Interactive Object Segmentation with Inside-Outside Guidance

Shiyin Zhang[1,2], Jun Hao Liew[3], Yunchao Wei[4], Shikui Wei[1,2*], Yao Zhao[1,2]
[1]Institute of Information Science, Beijing Jiaotong University
[2]Beijing Key Laboratory of Advanced Information Science and Network Technology
[3]National University of Singapore [4]ReLER, University of Technology Sydney
https://github.com/shiyinzhang/Inside-Outside-Guidance

Abstract

This paper explores how to harvest precise object segmentation masks while minimizing the human interaction cost. To achieve this, we propose an Inside-Outside Guidance (IOG) approach in this work. Concretely, we leverage an inside point that is clicked near the object center and two outside points at the symmetrical corner locations (top-left and bottom-right or top-right and bottom-left) of a tight bounding box that encloses the target object. This results in a total of one foreground click and four background clicks for segmentation. The advantages of our IOG are four-fold: 1) the two outside points can help to remove distractions from other objects or background; 2) the inside point can help to eliminate the unrelated regions inside the bounding box; 3) the inside and outside points are easily identified, reducing the confusion raised by the state-of-the-art DEXTR in labeling some extreme samples; 4) our approach naturally supports additional clicks annotations for further correction. Despite its simplicity, our IOG not only achieves state-of-the-art performance on several popular benchmarks, but also demonstrates strong generalization capability across different domains such as street scenes, aerial imagery and medical images, without fine-tuning. In addition, we also propose a simple two-stage solution that enables our IOG to produce high quality instance segmentation masks from existing datasets with off-the-shelf bounding boxes such as ImageNet and Open Images, demonstrating the superiority of our IOG as an annotation tool.

1. Introduction

Over the past few years, we have witnessed a revolutionary advancement in semantic [44, 40, 68, 69, 10, 11, 12, 30, 15, 31] and instance segmentation [25, 36, 8, 60, 13, 65, 34, 4, 9, 43, 29] for different domains, such as general scenes [20, 41, 70], autonomous driving [17, 48, 21], aerial imagery [57, 16], medical diagnosis [22, 56], *etc.* Successful segmentation models are usually built on the shoulders of large volumes of high-quality training data. However, the process to create the pixel-level training data necessary to build these models is often expensive, laborious and time-consuming. Thus, interactive segmentation, which allows the human annotators to quickly extract the object-of-interest by providing some user inputs such as bounding boxes [66, 52, 64] or clicks [67, 38, 45, 37], appears to be an attractive and efficient way to reduce the annotation effort.

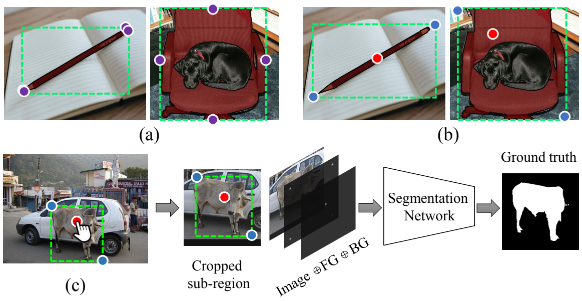

Figure 1. (a) User inputs of DEXTR [46]. (b) User inputs of the proposed IOG method. (c) An overview of our IOG framework.

Recently, Maninis *et al.* [46] explored the use to extreme points of an object (*left-most, right-most, top, bottom pixels*) for interactive image segmentation. Despite its simplicity, the extreme points have demonstrated fast interactive annotation speed and high segmentation quality across different application domains. Nevertheless, we argue that the clicking paradigm of extreme points also brings some issues: 1) annotating extreme points requires users to carefully click at the object boundaries, which usually consumes much more time as compared to the common clicking setting where users can click at *any* of the interior and exterior of object regions; 2) the annotation process can sometimes be confusing when multiple extreme points appear at similar spatial locations (pencil in Figure 1(a)) or when there are unrelated objects or background lying inside the target object (dog in Figure 1(a)).

To tackle the aforementioned issues as well as to pro-

[*]Corresponding author

mote the effectiveness and efficiency of the interactive process, we propose an approach named Inside-Outside Guidance (IOG), which requires only **three** points (an inside point and two outside points) to indicate the target object. Specifically, the inside point usually locates around the center of the object instance while the two outside ones can be clicked at any symmetrical corner locations of a tight bounding box enclosing the target instance (either the *top-left and bottom-right* or *top-right and bottom-left* pixels). Figure 1(b) shows two examples of our proposed labeling scheme. Similar to [46], our IOG relaxes the generated bounding box by several pixels before cropping from the input image to include context. This results in a total of one foreground and four background clicks (two clicked outside points and two additional inferred ones based on the bounding box), which are then encoded as foreground/ background localization heatmaps and concatenated with the cropped image for training the segmentation network. The overview of our IOG is shown in Figure 1(c).

Our IOG strategy not only improves the annotation speed by reducing the confusion raised by [46], but also naturally supports annotation of additional points at the erroneous regions for further refinement. We perform extensive experiments on PASCAL [20], GrabCut [52] and COCO [41] to demonstrate the effectiveness of our IOG as an annotation tool. In particular, given only three points, our IOG achieves 93.2% mIoU score on PASCAL, which is the new state-of-the-art. Our IOG further improves to 94.4% when the 4th click is added for interactive correction.

In addition, we also show that our model generalizes well in cross-domain annotation, where our PASCAL- or COCO-trained model produces high quality segmentation masks when annotating street scenes [17], aerial imagery [57, 16] and medical images [22] **without** fine-tuning. Beyond this, we also propose a simple two-stage solution that enables our IOG to harvest precise instance segmentation masks from the off-the-shelf datasets with bounding box annotations such as ImageNet [53] and Open Images [35] without any human interaction. Finally, we release `Pixel-ImageNet`[1], a dataset with 0.615M instance masks of ImageNet [53] collected using our IOG. We hope this work can significantly benefit the future researchers in collecting large-scale pixel-level annotations.

