Focused and Collaborative Feedback Integration for Interactive Image Segmentation

Qiaoqiao Wei Hui Zhang Jun-Hai Yong
School of Software, BNRist, Tsinghua University, Beijing, China
wqq18@mails.tsinghua.edu.cn {huizhang,yongjh}@tsinghua.edu.cn

Abstract

Interactive image segmentation aims at obtaining a segmentation mask for an image using simple user annotations. During each round of interaction, the segmentation result from the previous round serves as feedback to guide the user's annotation and provides dense prior information for the segmentation model, effectively acting as a bridge between interactions. Existing methods overlook the importance of feedback or simply concatenate it with the original input, leading to underutilization of feedback and an increase in the number of required annotations. To address this, we propose an approach called Focused and Collaborative Feedback Integration (FCFI) to fully exploit the feedback for click-based interactive image segmentation. FCFI first focuses on a local area around the new click and corrects the feedback based on the similarities of high-level features. It then alternately and collaboratively updates the feedback and deep features to integrate the feedback into the features. The efficacy and efficiency of FCFI were validated on four benchmarks, namely GrabCut, Berkeley, SBD, and DAVIS. Experimental results show that FCFI achieved new state-of-the-art performance with less computational overhead than previous methods. The source code is available at https://github.com/veizgyauzgyauz/FCFI.

1. Introduction

Interactive image segmentation aims to segment a target object in an image given simple annotations, such as bounding boxes [17, 32, 37, 38, 40], scribbles [1, 10, 18], extreme points [2, 27, 29, 42], and clicks [6, 15, 23, 24, 34, 39]. Due to its inherent characteristic, i.e., interactivity, it allows users to add annotations and receive refined segmentation results iteratively. Unlike semantic segmentation, interactive image segmentation can be applied to unseen categories (categories that do not exist in the training dataset), demonstrating its generalization ability. Additionally, compared with instance segmentation, interactive segmentation is specific since it only localizes the annotated instance. Owing to these advantages, interactive image segmentation is a preliminary step for many applications, including image editing and image composition. This paper specifically focuses on interactive image segmentation using click annotations because clicks are less labor-intensive to obtain than other types of annotations.

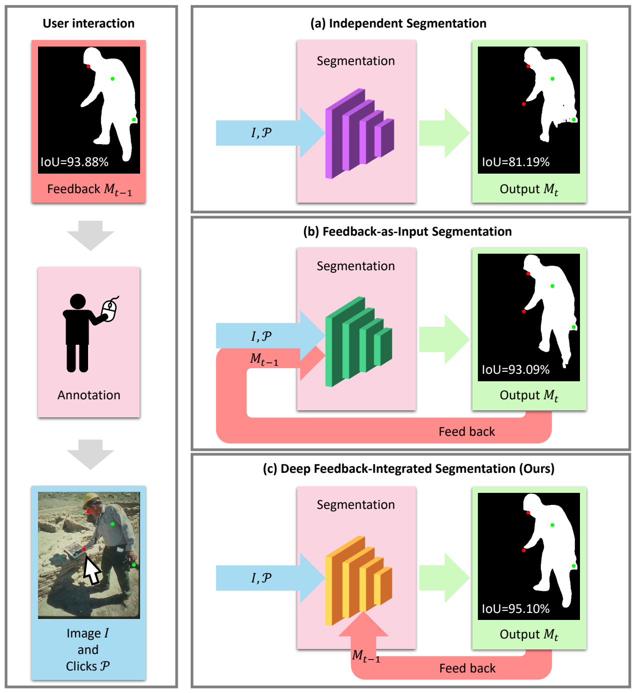

Figure 1. An overview of the interactive system and the comparison among (a) independent segmentation [24], (b) feedback-as-input segmentation [35], and (c) deep feedback-integrated segmentation. See the description in Sec. 1. Throughout this paper, green/red clicks represent foreground/background annotations.

The process of interaction is illustrated in Fig. 1. Given an image, users start by providing one or more clicks to label background or foreground pixels. The segmentation model then generates an initial segmentation mask for the target object based on the image and the clicks. If the segmentation result is unsatisfactory, users can continue to interact with the segmentation system by marking new clicks to indicate wrongly segmented regions and obtain a new segmentation mask. From the second round of interaction, the segmentation result of the previous interaction - referred to as *feedback* in this paper - will be fed back into the current interaction. This feedback is instrumental in user labeling and provides the segmentation model with prior information, which can facilitate convergence and improve the accuracy of segmentation [35].

Previous methods have made tremendous progress in interactive image segmentation. Some of them, such as [24,26,35], focused on finding useful ways to encode annotated clicks, while others, like [5,15,19,21,34], explored efficient neural network architectures to fully utilize user annotations. However, few methods have investigated how to exploit informative segmentation feedback. Existing methods typically treated each round of interaction independent [5, 12, 15, 19, 24, 26] or simply concatenated feedback with the initial input [6,23,25,34,35]. The former approach (Fig. 1(a)) failed to leverage the prior information provided by feedback, resulting in a lack of consistency in the segmentation results generated by two adjacent interactions. In the latter case (Fig. 1(b)), feedback was only directly visible to the first layer of the network, and thus the specific spatial and semantic information it carried would be easily diluted or even lost through many convolutional layers, similar to the case of annotated clicks [12].

In this paper, we present Focused and Collaborative Feedback Integration (FCFI) to exploit the segmentation feedback for click-based interactive image segmentation. FCFI consists of two modules: a Focused Feedback Correction Module (FFCM) for local feedback correction and a Collaborative Feedback Fusion Module (CFFM) for global feedback integration into deep features. Specifically, the FFCM focuses on a local region centered on the new annotated click to correct feedback. It measures the feature similarities between each pixel in the region and the click. The similarities are used as weights to blend the feedback and the annotated label. The CFFM adopts a collaborative calibration mechanism to integrate the feedback into deep layers (Fig. 1(c)). First, it employs deep features to globally update the corrected feedback for further improving the quality of the feedback. Then, it fuses the feedback with deep features via a gated residual connection. Embedded with FCFI, the segmentation network leveraged the prior information provided by the feedback and outperformed many previous methods.

2. Related Work

The Method changes in interactive segmentation. Early work approached interactive segmentation based on graph models, such as graph cuts [3, 32] and random walks [8, 10]. These methods build graph models on the low-level features of input images. Therefore, they are sensitive to user annotations and lack high-level semantic information. Xu et al. [39] first introduced deep learning into interactive image segmentation. Deep convolutional neural networks are optimized over large amounts of data, which contributes to robustness and generalization. Later deep-learning-based methods [5, 15, 19] showed striking improvements.

