmentation with solely box supervision for the first time. In the future, it can be used as a tool in the AI system [51, 76] or a type of condition in the diffusion model.

Acknowledgments: This research was supported by the National Key R&D Program of China (Grant No. 2020AAA0108303), Shenzhen Science and Technology Project (Grant No. JCYJ20200109143041798), and Shenzhen Stable Supporting Program (WDZC20200820200655001).

References

[1] David Acuna, Huan Ling, Amlan Kar, and Sanja Fidler. Efficient interactive annotation of segmentation datasets with polygon-rnn++. In *CVPR*, 2018.

[2] Jiwoon Ahn and Suha Kwak. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In *CVPR*, 2018.

[3] Pablo Andrés Arbeláez, Jordi Pont-Tuset, Jonathan T. Barron, Ferran Marqués, and Jitendra Malik. Multiscale combinatorial grouping. In *CVPR*, 2014.

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

[5] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. YOLACT: real-time instance segmentation. In *ICCV*, 2019.

[6] Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: high quality object detection and instance segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(5):1483–1498, 2021.

[7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *ECCV*, 2020.

[8] Vicent Caselles, Ron Kimmel, and Guillermo Sapiro. Geodesic active contours. *Int. J. Comput. Vis.*, 22(1):61–79, 1997.

[9] Lluís Castrejón, Kaustav Kundu, Raquel Urtasun, and Sanja Fidler. Annotating object instances with a polygon-rnn. In *CVPR*, 2017.

[10] Tony F Chan and Luminita A Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, 2001.

[11] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. Hybrid task cascade for instance segmentation. In *CVPR*, 2019.

[12] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022.

[13] Tianheng Cheng, Xinggang Wang, Lichao Huang, and Wenyu Liu. Boundary-preserving mask R-CNN. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *ECCV*, 2020.

[14] 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.

[15] Jifeng Dai, Kaiming He, and Jian Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *ICCV*, 2015.

[16] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[17] Bin Dong, Fangao Zeng, Tiancai Wang, Xiangyu Zhang, and Yichen Wei. SOLQ: segmenting objects by learning queries. In *NIPS*, 2021.

[18] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *ICCV*, 2019.

[19] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The pascal visual object classes (VOC) challenge. *Int. J. Comput. Vis.*, 88(2):303–338, 2010.

[20] Qi Fan, Lei Ke, Wenjie Pei, Chi-Keung Tang, and Yu-Wing Tai. Commonality-parsing network across shape and appearance for partially supervised instance segmentation. In *ECCV*, 2020.

[21] Yuxin Fang, Shusheng Yang, Xinggang Wang, Yu Li, Chen Fang, Ying Shan, Bin Feng, and Wenyu Liu. Instances as queries. In *ICCV*, 2021.

[22] Rafael C Gonzalez. *Digital image processing*. Addison-Wesley Longman Publishing Co., Inc., 2009.

[23] Chunming He, Kai Li, Guoxia Xu, Jiangpeng Yan, Longxiang Tang, Yulun Zhang, Xiu Li, and Yaowei Wang. Hqg-net: Unpaired medical image enhancement with high-quality guidance. *arXiv preprint arXiv:2307.07829*, 2023.

[24] Chunming He, Kai Li, Yachao Zhang, Longxiang Tang, Yulun Zhang, Zhenhua Guo, and Xiu Li. Camouflaged object detection with feature decomposition and edge reconstruction. In *CVPR*, 2023.

[25] Chunming He, Kai Li, Yachao Zhang, Guoxia Xu, Longxiang Tang, Yulun Zhang, Zhenhua Guo, and Xiu Li. Weakly-supervised concealed object segmentation with sam-based pseudo labeling and multi-scale feature grouping. *arXiv preprint arXiv:2305.11003*, 2023.

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

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

[28] 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.

[29] Jianwen Jiang, Yu Cao, Lin Song, Shiwei Zhang, Yunkai Li, Ziyao Xu, Qian Wu, Chuang Gan, Chi Zhang, and Gang Yu. Human centric spatio-temporal action localization. In *ActivityNet Workshop on CVPR*, 2018.

[30] Michael Kass, Andrew P. Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *Int. J. Comput. Vis.*, 1(4):321–331, 1988.

[31] Lei Ke, Martin Danelljan, Xia Li, Yu-Wing Tai, Chi-Keung Tang, and Fisher Yu. Mask transfiner for high-quality instance segmentation. In *CVPR*, 2022.

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

[33] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NIPS*, 2011.