2. Related Work

Interactive Segmentation: Most interactive segmentation methods target at relieving human effort when labeling pixel-level annotations with the help of bounding box [66, 64, 52], clicks [67, 38, 37, 45] or contours [7, 1, 42] as guidance. GrabCut [52] employs bounding boxes to guide the segmentation process, which is one of the pioneering works for interactive segmentation task. Similarly, Xu *et al.* [66] also take bounding boxes as inputs to train a deep convolutional neural network (CNN) for interactive segmentation. On the other hand, iFCN [67] is proposed to conduct interactive segmentation by guiding a CNN with positive (foreground) and negative (background) points clicked by the users. RIS-Net [38] improves the iFCN by augmenting a local context branch. Recently, Maninis *et al.* [46] propose DEXTR that leverages only 4 extreme points for segmentation and achieve the new state-of-the-art. Comparing with DEXTR, we further advance the interactive annotation process by reducing the number of clicks required from 4 to 3. Our IOG not only tackles the issues raised by DEXTR, but also achieves much better segmentation performance. Beyond the above mentioned approaches, some other works [7, 1, 42] alternatively propose to conduct interactive segmentation by directly predicting a polygon or spline around the target object.

Weakly Supervised Segmentation: Among many alternatives in addressing the expensive pixel-level annotations, weakly supervised learning has been extensively studied in the literature. Particularly, image-level labels [50, 61, 63, 62, 27, 32], points [3, 51], bounding boxes [18, 33] and scribbles [39, 58, 59] have been employed as guidance to supervise the training of semantic segmentation networks. Different from these methods, our proposed IOG still relies on fully annotated masks as supervision and utilizes three additional points as the guidance to produce the segmentation mask of the target object.

Semantic Segmentation: Fully convolutional networks (FCN) [44, 40, 68, 69, 10, 11, 12, 30, 15] has greatly advanced the semantic image segmentation. The success of CNN-based interactive segmentation has benefited significantly from the development of FCNs. Specifically, FCN [44], DeepLab series [10, 11, 12] and PSP [68] have been directly applied to tackle the interactive segmentation [67, 38, 45, 46]. In this work, we investigate which type of network is more suitable for conducting interactive segmentation tasks and choose to adopt a coarse-to-fine network structure [14] as the backbone of our IOG method. We experimentally validate our choice can further boost the accuracy of interactive segmentation by a large margin.

3. Method

3.1. Inside-Outside Guidance

Our Inside-Outside Guidance clicking paradigm consists of **three** points: an interior click (inside point) located roughly at the object center and two exterior clicks (outside points) at any symmetrical corner locations (either *top-left* and *bottom-right* or *top-right* and *bottom-left*) that form an almost-tight bounding box enclosing the target-of-interest

[1] https://github.com/shiyinzhang/Pixel-ImageNet

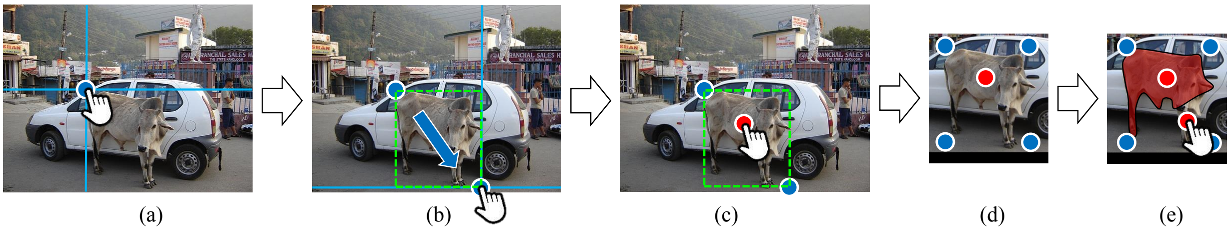

Figure 2. **Inside-Outside guidance.** (a) The vertical and horizontal guide lines are used to assist the user in clicking on the corner of an imaginary box enclosing the object. (b) A box is generated on-the-fly when the user moves the cursor. (c) An interior click is placed around the object center. (d) The box is relaxed by several pixels before cropping to include context. The interior click (red) with four exterior clicks (two clicked corners and two automatically inferred ones) (blue) constitute our Inside-Outside guidance that encode the foreground and background regions, respectively. (e) Our method naturally supports additional clicks annotation for further refinement.

(Figure 2). In this way, the two exterior clicks, together with two additional inferred ones based on the generated bounding box, provide an "outside" guidance (indicating the background regions) while the interior click gives an "inside" guidance (indicating the foreground regions), thus giving the name *Inside-Outside Guidance (IOG)*.

Outside guidance: The outside guidance is formulated by the corners of the bounding box enclosing the object. However, it was previously reported that drawing a tight box can be time consuming ([55] reported 25.5s for drawing one box on ImageNet [53][2]). This is due to the difficulty of clicking on the corners of an imaginary box where these corners are often not on the object [49]. Thus, several adjustments are usually required to ensure the resulting box is tight. However, with some simple modifications to the annotation interface, such as using a horizontal and a vertical guide line to make the box visible when clicking on a corner, the burden of drawing a bounding box can be largely relieved as shown in Figure 2(a)-(b). Moreover, in our case, we do not necessarily need a tight bounding box where an almost-tight box usually suffices. In our user study, we observe that drawing a bounding box typically take about 6.7s with the help of the guide lines.

Inside guidance: The inside guidance is formulated as an interior click located around the object center for disambiguating the segmentation target since there could be multiple objects within the same box. To simulate clicks annotated by human annotators, we propose to sample the inside point at the location that is furthest away from the object boundaries. In particular, let O denotes the pixels belonging to the object, we first compute a distance map D based on Euclidean distance transformation as follows:

$$D_i = \min_{\forall j \in O} \text{dist}(i, j) \quad (1)$$

where D_i refers to the value of D at pixel location i while dist(i, j) denotes the Euclidean distance between pixel locations i and j. Then, the interior click is sampled at the location $k = \arg\max_{\forall i \in O} D_i$. The validity of such sampling scheme is verified in Section 4.5 by comparing with the actual interior clicks collected from real users. Note that annotating the inside point is very fast, taking about 1.5s in our user study.

Clicks representations: We use the same clicks representation as DEXTR[46] by centering a 2D Gaussian around each click, creating two separate heatmaps for foreground and background clicks. The resulting heatmaps are concatenated with the RGB input image to form a 5-channel input for the network. Similar to [46], the bounding box is first relaxed by several pixels to include context, followed by cropping to focus on the object-of-interest (Figure 2(d)).