The Local refinement for interactive segmentation. To refine the primitive predictions, previous methods [5, 6, 12, 20] introduced additional convolutional architectures to fulfill the task. These methods followed a coarse-to-fine scheme and performed global refinement. Later, Sofiiuk et al. [34] proposed the Zoom-In strategy. From the third interaction, it cropped and segmented the area within an expanded bounding box of the inferred object. Recent approaches [6, 23] focused on local refinement. After a forward pass of the segmentation network, they found the largest connected component on the difference between the current prediction and the previous prediction. Then, they iteratively refined the prediction in the region using the same network or an extra lighter network. The proposed FFCM differs from these methods in that it does not need to perform feedforward propagation multiple times. Instead, it utilizes the features already generated by the backbone of the network to locally refine the feedback once. Thus, it is non-parametric and fast.

The Exploitation of segmentation feedback. Prior work has found that segmentation feedback has instructive effects on the learning of neural networks. Mahadevan et al. [25] were the first to incorporate feedback from a previous iteration into the segmentation network. They took the feedback as an optional channel of the input. Later methods [6, 23, 34, 35] followed this and concatenated the feedback with the input. However, this naive operation may be suboptimal for feedback integration due to the dilution problem [12]. Different from previous approaches, the proposed CFFM integrates feedback into deep features. Since both the feedback and the high-level features contain semantic knowledge of specific objects, fusing them together helps to improve the segmentation results.

3. Method

3.1. Interaction Pipeline

The pipeline of our method is illustrated in Fig. 2. The process includes two parts: interaction and segmentation.

Interaction. In the interaction step, users give hints about the object of interest to the segmentation model by providing clicks. In the first interaction, users provide clicks only based on the input image $\boldsymbol{I} \in \mathbb{R}^{H\times W\times 3}$, where H and W denote the height and width, respectively; in the subse-

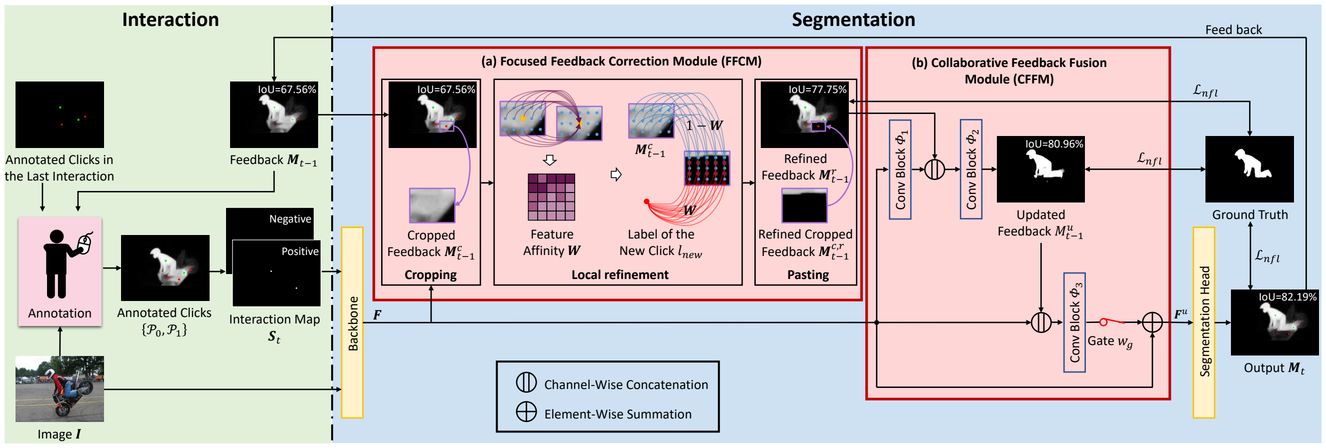

Figure 2. The pipeline of the proposed method. See detailed description in Sec. 3.1.

quent interactions, users mark more clicks according to the input image and the segmentation feedback. The annotated clicks over all interactions are denoted as $\{\mathcal{P}_0, \mathcal{P}_1\}$, where $\mathcal{P}_0$ and $\mathcal{P}_1$ represent the background and foreground click sets, respectively, and each element in $\mathcal{P}_0$ and $\mathcal{P}_1$ is the xy position of a click. The annotated clicks are converted into an interaction map $\boldsymbol{S} \in \mathbb{R}^{H\times W\times 2}$ using disk encoding [2], which represents each click as a disk with a small radius.

Segmentation. In the segmentation step, a commonly used fully convolutional baseline architecture embedded with FCFI is utilized as the segmentation network. The baseline network comprises a backbone and a segmentation head, as shown in the yellow blocks in Fig. 2. The backbone gradually captures long-range dependencies of pixels to generate high-level features, and the segmentation head, including a Sigmoid function, recovers spatial resolutions and extracts semantic information. To correct feedback locally and incorporate it into the segmentation network, an FFCM and a CFFM are inserted between the backbone and the segmentation head. In the t-th interaction, the input image $\boldsymbol{I}$ and the interaction map $\boldsymbol{S}_t$ are concatenated in depth and fed into the backbone, and the feedback $\boldsymbol{M}_{t-1}$ is directly fed into the FFCM. The segmentation network outputs a segmentation mask $\boldsymbol{M}_t \in \mathbb{R}^{H\times W}$.

3.2. Focused Feedback Correction Module

From the second interaction, users are expected to place new clicks on the mislabeled regions. However, most previous methods [5, 15, 15, 26, 34, 35] treated all clicks equally and only performed global prediction, which may weaken the guidance effect of the newly annotated clicks and cause unexpected changes in correctly segmented regions that are far away from the newly annotated click [6]. For instance, in Fig. 1(a) and (b), comparing $\boldsymbol{M}_{t-1}$ and $\boldsymbol{M}_t$, adding a new negative click near the person's right hand modified the segmentation result in the person's feet. Considering that clicks are usually marked to correct small regions of a segmentation mask, locally refining the feedback can not only preserve the segmentation results in other regions but also reduce processing time. Therefore, we propose the FFCM to correct the feedback from a local view.

As shown in Fig. 2(a), the feedback modification in the FFCM requires three steps: cropping, local refinement, and pasting. We first narrow down the refinement region, then refine the feedback using the pixel similarities in the high-level feature space, and finally fuse the refined feedback with the original one.