[34] Viveka Kulharia, Siddhartha Chandra, Amit Agrawal, Philip H. S. Torr, and Ambrish Tyagi. Box2seg: Attention weighted loss and discriminative feature learning for weakly supervised segmentation. In *ECCV*, 2020.

[35] Weicheng Kuo, Anelia Angelova, Jitendra Malik, and Tsung-Yi Lin. Shapemask: Learning to segment novel objects by refining shape priors. In *ICCV*, 2019.

[36] Shiyi Lan, Zhiding Yu, Christopher Choy, Subhashree Radhakrishnan, Guilin Liu, Yuke Zhu, Larry S Davis, and Anima Anandkumar. Discobox: Weakly supervised instance segmentation and semantic correspondence from box supervision. In *ICCV*, 2021.

[37] Justin Lazarow, Weijian Xu, and Zhuowen Tu. Instance segmentation with mask-supervised polygonal boundary transformers. In *CVPR*, 2022.

[38] Victor Lempitsky, Pushmeet Kohli, Carsten Rother, and Toby Sharp. Image segmentation with a bounding box prior. In *ICCV*, 2009.

[39] Wentong Li, Wenyu Liu, Jianke Zhu, Miaomiao Cui, Xian-Sheng Hua, and Lei Zhang. Box-supervised instance segmentation with level set evolution. In *ECCV*, 2022.

[40] Yanwei Li, Lin Song, Yukang Chen, Zeming Li, Xiangyu Zhang, Xingang Wang, and Jian Sun. Learning dynamic routing for semantic segmentation. In *CVPR*, 2020.

[41] Justin Liang, Namdar Homayounfar, Wei-Chiu Ma, Yuwen Xiong, Rui Hu, and Raquel Urtasun. Polytransform: Deep polygon transformer for instance segmentation. In *CVPR*, 2020.

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

[43] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *ECCV*, 2014.

[44] Huan Ling, Jun Gao, Amlan Kar, Wenzheng Chen, and Sanja Fidler. Fast interactive object annotation with curve-gcn. In *CVPR*, 2019.

[45] Zichen Liu, Jun Hao Liew, Xiangyu Chen, and Jiashi Feng. DANCE : A deep attentive contour model for efficient instance segmentation. In *WACV*, 2021.

[46] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021.

[47] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

[48] Feng Luo, Xiu Li, Bin-Bin Gao, and Jiangpeng Yan. A coarse-to-fine instance segmentation network with learning boundary representation. In *IJCNN*, 2021.

[49] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *3DV*, 2016.

[50] Eslam Mohamed, Abdelrahman Shaker, Hazem Rashed, Ahmad El Sallab, and Mayada Hadhoud. INSTA-YOLO: real-time instance segmentation. *arXiv:2102.06777*, 2021.

[51] OpenAI. Gpt-4 technical report, 2023.

[52] Stanley Osher and James A Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of computational physics*, 79(1):12–49, 1988.

[53] Yimin Ou, Rui Yang, Lufan Ma, Yong Liu, Jiangpeng Yan, Shang Xu, Chengjie Wang, and Xiu Li. Uniinst: Unique representation for end-to-end instance segmentation. *Neurocomputing*, 514:551–562, 2022.

[54] Sida Peng, Wen Jiang, Huaijin Pi, Xiuli Li, Hujun Bao, and Xiaowei Zhou. Deep snake for real-time instance segmentation. In *CVPR*, 2020.

[55] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian D. Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *CVPR*, 2019.

[56] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "grabcut": interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, 2004.

[57] Moshe Shimrat. Algorithm 112: position of point relative to polygon. *Communications of the ACM*, 5(8):434, 1962.

[58] Lin Song, Yanwei Li, Zhengkai Jiang, Zeming Li, Hongbin Sun, Jian Sun, and Nanning Zheng. Fine-grained dynamic head for object detection. *NIPS*, 2020.

[59] Lin Song, Yanwei Li, Zhengkai Jiang, Zeming Li, Xiangyu Zhang, Hongbin Sun, Jian Sun, and Nanning Zheng. Rethinking learnable tree filter for generic feature transform. In *NIPS*, 2020.

[60] Lin Song, Yanwei Li, Zeming Li, Gang Yu, Hongbin Sun, Jian Sun, and Nanning Zheng. Learnable tree filter for structure-preserving feature transform. *NIPS*, 2019.