Compared with existing clicks-based [67, 46] and box-based [66] interactive segmentation approaches, our proposed IOG has the best of both worlds: (i) **flexibility**: since the annotated three points are encoded as foreground and background clicks, our IOG naturally supports additional clicks annotations for further correction (Figure 2(e)); (ii) **more information**: our approach encodes more prior information about the object, including the location of hard background and the rough size of the target.

3.2. Segmentation Network

Here, we discuss the architectural design of our segmentation network. We employ a ResNet-50 [26]-based DeepLabv3+ [12] as our starting point and we already observe decent segmentation performance (90.0% IoU on PASCAL), demonstrating the effectiveness of our proposed IOG. Nevertheless, closer inspection on the segmentation quality reveals that segmentation errors mostly occur along the object boundaries as shown in Figure 3. Simply replacing the backbone with a deeper network such as ResNet-101 only brings marginal improvement (Vanilla IOG in Figure 6 right), suggesting some architectural modifications have to be made to ensure the network focuses on refining the inaccurate segmentation along the object boundaries.

[2] Some papers reported much faster timings (*e.g.* [54] reported 10.21s while [19] reported 7.0s). However, [49] argue that the annotated boxes are of low quality (not tight around the object).

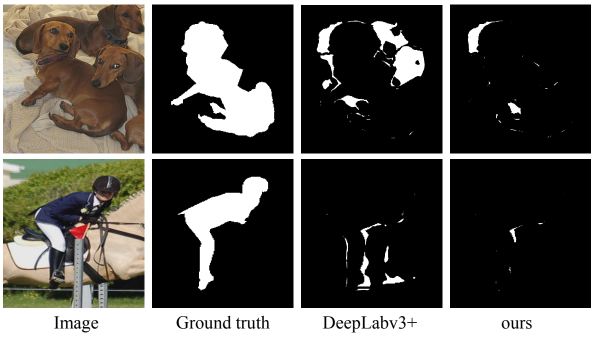

Figure 3. **Qualitative comparison in terms of segmentation errors.** Note that the segmentation errors mostly occur along the object boundaries when using DeepLabv3+ [12] as backbone whereas our coarse-to-fine structure produces precise boundaries.

In this work, we propose to adopt a coarse-to-fine design for addressing the aforementioned issue (Figure 4). In particular, we employ a cascaded structure similar to [14] which was originally proposed for human pose estimation task. Specifically, the segmentation network consists of two subnetworks. The first subnetwork, CoarseNet applies an FPN-like design [40] that progressively fuses the semantic information from the deeper layers with low-level details from the earlier layers via lateral connections. Different from [14], we also append a pyramid scene parsing (PSP) module [68] at the deepest layer for enriching the representation with global contextual information. Given a coarse prediction from the CoarseNet, the second subnetwork, FineNet aims at recovering the missing boundary details. This is achieved with a multi-scale fusion structure that fuses the information across different levels in the CoarseNet via upsampling and concatenation operations. Similar to [14], we also apply more convolution blocks for features at deeper layers for better trade-off between the accuracy and efficiency. We refer the readers to the supplementary materials for more details. Note that we do **not** claim any novelty in the network design. Instead, our contribution lies in the finding that a coarse-to-fine structure is necessary for obtaining more precise segmentation masks whereas stacking more layers does not. We believe other coarse-to-fine structure might also work and we leave it as our future works.

Training and testing: Our segmentation network is trained end-to-end using binary cross-entropy loss. In addition, we also apply side losses at each level of CoarseNet as a form of deep supervision [14]. During inference, the segmentation mask is obtained by simply thresholding the final network prediction. Since our approach does not involve any post-processing, it is extremely fast, where a single forward pass on a ResNet-101 backbone typically requires only 20 ms on a Nvidia GeForce GTX 1080 GPU. It is thus well-suited for practical interactive image segmentation application.

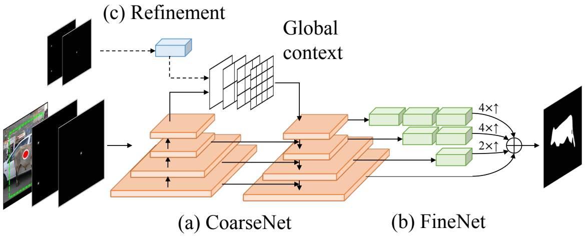

Figure 4. **Network Architecture.** (a)-(b) Our segmentation network adopts a coarse-to-fine structure similar to [14], augmented with a pyramid scene parsing (PSP) module [68] for aggregating global contextual information. (c) We also append a lightweight branch before the PSP module to accept the additional clicks input for interactive refinement.

3.3. Beyond Three Clicks...

Although our IOG approach requires only three clicks to perform segmentation, our framework naturally supports *interactive* adding of new foreground and background clicks for further refinement if the user is not satisfied with the current segmentation output. To achieve this, we append a lightweight branch before the PSP module to accept the two-channel Gaussian heatmaps encoding all the foreground and background clicks (Figure 4(c)). We empirically found that this setting not only works better than modifying the inputs at the beginning of the segmentation network, but also runs much faster since the encoder features only needs to be computed once.

During training, we adopt an iterative training strategy to simulate the interactive process where an additional click is introduced to the erroneous regions by the user for correction. More specifically, an initial segmentation mask is first obtained given only three clicks. A new click is then added to the center of the largest erroneous region and second forward pass is conducted. Results presented in Section 4.3 shows that such iterative training strategy is necessary.

4. Experiments

We conduct extensive experiments on **11** publicly available benchmarks, including PASCAL [20], GrabCut [52], COCO [41], ImageNet [53], Open Images [53], Cityscapes [17], Rooftop [57], Agriculture-Vision [16], ssTEM [22], Pascal-Context [47], and COCO-Stuff [6], to demonstrate the effectiveness and the generalization capabilities of our IOG. We choose ResNet-50 and ResNet-101 as the two backbones of the IOG for fair comparison with previous approaches. Following the common practice [46], we employ PASCAL as the main benchmark to verify the importance of each component proposed in our IOG.