Cropping. To exclude irrelevant regions and focus on the area around the newly annotated click, we select a small rectangle patch centered on the new annotated click and crop it from the features $\boldsymbol{F} \in \mathbb{R}^{H'\times W'\times 3}$ generated by the backbone, where H' and W' are the height and width of $\boldsymbol{F}$. The patch has a size of $rH' \times rW'$, where the expansion ratio r is 0.3 by default. The feedback $\boldsymbol{M}_{t-1}$ is cropped in the same way. The cropped features and the cropped feedback are denoted as $\boldsymbol{F}^c$ and $\boldsymbol{M}^c_{t-1}$, respectively.

Local refinement. In this step, per-pixel matching is performed. We measure the feature affinity between each pixel in the patch and the new annotated click and then blend the feedback with the annotated label of the new click according to the affinity. The feature affinity $\boldsymbol{W} \in \mathbb{R}^{rH'\times rW'}$ is defined as the cosine similarity:

$$\boldsymbol{W}(i,j) = \frac{\boldsymbol{F}^c(i,j)\boldsymbol{F}^c(x_{new}, y_{new})}{||\boldsymbol{F}^c(i,j)||_2||\boldsymbol{F}^c(x_{new}, y_{new})||_2}, \quad (1)$$

where (x_{new}, y_{new}) is the coordinate of the new annotated click in the patch. Each element in the feature affinity $\boldsymbol{W}$ is between 0 to 1. The larger the affinity is, the more likely that the pixel belongs to the same class (background/foreground) as the new annotated click. The refined cropped feedback $\boldsymbol{M}^r_{t-1}$ is generated by blending the original feedback $\boldsymbol{M}^c_{t-1}$ and the annotated label of the new click

l_{new} via a linear combination:

$$M_{t-1}^{c,r} = l_{new} \cdot \boldsymbol{W} + (1 - \boldsymbol{W}) \odot M_{t-1}^{c}. \quad (2)$$

Pasting. After obtaining the refined cropped feedback, we paste it back to the original position on the feedback M_{t-1} and denote the refined full-size feedback as M_{t-1}^{r}.

To enable optimization in backward-propagation during training, the cropping and pasting operations are not applied in the FFCM. Instead, a box mask $M_{box} \in \mathbb{R}^{H' \times W'}$ is used to filter out the pixels that are far away from the new annotated click. A local region is selected first. It is centered on the new annotated click and has a size of $rH' \times rW'$. Pixels within the region are set to 1, and others are set to 0. Similar to the local refinement, we perform global refinement on the full-size features and feedback. A full-size refined feedback $M_{t-1}^{r'}$ is obtained. Finally, we keep the focused area and obtain the corrected feedback M_{t-1}^{r} as

$$M_{t-1}^{r} = M_{box} \odot M_{t-1}^{r'} + (1 - M_{box}) \odot M_{t-1}. \quad (3)$$

3.3. Collaborative Feedback Fusion Module

In an interaction, a user usually adds new clicks based on the segmentation feedback. Basically, interactions are successive, and previous segmentation results provide prior information about the location and shape of the target object for the current interaction. Therefore, it is natural to integrate feedback into the segmentation network, and this integration is supposed to improve the segmentation quality.

Previous methods [6, 34, 35] simply concatenated the feedback with the input image and the interaction map and then fed them into the segmentation network. However, experimental results show that this naive method cannot exploit the prior information provided by the feedback. There are two reasons. First, early fusion - fusing at the beginning or shallow layers of a network - easily causes information dilution or loss [12]. Second, from the semantic information perspective, the feedback contains dense semantic information compared with the low-level input image and sparse clicks. Thus, theoretically, fusing the feedback into deep features rather than the input enables the segmentation network to obtain segmentation priors while preserving the extracted semantic information.

The CFFM is introduced in this paper to integrate segmentation feedback into high-level features at deep layers of the segmentation network. Fig. 2(b) illustrates the architecture of the CFFM. The CFFM comprises two information pathways, called feedback pathway and feature pathway, respectively. The two pathways utilize the feedback and the high-level features to collaboratively update each other.

Feedback pathway. This pathway performs global refinement on the feedback with the aid of deep features. First, after being encoded by a convolution block Φ_1, the features $\boldsymbol{F}$ are concatenated with the corrected feedback signal M_{t-1}^{r} in the channel dimension. Then, convolutional layers followed by a Sigmoid function, denoted as Φ_2, are applied to update the feedback:

$$M_{t-1}^{u} = \Phi_2(\text{concat}(\Phi_1(\boldsymbol{F}; \theta_1), M_{t-1}^{r}); \theta_2), \quad (4)$$

where θ_1 and θ_2 denote the learnable parameters of Φ_1 and Φ_2, and "concat$(\cdot,\cdot)$" denotes channel-wise concatenation.

Feature pathway. In this pathway, we update the high-level features with the feedback as a guide. The features are fused with the updated feedback M_{t-1}^{u} through a convolution block Φ_3. To avoid negative outcomes from useless learned features, we update the features $\boldsymbol{F}$ via a skip connection following ResNet [14]. However, the skip connection leads to an inaccurate prediction of the final output M_t in the first interaction, but not in the other interactions. The reason is that the feedback is initialized to an all-zero map and has the maximum information entropy in the first interaction [25]. Consequently, it may add noise to deep features. The reliability of feedback grows dramatically from the first interaction to the second interaction, making the feedback more instructive. To tackle this problem, we introduce a gate w_g to control the information flow from the feedback pathway. The gate is set to 0 in the first interaction to prevent the inflow of noise from hurting performance, and it is set to 1 in the subsequent interactions. Mathematically, the fused features $\boldsymbol{F}^u$ can be formulated as

$$\boldsymbol{F}^{u} = w_g \cdot \Phi_3(\text{concat}(\boldsymbol{F}, M_{t-1}^{u}); \theta_3) + \boldsymbol{F}, \quad (5)$$

where θ_3 denotes the learnable parameters of Φ_3.

3.4. Training Loss

The normalized focal loss L_{nfl} [33] is applied as the objective function because training with it yields better performance in interactive image segmentation, which is demonstrated in RITM [35]. The constraint is imposed on M_{t-1}^{r}, M_t, and M_{t-1}^{u}. Particularly, we only calculate the loss in the cropped area for M_{t-1}^{r}. The full loss function is

$$\begin{aligned} L = L_{nfl}(M_{t-1}^{r}, M_{gt}) + L_{nfl}(M_{t-1}^{u}, M_{gt}) \\ + L_{nfl}(M_t, M_{gt}), \end{aligned} \quad (6)$$

where M_{gt} denotes the ground truth segmentation mask.