[61] Lin Song, Songyang Zhang, Songtao Liu, Zeming Li, Xuming He, Hongbin Sun, Jian Sun, and Nanning Zheng. Dynamic grained encoder for vision transformers. *NIPS*, 2021.

[62] Lin Song, Shiwei Zhang, Gang Yu, and Hongbin Sun. Tacnet: Transition-aware context network for spatio-temporal action detection. In *CVPR*, 2019.

[63] Daniel Sunday. *Practical Geometry Algorithms with C++ Code*. Daniel Sunday, 2021.

[64] Meng Tang, Lena Gorelick, Olga Veksler, and Yuri Boykov. Grabcut in one cut. In *ICCV*, 2013.

[65] Zhi Tian, Hao Chen, Xinlong Wang, Yuliang Liu, and Chunhua Shen. AdelaiDet: A toolbox for instance-level recognition tasks. https://git.io/adelaidet, 2019.

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

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

[68] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017.

[69] Jianfeng Wang, Lin Song, Zeming Li, Hongbin Sun, Jian Sun, and Nanning Zheng. End-to-end object detection with fully convolutional network. In *CVPR*, 2021.

[70] Xinggang Wang, Jiapei Feng, Bin Hu, Qi Ding, Longjin Ran, Xiaoxin Chen, and Wenyu Liu. Weakly-supervised instance segmentation via class-agnostic learning with salient images. In *CVPR*, 2021.

[71] Xinlong Wang, Rufeng Zhang, Tao Kong, Lei Li, and Chunhua Shen. Solov2: Dynamic and fast instance segmentation. In *NIPS*, 2020.

[72] Enze Xie, Peize Sun, Xiaoge Song, Wenhai Wang, Xuebo Liu, Ding Liang, Chunhua Shen, and Ping Luo. Polarmask: Single shot instance segmentation with polar representation. In *CVPR*, 2020.

[73] Chenyang Xu and Jerry L. Prince. Gradient vector flow: A new external force for snakes. In *CVPR*, 1997.

[74] Jinrong Yang, Lin Song, Songtao Liu, Zeming Li, Xiaoping Li, Hongbin Sun, Jian Sun, and Nanning Zheng. Dbq-ssd: Dynamic ball query for efficient 3d object detection. *arXiv preprint arXiv:2207.10909*, 2022.

[75] Rui Yang, Hailong Ma, Jie Wu, Yansong Tang, Xuefeng Xiao, Min Zheng, and Xiu Li. Scalablevit: Rethinking the context-oriented generalization of vision transformer. In *European Conference on Computer Vision*, pages 480–496. Springer, 2022.

[76] Rui Yang, Lin Song, Yanwei Li, Sijie Zhao, Yixiao Ge, Xiu Li, and Ying Shan. Gpt4tools: Teaching large language model to use tools via self-instruction. *arXiv preprint arXiv:2305.18752*, 2023.

[77] Siwei Yang, Longlong Jing, Junfei Xiao, Hang Zhao, Alan L. Yuille, and Yingwei Li. Asyinst: Asymmetric affinity with depthgrad and color for box-supervised instance segmentation. *arXiv preprint arXiv:2212.03517*, 2022.

[78] Bingfeng Zhang, Jimin Xiao, Jianbo Jiao, Yunchao Wei, and Yao Zhao. Affinity attention graph neural network for weakly supervised semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(11):8082–8096, 2022.

[79] Shiwei Zhang, Lin Song, Changxin Gao, and Nong Sang. Glnet: Global local network for weakly supervised action localization. *IEEE Transactions on Multimedia*, 22(10):2610–2622, 2019.

[80] Songyang Zhang, Lin Song, Songtao Liu, Zheng Ge, Zeming Li, Xuming He, and Jian Sun. Workshop on autonomous driving at cvpr 2021: Technical report for streaming perception challenge. *arXiv preprint arXiv:2108.04230*, 2021.

[81] Tao Zhang, Shiqing Wei, and Shunping Ji. E2EC: an end-to-end contour-based method for high-quality high-speed instance segmentation. In *CVPR*, 2022.

[82] Wenwei Zhang, Jiangmiao Pang, Kai Chen, and Chen Change Loy. K-net: Towards unified image segmentation. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *NIPS*, pages 10326–10338, 2021.

[83] Zhaohui Zheng, Ping Wang, Wei Liu, Jinze Li, Rongguang Ye, and Dongwei Ren. Distance-iou loss: Faster and better learning for bounding box regression. In *AAAI*, 2020.

[84] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: deformable transformers for end-to-end object detection. In *ICLR*, 2021.