4.1. Implementation Details

Simulated inside-outside points: We use the ground truth masks to generate the inside-outside points for training. For

Methods	Number of Clicks		IoU(%) @ 4 clicks	
	PASCAL@85%	GrabCut@90%	PASCAL	GrabCut
Graph cut [5]	> 20	> 20	41.1	59.3
Random walker [23]	16.1	15	55.1	56.9
Geodesic matting [2]	> 20	> 20	45.9	55.6
iFCN [66]	8.7	7.5	75.2	84.0
RIS-Net [38]	5.7	6	80.7	85.0
DEXTR [46]	4	4	91.5	94.4
Li et al. [37]	-	4.79	-	-
ITIS [45]	3.4	5.7	-	-
FCTSFN [28]	4.58	3.76	-	-
IOG-ResNet101 (**ours**)	3	3	93.2*	96.3*
IOG-ResNet101 (**ours**)	4	4	94.4	96.9

Table 1. Comparison with the state-of-the-art methods on PASCAL and GrabCut in terms of the number of clicks to reach a certain IoU and in terms of quality at 4 clicks. *denotes the IoU of our IOG given only 3 clicks.

	Train	Test	DEXTR [46]	ours
Unseen Classes	PASCAL	COCO MVal (seen)	79.9%	81.7%
	PASCAL	COCO MVal (unseen)	80.3%	82.1%
Generalization	PASCAL	COCO MVal	80.1%	81.9%
	COCO	COCO MVal	82.1%	85.2%
	COCO	PASCAL	87.8%	91.6%
	PASCAL	PASCAL	89.8%	93.2%

Table 2. Comparison in terms of generalization ability between the state-of-the-art DEXTR and our IOG.

Backbone	Context	FineNet	Side losses	Dataset	IoU
ResNet-50	✗	✓	✓	PASCAL-1k	91.2
ResNet-50	✓	✗	✓	PASCAL-1k	90.8
ResNet-50	✓	✓	✗	PASCAL-1k	90.6
ResNet-50	✓	✓	✓	PASCAL-1k	91.6
ResNet-50	✓	✓	✓	PASCAL-10k	92.8
ResNet-101	✓	✓	✓	PASCAL-1k	92.0
ResNet-101	✓	✓	✓	PASCAL-10k	93.2

Table 3. **Ablation Study.** Justification of each component in the segmentation network on the PASCAL VOC 2012 *val* set.

outside points, we take the corners of the bounding-box extracted from ground truth masks and relax by 10 pixels to simulate a loosen box provided by real users. For inside point, we sample a click that is furthest from the object boundaries and apply random perturbation. The effects of perturbation will be studied in Section 4.5.

Training and testing details: IOG is trained on PASCAL 2012 Segmentation [3] for a maximum of 100 epochs or on MS COCO 2014 for a maximum of 10 epochs. We acquire the results from the best performing epoch. For PASCAL, the batch size is set to 5 whereas for COCO, we train on 2 GPUs with an effective batch size of 10. For COCO, we also construct a set of "void" pixels around the boundaries of the ground truth masks and ignore them during training. The learning rate, momentum and weight decay are set to 10^{-8}, 0.9 and 5×10^{-4}, respectively.

4.2. Comparison with the State-of-the-Arts

We first compare our IOG with the state-of-the-art approaches on two popular benchmarks, *i.e.*, PASCAL VOC *val* set and GrabCut. Table 1 summarizes the number of clicks needed for each method to reach a certain performance, and the corresponding IoU scores when only 4 clicks are provided. It can be observed that our IOG outperforms all others by more than 1.7% and 1.9% on PASCAL and GrabCut, respectively. When allowing iterative refinement (*i.e.* from 3 to 4 clicks), the performance can be further enhanced to 94.4% and 96.9%, which well demonstrate the effectiveness of our IOG in handling the additional user inputs for further correction.

Next, we compare the generalization ability between the state-of-the-art DEXTR and our IOG on unseen classes and across different datasets. Following the setting in [46], we compare the performances on two benchmarks, *i.e.* PASCAL and COCO mini-val (MVal). For the *Unseen Classes* setting, we leverage the model trained on PASCAL and evaluate its IoU on COCO MVal *seen* (*i.e.* images with the same categories as PASCAL) and COCO MVal *unseen* (*i.e.* images with different categories as PASCAL). For the *Generalization* setting, we train the model on PASCAL (or COCO) and evaluate the performance on COCO MVal (or PASCAL), regardless of the testing categories. As shown in Table 2, our IOG makes consistent improvements over DEXTR on various settings despite using only 3 clicks. Some qualitative results are shown in Figure 5.

4.3. Ablation Study

Justification of each component of IOG: We perform ablation experiments on PASCAL VOC *val* set to validate the effectiveness of each component in our segmentation network. Particularly, we quantitatively justify various design choices, including the different backbones (ResNet-50 *vs.* ResNet-101), different number of training images (PASCAL-1K *vs.* PASCAL-10K), inclusion of PSP module for global contextual information (Context), FineNet and the use of side losses for training. As shown in Table 3, "Context", "FineNet" and "Side losses" can respectively lead to performance boost of 0.4%, 0.8% and 1.0% under the setting of ResNet-50 and PASCAL-1K. When augmenting additional labels from SBD (PASCAL-10k), the performance can be further improved from 91.6% to 92.8%. Finally, we obtain the state-of-the-art performance when replacing the backbone with ResNet-101 (93.2%).

Iterative training for interactive refinement: In the previous section, we have demonstrated the effectiveness of our IOG under the default setting when only 3 clicks are provided. Next, we examine the case when the user is not satisfied with the result and wants to annotate additional clicks for further correction. Specifically, we progressively

[3] We denote the PASCAL train set augmented with additional labels from SBD [24] and the one without SBD labels as PASCAL-10k (10,582 images) and PASCAL-1k (1,464 images), respectively.