4. Experiments

4.1. Experimental Settings

Segmentation backbones. We conducted experiments using DeepLabV3+ [4] and HRNet+OCR [36, 41] as the baseline network, respectively. The ResNet-101 [14], HRNet-18s [36], and HRNet-18 [36] were employed as backbones. Improvements introduced by different network baselines are not the focus of this paper. Therefore, we mainly report experimental results achieved by

Method	Backbone	Train set	GrabCut		Berkeley	SBD		DAVIS	
			NoC@85%	NoC@90%	NoC@90%	NoC@85%	NoC@90%	NoC@85%	NoC@90%
DOS w/o GC [39]	FCN-8s	SBD	8.02	12.59	-	14.30	16.79	12.52	17.11
DOS w/ GC [39]	FCN-8s	SBD	5.08	6.08	-	9.22	12.80	9.03	12.58
RIS-Net [20]	VGG-16	Pascal VOC	-	5.00	-	6.03	-	-	-
LD [19]	VGG-19	SBD	3.20	4.79	-	7.41	-	5.95	9.57
CAG [26]	FCN-8s	Augmented SBD	-	3.58	5.60	-	-	-	-
BRS [15]	DenseNet	SBD	2.60	3.60	5.08	6.59	9.78	5.58	8.24
f-BRS-B [34]	ResNet-101	SBD	2.30	2.72	4.57	4.81	7.73	5.04	7.41
FCA-Net [24]	ResNet-101	Augmented SBD	1.88	2.14	4.19	-	-	5.38	7.90
IA+SA [16]	ResNet-101	Augmented SBD	-	3.07	4.94	-	-	5.16	-
CDNet [5]	ResNet-101	SBD	2.42	2.76	3.65	4.73	7.66	5.33	6.97
FocusCut [23]	ResNet-101	SBD	**1.46**	**1.64**	3.01	3.40	**5.31**	4.85	**6.22**
Ours	ResNet-101	SBD	1.64	1.80	**2.84**	**3.26**	5.35	**4.75**	6.48
RITM [35]	HRNet-18s	COCO+LVIS	1.54	1.68	2.60	4.26	6.86	4.79	6.00
FocalClick-S1 [6]	HRNet-18s	COCO+LVIS	1.72	1.94	3.40	4.75	7.22	5.19	7.95
FocalClick-S2 [6]	HRNet-18s	COCO+LVIS	1.52	1.66	2.41	4.37	6.59	4.20	5.49
Ours	HRNet-18s	COCO+LVIS	**1.50**	**1.56**	**2.05**	**3.88**	**6.24**	**3.70**	**5.16**
EdgeFlow [12]	HRNet-18	COCO+LVIS	1.60	1.72	2.40	-	-	4.54	5.77
RITM [35]	HRNet-18	COCO+LVIS	1.42	1.54	2.26	3.80	6.06	4.36	5.74
Ours	HRNet-18	COCO+LVIS	**1.38**	**1.46**	**1.96**	**3.63**	**5.83**	**3.97**	**5.16**

Table 1. Evaluation results on the GrabCut, Berkeley, SBD, and DAVIS datasets. The augmented SBD is a combination of the Pascal VOC training set [9], a part of the SBD training set, and a part of the SBD validation set. Throughout this essay, the best and the second-best results for different mainstream backbones are written in bold and underlined, respectively.

B	Method	Berkeley					DAVIS				
		NoF_{20}@90%↓	IoU@5↑	BIoU@5↑	ASSD@5↓	SPC↓	NoF_{20}@90%↓	IoU@5↑	BIoU@5↑	ASSD@5↓	SPC↓
ResNet-101	f-BRS-B [34]	6	0.875	0.730	4.626	0.072	77	0.826	0.717	11.267	0.102
	FCA-Net [24]	7	0.923	0.793	2.190	0.059	74	0.867	0.771	9.048	0.075
	CDNet [24]	4	0.921	0.803	2.576	0.079	60	0.876	0.783	8.973	0.108
	FocusCut [23]	**3**	0.933	0.811	2.050	3.152	61	0.873	0.778	8.880	3.872
	Baseline	6	0.904	0.782	4.370	**0.052**	75	0.867	0.770	9.421	**0.069**
	Ours	**3**	**0.943**	**0.838**	**1.789**	0.057	**59**	**0.893**	**0.815**	**8.226**	0.082
HRNet-18s	RITM [35] †	1	0.950	0.859	1.312	**0.035**	53	0.883	0.801	8.378	**0.036**
	FocalClick-S1 [6]	4	0.946	0.849	1.565	0.065	84	0.877	0.784	8.364	0.084
	FocalClick-S2 [6]	1	0.957	**0.889**	1.170	0.052	54	0.897	0.824	7.635	0.082
	Ours	**0**	**0.958**	0.883	**1.012**	0.044	**51**	**0.907**	**0.830**	**6.434**	0.048

Table 2. Comparisons of effectiveness and efficiency on the Berkeley and DAVIS datasets. "B" in the table header denotes the term "backbone". †RITM is the baseline of our method.

DeepLabV3+ and provide essential results achieved by HRNet+OCR. The CFFM was inserted after the upsampling operator for DeepLabV3+ and the HRNet for HRNet+OCR. Both the ResNet-101 backbone and the HRNet-18 backbone were pre-trained on the ImageNet dataset [7].

Datasets. The DeepLabV3+ was trained on the training set of SBD [13], and the HRNet+OCR was trained on the combination of COCO [22] and LVIS [11]. We evaluated our method on four benchmarks: GrabCut [32], Berkeley [28], SBD, and DAVIS [31]. SBD contains 8,498 images for training, and its validation set contains 2,820 images including 6,671 instance-level masks. COCO+LVIS contains 104k images with 1.6M instance-level masks. GrabCut contains 50 images with an instance mask for each image. Berkeley consists of 96 images and 100 instance masks. For DAVIS, we used the same 345 sampled frames as previous methods [6, 23, 24, 34, 35] for evaluation.

Implementation details. During training, we resized input images with a random scale factor between 0.75 and 1.40 and randomly cropped them to 320×480. A horizontal flip and random jittering of brightness, contrast, and RGB values were also applied. We utilized an Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ to train the network for 60 epochs. The learning rate for the backbone was initialized to 5×10^{-5} and multiplied by 0.1 on the 50th epoch. The learning rate for other parts of the segmentation network was 10 times larger than that for the backbone. The batch size was set to 24. Following RITM, we used the iterative sampling strategy to simulate user interaction. For each batch, new annotated clicks were iteratively sampled for 1 to 4 iterations. In each iteration, up to 24 annotated clicks were randomly sampled from the eroded mislabelled regions of the last prediction.

During inference, only one click was added in each

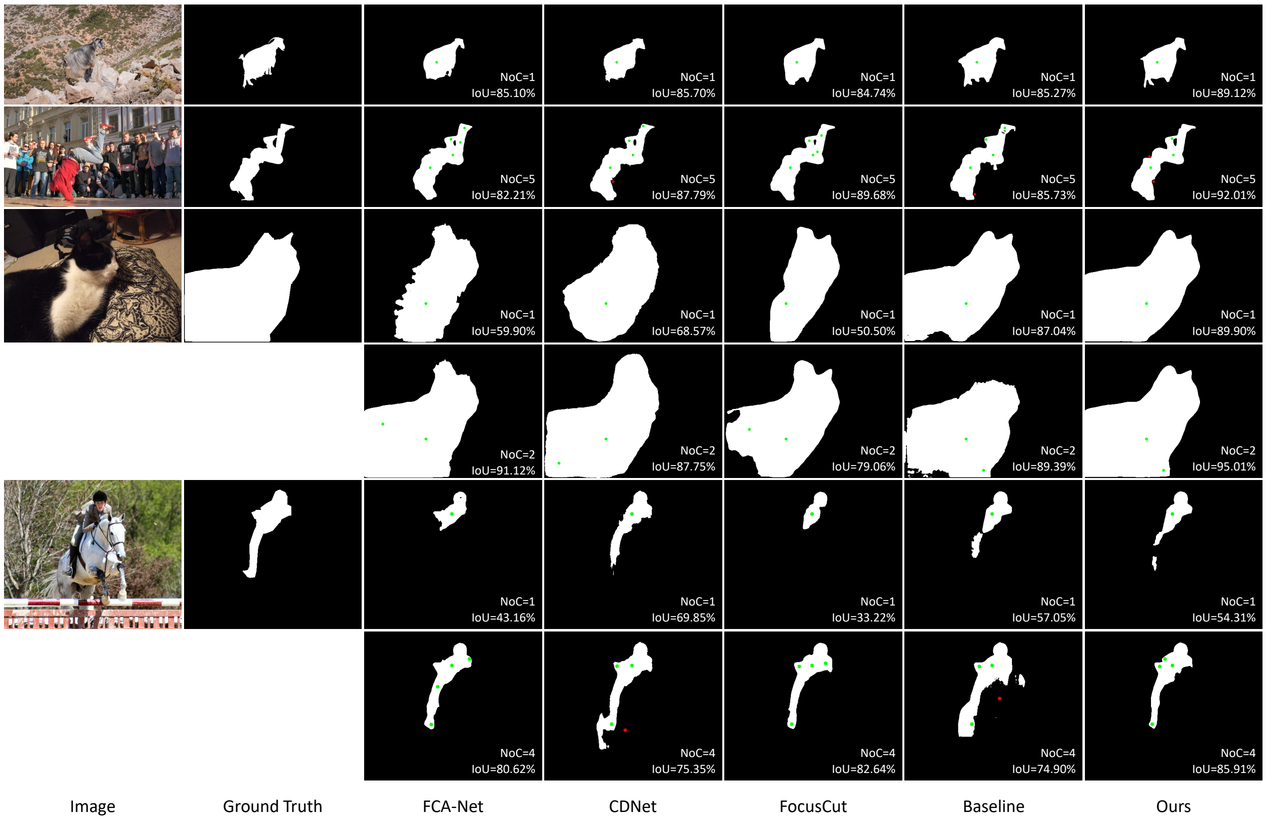

Figure 3. Qualitative comparisons of FCA-Net [24], CDNet [5], FocusCut [23], our baseline, and our method.

interaction. The new click was sampled from misclassified regions and was the farthest from the region boundaries. Mathematically, the distance between two points $\boldsymbol{p}$ and $\boldsymbol{q}$ is defined as $d(\boldsymbol{p}, \boldsymbol{q}) = ||\boldsymbol{p} - \boldsymbol{q}||_2$, and the distance between a point $\boldsymbol{p}$ and a point set $\mathcal{P}$ is defined as $d(\boldsymbol{p}, \mathcal{P}) = \min_{\boldsymbol{q} \in \mathcal{P}} d(\boldsymbol{p}, \boldsymbol{q})$. The set of false positive points and false negative points is defined as $\mathcal{P}_f = \{\boldsymbol{p} | \boldsymbol{M}_t(\boldsymbol{p}) = 0, \boldsymbol{M}_{gt}(\boldsymbol{p}) = 1\} \cup \{\boldsymbol{p} | \boldsymbol{M}_t(\boldsymbol{p}) = 1, \boldsymbol{M}_{gt}(\boldsymbol{p}) = 0\}$. For a pixel $p \in \mathcal{P}_f$, the largest connected component in which it is located is denoted as $\mathcal{P}_c(\boldsymbol{p})$. The distance from $\boldsymbol{p}$ to the boundaries of $\mathcal{P}_c(\boldsymbol{p})$ is defined as $\eta(\boldsymbol{p}) = d(\boldsymbol{p}, \mathcal{P}_c^C(\boldsymbol{p}))$, where $\mathcal{P}_c^C(\boldsymbol{p})$ denotes the complement of $\mathcal{P}_c(\boldsymbol{p})$. Following the standard protocol [6, 15, 23, 24, 34, 35], the new annotated pixel was selected by $\boldsymbol{p}^* = \{\boldsymbol{p} | \max_{\boldsymbol{p} \in \mathcal{P}_f} \eta(\boldsymbol{p})\}$. The maximum number of clicks was limited to 20 for each sample in all experiments. Besides, following previous methods [5, 6, 34, 35], we adopted test time augmentations, i.e., the Zoom-In strategy and averaging the predictions of the original image and the horizontally flipped image.

The proposed method was implemented in PyTorch [30]. All the experiments were conducted on a computer with an Intel Xeon Gold 6326 2.90 GHz CPU and NVIDIA GeForce RTX 3090 GPUs.