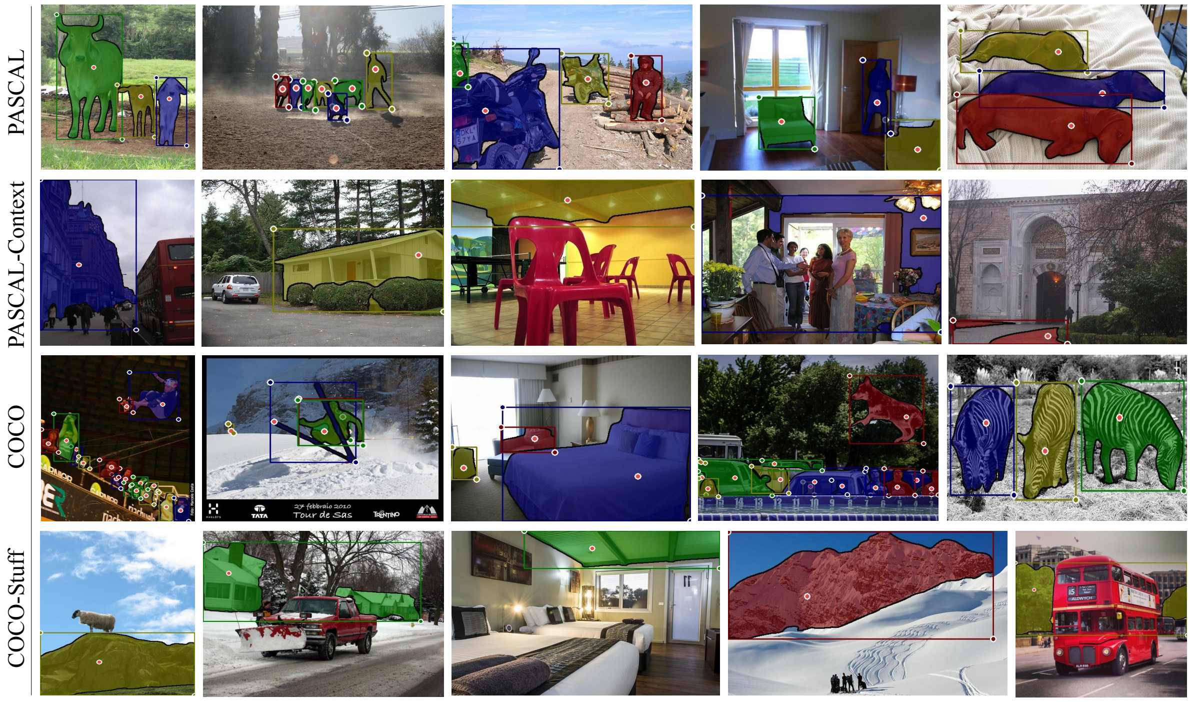

Figure 5. **Qualitative results on PASCAL [20], PASCAL-Context [47], COCO [41] and COCO-Stuff [6].** Each instance with the simulated inside-outside points and the corresponding segmentation masks are overlayed on the input image.

add a new click to the center of the largest erroneous regions similar to [67, 45]. The results are summarized in Figure 6 (left). We can observe that: 1) additional clicks do not bring significant performance gains without iterative training, demonstrating the importance of iterative training for interactive refinement; 2) adding the clicks to the intermediate layers of the segmentation network (Section 3.3) is more effective than modifying the inputs at the beginning of the network. An interesting observation is that adding clicks to the beginning of the model without iterative training will lead to performance degradation. One possible reason is that the inside points always locate around the object center whereas the newly added correction clicks are usually distributed near the object boundaries, which confuses the trained model and harms the performance. Some qualitative examples of interactive refinement can be found in Figure 7.

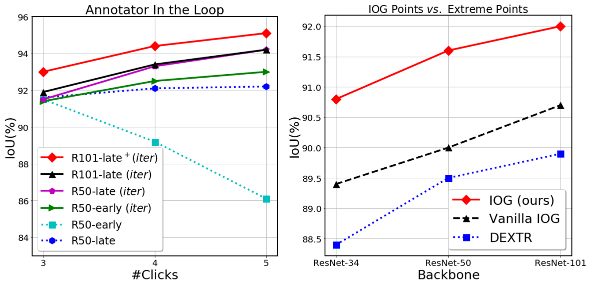

Figure 6. (left) The effect of iterative training for interactive refinement. “early” and “late” denote adding clicks input to the beginning or intermediate layer of the network, respectively. “iter” implies iterative training (Section 3.3) while “+” denotes training on larger dataset (PASCAL-10k). (right) Comparison between IOG points and extreme points.

IOG points vs. extreme points: We study the performance of our proposed IOG points when compared with the extreme points used in DEXTR. For fair comparison, we use the released code[4] and re-train DEXTR using DeepLabv3+ [12] as the fully convolutional architecture on PASCAL-1K. All the models are pre-trained only on ImageNet [53]. We conduct experiments using three different backbones, *i.e.* ResNet-34, ResNet-50 and ResNet-101, to validate the robustness of the proposed method. As shown in Figure 6 (right), our proposed IOG points consistently outperform the extreme points given the same network architecture (Vanilla IOG *vs.* DEXTR). When using a coarse-to-fine network structure (Section 3.2), we can see that our IOG significantly outperforms the baselines by a large margin. Interestingly, our IOG with ResNet-34 as backbone already surpasses the state-of-the-art DEXTR using ResNet-101, demonstrating the effectiveness of the proposed IOG over the extreme points.

[4] https://github.com/scaelles/DEXTR-PyTorch

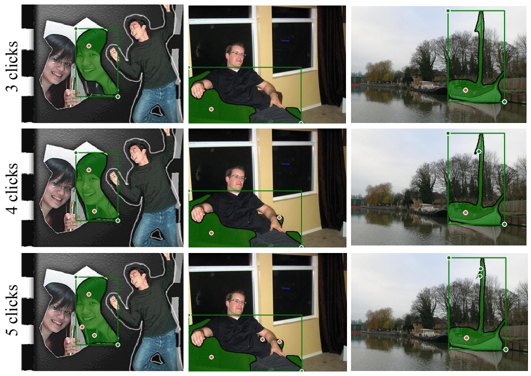

Figure 7. **Interactive refinement.** Our proposed IOG supports interactive adding of new clicks for further refinement.

4.4. Cross-Domain Evaluation

In Section 4.2, we have demonstrated the generalization capability of our IOG on unseen classes and across different datasets (train on PASCAL and test on COCO and *vice versa*). However, images in both the PASCAL and COCO datasets are of general scenes while a powerful annotation tool should generalize well even on different imagery types. In the following section, we examine the generalization ability of our model across different domains, including non-PASCAL-style object categories and stuff categories.

Object categories: Following [42, 1], we evaluated the performance of our IOG on 3 imagery types, including street scenes (Cityscapes [17]), aerial imagery (Rooftop [57]) and medical images (ssTEM [22]). The results are summarized in Table 4, 5 and 6. We can see that our model outperforms the baselines by a large margin on Rooftop and ssTEM datasets even **without** fine-tuning. More interestingly, as shown in Table 4, our PASCAL-trained model already performs on-par with the Cityscapes-trained methods when evaluating on the Cityscapes dataset. This suggests that our IOG generalizes well even across different domains. Moreover, the model performance can be further improved by fine-tuning using only 10% of the new dataset, where our model significantly outperforms all the other baselines. In addition, we also applied our IOG on the more challenging Agriculture-Vision dataset [16] and still achieved satisfactory performance when using small amounts of data for fine-tuning. Fig 8 provides some qualitative examples.