Evaluation metrics. The proposed method was evaluated using the following six metrics: 1) NoC@α: the mean number of clicks (NoC) required to reach a predefined intersection over union (IoU) threshold α for all images; 2) IoU@N: the mean IoU achieved by a particular NoC N; 3) BIoU@N: the mean boundary IoU achieved by a particular NoC N; 4) ASSD@N: the average symmetric surface distance with a particular NoC N, which was used to evaluate the boundary quality of the prediction; 5) NoF$_N$@α: the number of failures (NoF) that cannot reach a target IoU α with a certain NoC N; 6) SPC: the average running time in seconds per click.

4.2. Comparison with Previous Work

4.2.1 Effectiveness Analysis

We have tabulated the quantitative results in Tab. 1 and Tab. 2. The quantitative results demonstrate that our method can generalize across various datasets and different backbones. The number of clicks of our method required to reach 85% and 90% IoU are much lower than previous methods, and our method also outperformed previous work in boundary quality. The results indicate that our method can achieve satisfactory segmentation results with less user effort. The line charts of mIoU-NoC on four datasets are plotted in Fig. 4. Those results indicate that our method achieved ac-

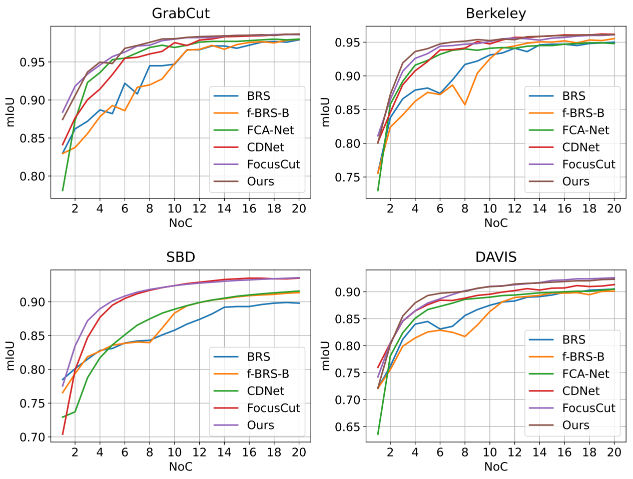

Figure 4. Comparisons of the mIoU-NoC curves on four datasets.

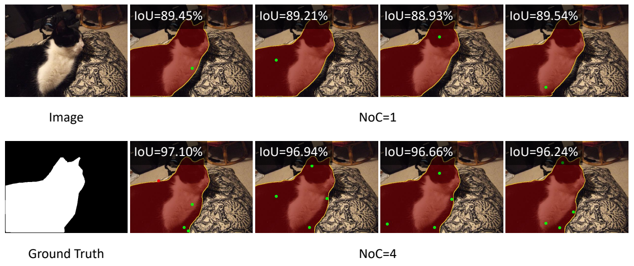

Figure 5. Results obtained by different annotations.

ceptable results given only a few clicks, steadily improved segmentation results with additional clicks, and ultimately converged to better results. Fig. 3 visualizes the qualitative results of some previous work and ours. Compared with other methods, our method required fewer clicks to obtain relatively complete segments and fine boundary details. Additionally, it could handle challenging scenarios, including color interference (like the goat and the cat), complex backgrounds (as depicted in the break-dancer photo), and occlusions (as seen from the racehorse rider). Please refer to the supplementary material for more visualization examples. Fig. 5 exhibits the sensitivity of our method to click locations. Our method attained approximate performance for different annotation positions when provided with reasonable annotations.

4.2.2 Efficiency Analysis

Tab. 2 presents the average inference speed of different methods, among which our method achieved a desirable trade-off between speed and accuracy. Notably, our proposed modules exhibited a low computational budget, requiring less than 13 ms for all modules, compared to the baseline. In summary, the proposed framework achieved competitive results with relatively low computation costs.

Method	Backbone	Berkeley		DAVIS	
		NoC @90%↓	NoF$_{20}$ @90%↓	NoC @90%↓	NoF$_{20}$ @90%↓
Baseline	ResNet-101	4.31	6	7.56	75
+ FFC	ResNet-101	3.75	4	6.75	65
+ CFF	ResNet-101	3.07	**1**	6.62	66
+ FFC + CFF	ResNet-101	**2.84**	3	**6.48**	**59**
Baseline	HRNet-18	2.60	1	5.73	54
+ FFC	HRNet-18	2.11	1	5.52	54
+ CFF	HRNet-18	2.05	**0**	5.32	52
+ FFC + CFF	HRNet-18	**1.96**	**0**	**5.16**	**51**

Table 3. An ablation study for the core components.

4.3. Ablation Study

We evaluated the efficacy of each proposed component. Berkeley and DAVIS were chosen as the main evaluation datasets because they cover challenging scenarios, such as unseen categories, motion blur, and occlusions. Besides, they have better annotation quality than SBD.

Tab. 3 tabulates the quantitative ablation results. To integrate the feedback into the network, the corrected feedback M^c_{t-1} was concatenated with the features $\boldsymbol{F}$ in the "+ FFC" variant. When only one module was embedded, both the FFCM and the CFFM improved the results. The CFFM boosted the performance more; this proves the effectiveness of deep feedback integration. The FFCM has also improved the results. The reason is that the CFFM relies on the quality of the feedback, and the FFCM provides refined feedback for it. With the two modules working synergistically, our method significantly reduced the NoC and NoF.

Qualitative results for the FFCM and the CFFM are shown in Fig. 6. The FFCM utilizes the feature affinity to refine the feedback from a local perspective. For instance, in the 14th round of interaction, the FFCM yielded finer segmentation boundaries on the front fender of the motorcycle. The CFFM refines the feedback in a global view and updates the deep features to improve the overall segmentation results, e.g., the boy wearing a hat and the bull.

Analysis of the FFCM. In Tab. 4, we analyzed the effect of different similarity measurements, e.g. exponential similarity and cosine similarity. Using cosine similarity to measure the feature affinity achieved better results. We also verified the robustness of the FFCM with respect to the expansion ratio r. As illustrated in Tab. 5, an expansion ratio of 0.3 yielded the best results.

Analysis of the CFFM. Different implementations of feedback fusion have been explored: 1) directly concatenating feedback with the input, 2) fusing feedback into deep features but removing the residual connection, i.e., removing the second term of Eq. 5, and 3) fusing feedback into deep features but removing the gate. As shown in Tab.

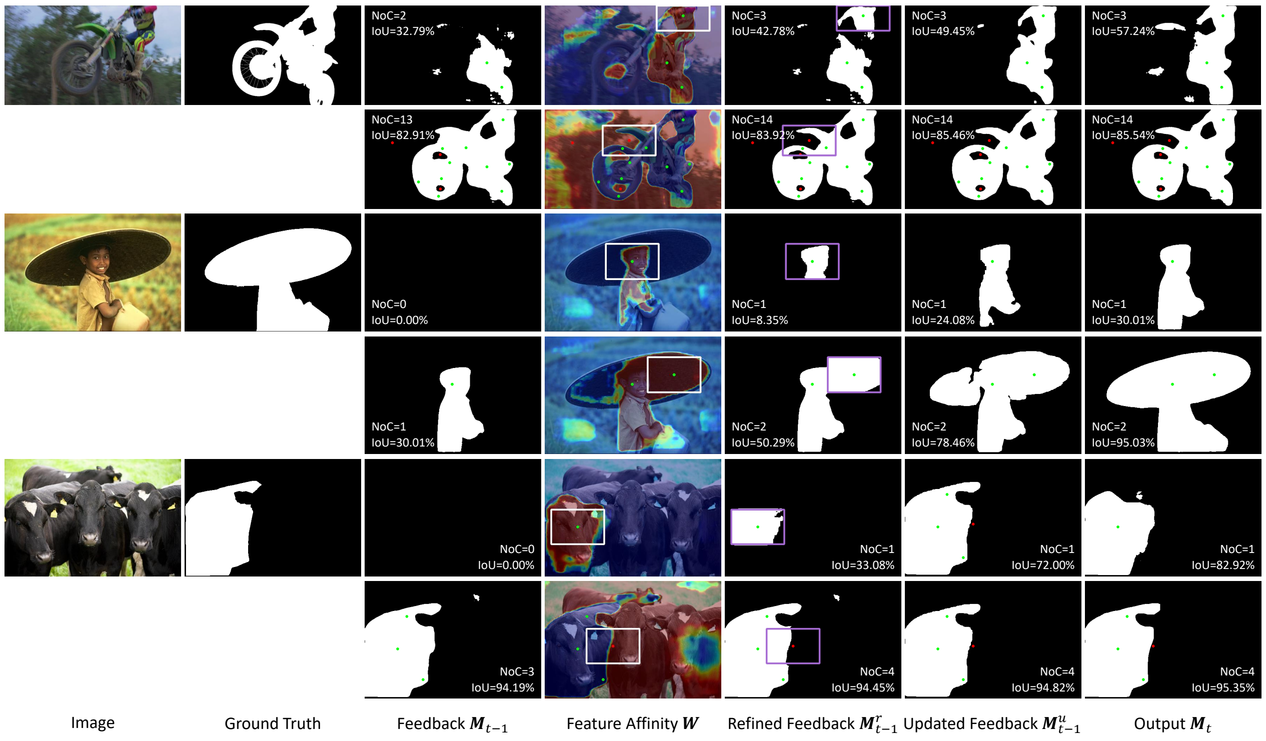

Figure 6. Visualization of the feedback, the feature affinity, the refined feedback, the updated feedback, and the output. The feature affinity is shown in a heatmap, where red color denotes high feature affinity, and blue color denotes low feature affinity.

Similarity Measurement	Berkeley		DAVIS	
	@85%	@90%	@85%	@90%
Exponential	2.06	3.60	5.26	7.51
Cosine	**1.88**	**2.85**	**4.60**	**5.82**

Table 4. The mean NoC with respect to the similarity measurement. The exponential similarity is defined as $\boldsymbol{W}(i,j) = e^{-||\boldsymbol{F}(i,j)-\boldsymbol{F}(x_{new},y_{new})||^2/\sigma}$.

Expansion Ratio r	Berkeley		DAVIS	
	@85%	@90%	@85%	@90%
0.1	1.91	3.43	5.52	7.46
0.2	1.97	3.10	5.12	6.82
0.3	1.88	**2.85**	**4.60**	**5.82**
0.4	**1.79**	3.12	4.74	6.42
0.5	1.84	3.27	5.15	6.72

Table 5. The mean NoC with respect to the expansion ratio r.

Method	Berkeley		DAVIS	
	@85%	@90%	@85%	@90%
w/o feedback	2.31	4.31	5.23	7.56
Concat feedback with input	2.01	4.00	5.03	7.08
CFFM w/o residual connection	1.98	3.52	4.91	6.63
CFFM w/o gate	1.96	3.36	4.85	6.37
CFFM	**1.81**	**3.07**	**4.75**	**6.10**

Table 6. The mean NoC with respect to different feedback fusion architectures.

6, integrating feedback into the network improves the performance, which demonstrates the instructive ability of the feedback. Compared with directly concatenating feedback with the input, feedback fusion in deep layers significantly reduces the required number of annotated clicks by 1.24 NoC@90% on Berkeley and 1.56 NoC@90% on DAVIS.

4.4. Limitations

Although our method benefits from the feedback guidance, it still has certain limitations. First, there is no guarantee that each round of interaction yields superior results compared to the previous one. Second, ambiguity has yet to be resolved in our method. For example, if a click is provided on a shirt, both the shirt and the person wearing it are likely to be the target object. Additionally, our method may struggle when handling thin structures, such as ropes, insect legs, and bicycle spokes.

5. Conclusion

The segmentation result from the last interaction (feedback) provides instructive information about the target object to the current interaction. To exploit the feedback, this paper proposes a deep feedback integration approach called FCFI. FCFI first performs local refinement on the feedback. Then, it collaboratively and globally updates the feedback and the features in deep layers of the segmentation network. We have experimentally demonstrated that FCFI has strong generalization capabilities and outperformed many previous methods with fast processing speed.

Acknowledgments

This research was supported by the National Key R&D Program of China (2020YFB1708900) and the National Nature Science Foundation of China (62021002).