Stuff categories: In Fig 5, we show some qualitative results of our IOG fine-tuned on PASCAL-Context [47] and COCO-Stuff [6] to verify the performance of our IOG when segmenting "stuff" categories. The results show that our IOG generalizes well to background classes too.

Methods	Train	Finetune	Backbone	#Clicks	IoU
Curve-GCN [42]	Cityscapes	N.A.	ResNet-50	2	76.3
Curve-GCN [42]	Cityscapes	N.A.	ResNet-50	2.4	77.6
Curve-GCN [42]	Cityscapes	N.A.	ResNet-50	3.6	80.2
DEXTR [42]	Cityscapes	N.A.	ResNet-101	4	79.4
IOG (ours)	PASCAL	✗	ResNet-50	3	77.9
IOG (ours)	PASCAL	✓	ResNet-50	3	82.2
IOG (ours)	PASCAL	✓	ResNet-101	3	82.7
IOG (ours)	COCO	✓	ResNet-101	3	**83.8**

Table 4. **Cross domain analysis on Cityscapes [17].** "Finetune" indicates that the method is fine-tuned on a small set of the Cityscapes dataset (10%).

Methods	Train	Finetune	Backbone	#Clicks	IoU
Curve-GCN [42]	CityScapes	✗	ResNet-50	2	68.3
Curve-GCN [42]	CityScapes	✓	ResNet-50	2	78.2
IOG (ours)	PASCAL	✗	ResNet-50	3	90.7
IOG (ours)	PASCAL	✓	ResNet-50	3	92.8
IOG (ours)	PASCAL	✓	ResNet-101	3	93.6
IOG (ours)	COCO	✓	ResNet-101	3	**94**

Table 5. **Cross domain analysis on Rooftop [57].** Even without fine-tuning, our method already outperforms Curve-GCN with fine-tuning, showing the strong generalization of our approach.

Methods	Train	Finetune	Backbone	#Clicks	IoU
Curve-GCN [42]	CityScapes	✗	ResNet-50	2	60.9
IOG (ours)	PASCAL	✗	ResNet-50	3	81.4
IOG (ours)	PASCAL	✗	ResNet-101	3	**83.7**

Table 6. **Cross domain analysis on ssTEM [22].** Note that ssTEM does not have a training split, therefore we do not perform fine-tuning on this dataset.

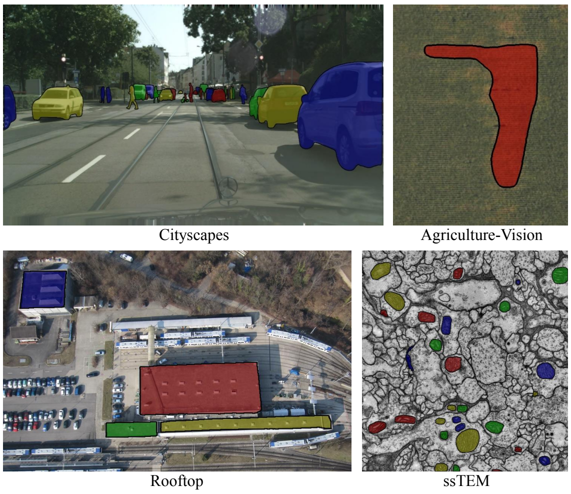
Cityscapes
Agriculture-Vision
Rooftop
ssTEM

Figure 8. **Cross-domain performance.** Qualitative results of our IOG on Cityscapes, Agricultural-Vision, Rooftop, and ssTEM.

4.5. More Discussions

Robustness to user variance when choosing the inside points: In the previous experiments, we examine the effectiveness of our IOG using the simulated inside-outside points as inputs. Nevertheless, in practice, it is often difficult for the users to reach consensus when choosing the

Method	$r = 0$	$r = 10$	$r = 30$	$r = 50$
Simulated inside points	93.2	92.9	92.8	92.0
Manual inside points	$90.8_{\pm 1.0}$	$91.6_{\pm 0.7}$	$92.3_{\pm 0.2}$	$92.0_{\pm 0.1}$

Table 7. **Manual vs. simulated inside points.** r denotes the radius of perturbation applied during training. All the models take ResNet-101 as backbone and are trained on PASCAL-10k.

Method	Backbone	Train	IoU
(A) Crop	ResNet-50	PASCAL-1k	87.5
(B) Geo	ResNet-50	PASCAL-1k	89.5
(C) Sim	ResNet-50	PASCAL-1k	86.1
(D) Outside only	ResNet-50	PASCAL-1k	89.5
(D) Outside only	ResNet-101	PASCAL-10k	90.9
(E) 2-stage	ResNet-101	PASCAL-10k	91.1

Table 8. **Extension to dataset with box annotations only.** All the results are reported on PASCAL *val* using box annotations only.

inside points although the users usually make consistent choices in annotating the outside points. The inconsistent inputs between training and testing will often have a negative impact on the segmentation performance, especially when applied to real annotation scenario.

To alleviate the negative effect caused by user variance in selecting the inside points, we randomly perturb the position of the inside points during training. In particular, we first identify a circular region centered at the inside point extracted from the ground truth mask with a pre-defined radius (r). Then, we randomly sample a click from this region to serve as the inside point for training. To validate the effectiveness of the proposed modification, we collected the inside points annotations on all instances in PASCAL *val* set from 5 different users. As shown in Table 7, we first notice a large performance degradation when testing the perturbation-free model with the human-provided inputs (from 93.2 to 90.8). However, the performance gaps gradually reduce when larger perturbation is applied during training. The model reaches the best trade-off when r is 30.

Extension to datasets with box annotations only: Many existing off-the-shelf datasets such as ImageNet and Open Images, have provided bounding box annotations. Here, we explore how to quickly harvest high-quality instance segmentation masks using our IOG when only bounding box annotations are available. Specifically, we consider the annotated bounding box as an incomplete annotation for our IOG where the inside point is absent. To this end, we propose a simple two-stage solution using a small network to predict a coarse mask based on the bounding box, where the mask is used to infer the inside point candidates for IOG later. We compare this against the following baselines and the results are summarized in Table 8.

(A) **Crop:** We train a network that takes the cropped RGB image as input and predicts the segmentation.

(B) **Geo:** We train a network that takes the geometric center of the box as inside point for segmentation.

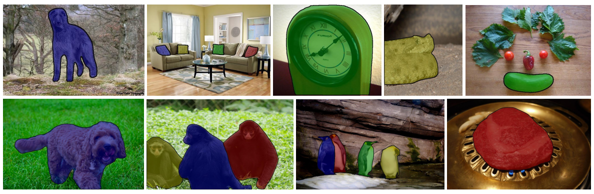

Figure 9. **Qualitative results on ImageNet (top) and Open Image (bottom) using our proposed 2-stage approach.** Note that only bounding box annotations are provided.

(C) **Sim:** We train our IOG with simulated clicks (Section 3.1) but using the geometric center of the given box as inside point during test time.

(D) **Outside only:** We train a single network that takes the outside points only to perform segmentation.

(E) **2-stage:** We extract the inside point from the segmentation masks produced by (D) and pass to our IOG for the final prediction.

We first observe that the setting (C) performs poorly due to train-test inconsistency. On the other hand, the methods (B) and (D) have similar performance. This is because the geometric center of the box always locates the same location after cropping, thus the network learns to ignore this input. By adopting stronger backbone and more training images, the performance of (D) can be further improved. Finally, taking the inside point from the segmentation masks predicted by (D) as inputs for our IOG produces the best result. Some qualitative results on ImageNet and Open Images are shown in Figure 9. With the annotated bounding boxes (∼0.615M) of ILSVRC-LOC, we apply our IOG to collect their pixel-level annotations, named `Pixel-ImageNet`, which are publicly available at https://github.com/shiyinzhang/Pixel-ImageNet.

5. Conclusion

We propose a simple yet effective Inside-Outside Guidance (IOG) approach for minimizing the labeling cost. The proposed IOG requires only three points from the users, *i.e.* an inside point near the object center and two outside points that form a box enclosing the target object. In addition, our method naturally supports interactive annotation of additional points for further correction. Despite its simplicity, extensive experiments show that our model generalizes well across different datasets and domains, demonstrating its superiority as an annotation tool.

Acknowledgments This work is in part supported by National Key Research and Development of China (2017YFC1703503), National Natural Science Foundation of China (No.61532005, 61972022), Grant U1936212, and ARC DECRA DE190101315.

References

[1] David Acuna, Huan Ling, Amlan Kar, and Sanja Fidler. Efficient interactive annotation of segmentation datasets with polygon-rnn++. In *CVPR*, pages 859–868, 2018.

[2] Xue Bai and Guillermo Sapiro. A geodesic framework for fast interactive image and video segmentation and matting. In *ICCV*, 2007.

[3] Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. What's the point: Semantic segmentation with point supervision. In *ECCV*, pages 549–565, 2016.

[4] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. YOLACT: real-time instance segmentation. In *ICCV*, pages 9157–9166, 2019.

[5] Yuri Y Boykov and Marie-Pierre Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In *ICCV*, 2001.

[6] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Cocostuff: Thing and stuff classes in context. pages 1209–1218, 2018.

[7] Lluis Castrejon, Kaustav Kundu, Raquel Urtasun, and Sanja Fidler. Annotating object instances with a polygon-rnn. In *CVPR*, pages 5230–5238, 2017.

[8] Hao Chen, Kunyang Sun, Zhi Tian, Chunhua Shen, Yongming Huang, and Youliang Yan. BlendMask: Top-down meets bottom-up for instance segmentation. In *CVPR*, 2020.

[9] 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.

[10] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *TPAMI*, 40(4):834–848, 2017.

[11] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[12] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018.

[13] Xinlei Chen, Ross Girshick, Kaiming He, and Piotr Dollár. Tensormask: A foundation for dense object segmentation. In *ICCV*, pages 2061–2069, 2019.

[14] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *CVPR*, 2018.

[15] Bowen Cheng, Liang-Chieh Chen, Yunchao Wei, Yukun Zhu, Zilong Huang, Jinjun Xiong, Thomas S Huang, Wen-Mei Hwu, and Honghui Shi. Spgnet: Semantic prediction guidance for scene parsing. In *ICCV*, pages 5218–5228, 2019.

[16] Mang Tik Chiu, Xingqian Xu, Yunchao Wei, Zilong Huang, Alexander Schwing, Robert Brunner, Hrant Khachatrian, Hovnatan Karapetyan, Ivan Dozier, Greg Rose, et al. Agriculture-vision: A large aerial image database for agricultural pattern analysis. In *CVPR*, 2020.

[17] 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.

[18] Jifeng Dai, Kaiming He, and Jian Sun. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *ICCV*, pages 1635–1643, 2015.

[19] Suyog Dutt Jain and Kristen Grauman. Predicting sufficient annotation strength for interactive foreground segmentation. In *ICCV*, pages 1313–1320, 2013.

[20] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (VOC) challenge. *IJCV*, 2010.

[21] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, pages 3354–3361, 2012.

[22] Stephan Gerhard, Jan Funke, Julien Martel, Albert Cardona, and Richard Fetter. Segmented anisotropic sstem dataset of neural tissue. *figshare*, pages 0–0, 2013.

[23] Leo Grady. Random walks for image segmentation. *TPAMI*, 2006.

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

[25] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017.

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

[27] Qibin Hou, PengTao Jiang, Yunchao Wei, and Ming-Ming Cheng. Self-erasing network for integral object attention. In *NeurIPS*, pages 549–559, 2018.

[28] Yang Hu, Andrea Soltoggio, Russell Lock, and Steve Carter. A fully convolutional two-stream fusion network for interactive image segmentation. *Neural Networks*, 109:31–42, 2019.

[29] Zhaojin Huang, Lichao Huang, Yongchao Gong, Chang Huang, and Xinggang Wang. Mask scoring r-cnn. In *CVPR*, pages 6409–6418, 2019.

[30] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *ICCV*, pages 603–612, 2019.

[31] Zilong Huang, Yunchao Wei, Xinggang Wang, Honghui Shi, Wenyu Liu, and Thomas S Huang. Alignseg: Feature-aligned segmentation networks. *arXiv preprint arXiv:2003.00872*, 2020.