References

[1] Junjie Bai and Xiaodong Wu. Error-tolerant scribbles based interactive image segmentation. In *CVPR*, pages 392–399, 2014. 1

[2] Rodrigo Benenson, Stefan Popov, and Vittorio Ferrari. Large-scale interactive object segmentation with human annotators. In *CVPR*, pages 11700–11709, 2019. 1, 3

[3] Yuri Y Boykov and M-P Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In *ICCV*, volume 1, pages 105–112, 2001. 2

[4] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, pages 801–818, 2018. 4

[5] Xi Chen, Zhiyan Zhao, Feiwu Yu, Yilei Zhang, and Manni Duan. Conditional diffusion for interactive segmentation. In *ICCV*, pages 7345–7354, 2021. 2, 3, 5, 6

[6] Xi Chen, Zhiyan Zhao, Yilei Zhang, Manni Duan, Donglian Qi, and Hengshuang Zhao. Focalclick: Towards practical interactive image segmentation. In *CVPR*, pages 1300–1309, 2022. 1, 2, 3, 4, 5, 6

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 5

[8] Xingping Dong, Jianbing Shen, Ling Shao, and Luc Van Gool. Sub-markov random walk for image segmentation. *IEEE TIP*, 25(2):516–527, 2015. 2

[9] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010. 5

[10] Leo Grady. Random walks for image segmentation. *IEEE TPAMI*, 28(11):1768–1783, 2006. 1, 2

[11] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *CVPR*, pages 5356–5364, 2019. 5

[12] Yuying Hao, Yi Liu, Zewu Wu, Lin Han, Yizhou Chen, Guowei Chen, Lutao Chu, Shiyu Tang, Zhiliang Yu, Zeyu Chen, and Baohua Lai. Edgeflow: Achieving practical interactive segmentation with edge-guided flow. In *ICCVW*, pages 1551–1560, 2021. 2, 4, 5

[13] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *ICCV*, pages 991–998, 2011. 5

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 4

[15] Won-Dong Jang and Chang-Su Kim. Interactive image segmentation via backpropagating refinement scheme. In *CVPR*, pages 5297–5306, 2019. 1, 2, 3, 5, 6

[16] Theodora Kontogianni, Michael Gygli, Jasper Uijlings, and Vittorio Ferrari. Continuous adaptation for interactive object segmentation by learning from corrections. In *ECCV*, pages 579–596, 2020. 5

[17] Victor Lempitsky, Pushmeet Kohli, Carsten Rother, and Toby Sharp. Image segmentation with a bounding box prior. In *ICCV*, pages 277–284, 2009. 1

[18] Yin Li, Jian Sun, Chi-Keung Tang, and Heung-Yeung Shum. Lazy snapping. *ACM TOG*, 23(3):303–308, 2004. 1

[19] Zhuwen Li, Qifeng Chen, and Vladlen Koltun. Interactive image segmentation with latent diversity. In *CVPR*, pages 577–585, 2018. 2, 5

[20] JunHao Liew, Yunchao Wei, Wei Xiong, Sim-Heng Ong, and Jiashi Feng. Regional interactive image segmentation networks. In *ICCV*, pages 2746–2754, 2017. 2, 5

[21] Jun Hao Liew, Scott Cohen, Brian Price, Long Mai, Sim-Heng Ong, and Jiashi Feng. Multiseg: Semantically meaningful, scale-diverse segmentations from minimal user input. In *ICCV*, pages 662–670, 2019. 2

[22] 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*, pages 740–755, 2014. 5

[23] Zheng Lin, Zheng-Peng Duan, Zhao Zhang, Chun-Le Guo, and Ming-Ming Cheng. Focuscut: Diving into a focus view in interactive segmentation. In *CVPR*, pages 2637–2646, 2022. 1, 2, 5, 6

[24] Zheng Lin, Zhao Zhang, Lin-Zhuo Chen, Ming-Ming Cheng, and Shao-Ping Lu. Interactive image segmentation with first click attention. In *CVPR*, pages 13339–13348, 2020. 1, 2, 5, 6

[25] Sabarinath Mahadevan, Paul Voigtlaender, and Bastian Leibe. Iteratively trained interactive segmentation. In *BMVC*, 2018. 2, 4

[26] Soumajit Majumder and Angela Yao. Content-aware multi-level guidance for interactive instance segmentation. In *CVPR*, pages 11602–11611, 2019. 2, 3, 5

[27] Kevis-Kokitsi Maninis, Sergi Caelles, Jordi Pont-Tuset, and Luc Van Gool. Deep extreme cut: From extreme points to object segmentation. In *CVPR*, pages 616–625, 2018. 1

[28] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, volume 2, pages 416–423, 2001. 5

[29] Dim P. Papadopoulos, Jasper R. R. Uijlings, Frank Keller, and Vittorio Ferrari. Extreme clicking for efficient object annotation. In *ICCV*, pages 4930–4939, 2017. 1

[30] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, volume 32, pages 8026–8037, 2019. 6

[31] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, pages 724–732, 2016. 5

[32] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "grabcut": interactive foreground extraction using iterated graph cuts. *ACM TOG*, 23(3):309–314, 2004. 1, 2, 5

[33] Konstantin Sofiiuk, Olga Barinova, and Anton Konushin. Adaptis: Adaptive instance selection network. In *ICCV*, pages 7355–7363, 2019. 4

[34] Konstantin Sofiiuk, Ilia Petrov, Olga Barinova, and Anton Konushin. f-brs: Rethinking backpropagating refinement for interactive segmentation. In *CVPR*, pages 8623–8632, 2020. 1, 2, 3, 4, 5, 6

[35] Konstantin Sofiiuk, Ilia A Petrov, and Anton Konushin. Reviving iterative training with mask guidance for interactive segmentation. *arXiv preprint arXiv:2102.06583*, 2021. 1, 2, 3, 4, 5, 6

[36] Ke Sun, Yang Zhao, Borui Jiang, Tianheng Cheng, Bin Xiao, Dong Liu, Yadong Mu, Xinggang Wang, Wenyu Liu, and Jingdong Wang. High-resolution representations for labeling pixels and regions. *arXiv preprint arXiv:1904.04514*, 2019. 4

[37] 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. 1

[38] Ning Xu, Brian Price, Scott Cohen, Jimei Yang, and Thomas Huang. Deep grabcut for object selection. In *BMVC*, 2017. 1

[39] Ning Xu, Brian Price, Scott Cohen, Jimei Yang, and Thomas S. Huang. Deep interactive object selection. In *CVPR*, pages 373–381, 2016. 1, 2, 5

[40] Hongkai Yu, Youjie Zhou, Hui Qian, Min Xian, and Song Wang. Loosecut: Interactive image segmentation with loosely bounded boxes. In *ICIP*, pages 3335–3339, 2017. 1

[41] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In *ECCV*, pages 173–190, 2020. 4

[42] Shiyin Zhang, Jun Hao Liew, Yunchao Wei, Shikui Wei, and Yao Zhao. Interactive object segmentation with inside-outside guidance. In *CVPR*, pages 12234–12244, 2020. 1