[32] Peng-Tao Jiang, Qibin Hou, Yang Cao, Ming-Ming Cheng, Yunchao Wei, and Hong-Kai Xiong. Integral object mining via online attention accumulation. In *ICCV*, pages 2070–2079, 2019.

[33] Anna Khoreva, Rodrigo Benenson, Jan Hosang, Matthias Hein, and Bernt Schiele. Simple does it: Weakly supervised instance and semantic segmentation. In *CVPR*, pages 876–885, 2017.

[34] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. Pointrend: Image segmentation as rendering. *arXiv preprint arXiv:1912.08193*, 2019.

[35] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv preprint arXiv:1811.00982*, 2018.

[36] Yi Li, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei. Fully convolutional instance-aware semantic segmentation. In *CVPR*, pages 2359–2367, 2017.

[37] Zhuwen Li, Qifeng Chen, and Vladlen Koltun. Interactive image segmentation with latent diversity. In *CVPR*, pages 577–585, 2018.

[38] Jun Hao Liew, Yunchao Wei, Wei Xiong, Sim-Heng Ong, and Jiashi Feng. Regional interactive image segmentation networks. In *ICCV*, pages 2746–2754. IEEE, 2017.

[39] Di Lin, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *CVPR*, pages 3159–3167, 2016.

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

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

[42] Huan Ling, Jun Gao, Amlan Kar, Wenzheng Chen, and Sanja Fidler. Fast interactive object annotation with curve-gcn. In *CVPR*, pages 5257–5266, 2019.

[43] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *CVPR*, pages 8759–8768, 2018.

[44] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015.

[45] Sabarinath Mahadevan, Paul Voigtlaender, and Bastian Leibe. Iteratively trained interactive segmentation. *arXiv preprint arXiv:1805.04398*, 2018.

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

[47] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Namgyu Cho, Seongwhan Lee, Sanja Fidler, Raquel Urtasun, and Alan L Yuille. The role of context for object detection and semantic segmentation in the wild. pages 891–898, 2014.

[48] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulò, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *ICCV*, 2017.

[49] Dim P Papadopoulos, Jasper RR Uijlings, Frank Keller, and Vittorio Ferrari. Extreme clicking for efficient object annotation. In *ICCV*, pages 4930–4939, 2017.

[50] George Papandreou, Liang-Chieh Chen, Kevin P Murphy, and Alan L Yuille. Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation. In *ICCV*, pages 1742–1750, 2015.

[51] Rui Qian, Yunchao Wei, Honghui Shi, Jiachen Li, Jiaying Liu, and Thomas Huang. Weakly supervised scene parsing with point-based distance metric learning. In *AAAI*, volume 33, pages 8843–8850, 2019.

[52] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM ToG*, 2004.

[53] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 2015.

[54] Olga Russakovsky, Li-Jia Li, and Li Fei-Fei. Best of both worlds: human-machine collaboration for object annotation. In *CVPR*, pages 2121–2131, 2015.

[55] Hao Su, Jia Deng, and Li Fei-Fei. Crowdsourcing annotations for visual object detection. In *AAAI*, 2012.

[56] Avan Suinesiaputra, Brett R Cowan, Ahmed O Al-Agamy, Mustafa A Elattar, Nicholas Ayache, Ahmed S Fahmy, Ayman M Khalifa, Pau Medrano-Gracia, Marie-Pierre Jolly, Alan H Kadish, et al. A collaborative resource to build consensus for automated left ventricular segmentation of cardiac mr images. *Medical image analysis*, 18(1):50–62, 2014.

[57] Xiaolu Sun, C Mario Christoudias, and Pascal Fua. Free-shape polygonal object localization. In *ECCV*, pages 317–332. Springer, 2014.

[58] Meng Tang, Abdelaziz Djelouah, Federico Perazzi, Yuri Boykov, and Christopher Schroers. Normalized cut loss for weakly-supervised cnn segmentation. In *CVPR*, pages 1818–1827, 2018.

[59] Bin Wang, Guojun Qi, Sheng Tang, Tianzhu Zhang, Yunchao Wei, Linghui Li, and Yongdong Zhang. Boundary perception guidance: a scribble-supervised semantic segmentation approach. In *IJCAI*, pages 3663–3669, 2019.

[60] Xinlong Wang, Tao Kong, Chunhua Shen, Yuning Jiang, and Lei Li. SOLO: Segmenting objects by locations. *arXiv preprint arXiv:1912.04488*, 2019.

[61] Yunchao Wei, Jiashi Feng, Xiaodan Liang, Ming-Ming Cheng, Yao Zhao, and Shuicheng Yan. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In *CVPR*, pages 1568–1576, 2017.

[62] Yunchao Wei, Xiaodan Liang, Yunpeng Chen, Xiaohui Shen, Ming-Ming Cheng, Jiashi Feng, Yao Zhao, and Shuicheng Yan. Stc: A simple to complex framework for weakly-supervised semantic segmentation. *TPAMI*, 39(11):2314–2320, 2016.

[63] Yunchao Wei, Huaxin Xiao, Honghui Shi, Zequn Jie, Jiashi Feng, and Thomas S Huang. Revisiting dilated convolution: A simple approach for weakly-and semi-supervised semantic segmentation. In *CVPR*, pages 7268–7277, 2018.

[64] Jiajun Wu, Yibiao Zhao, Jun-Yan Zhu, Siwei Luo, and Zhuowen Tu. Milcut: A sweeping line multiple instance learning paradigm for interactive image segmentation. In *CVPR*, pages 256–263, 2014.

[65] 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. 2020.

[66] Ning Xu, Brian Price, Scott Cohen, Jimei Yang, and Thomas Huang. Deep grabcut for object selection. *arXiv preprint arXiv:1707.00243*, 2017.

[67] Ning Xu, Brian Price, Scott Cohen, Jimei Yang, and Thomas S Huang. Deep interactive object selection. In *CVPR*, pages 373–381, 2016.

[68] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017.

[69] Hengshuang Zhao, Yi Zhang, Shu Liu, Jianping Shi, Chen Change Loy, Dahua Lin, and Jiaya Jia. Psanet: Point-wise spatial attention network for scene parsing. In *ECCV*, 2018.

[70] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, pages 633–641, 2017.