

# EdgeFlow: Achieving Practical Interactive Segmentation with Edge-Guided Flow

Yuying Hao<sup>1</sup>, Yi Liu<sup>1</sup>, Zewu Wu<sup>1</sup>, Lin Han<sup>2\*</sup>, Yizhou Chen<sup>3\*</sup>, Guowei Chen<sup>1</sup>  
Lutao Chu<sup>1</sup>, Shiyu Tang<sup>1</sup>, Zhiliang Yu<sup>1</sup>, Zeyu Chen<sup>1</sup>, Baohua Lai<sup>1</sup>  
<sup>1</sup>Baidu, Inc.    <sup>2</sup>NYU    <sup>3</sup>CQJTU

## Abstract

High-quality training data play a key role in image segmentation tasks. Usually, pixel-level annotations are expensive, laborious and time-consuming for the large volume of training data. To reduce labelling cost and improve segmentation quality, interactive segmentation methods have been proposed, which provide the result with just a few clicks. However, their performance does not meet the requirements of practical segmentation tasks in terms of speed and accuracy. In this work, we propose EdgeFlow, a novel architecture that fully utilizes interactive information of user clicks with edge-guided flow. Our method achieves state-of-the-art performance without any post-processing or iterative optimization scheme. Comprehensive experiments on benchmarks also demonstrate the superiority of our method. In addition, with the proposed method, we develop an efficient interactive segmentation tool for practical data annotation tasks. The source code and tool is available at <https://github.com/PaddlePaddle/PaddleSeg>.

## 1. Introduction

Deep learning has seen tremendous success in computer vision areas, such as image recognition [7, 23], object detection [13, 22] and image segmentation [12, 32, 30]. In order to learn powerful abstraction, large volumes of labelled image data are usually essential for the model training process. As the amount of data increasing, the cost of manual annotation grows rapidly, especially when it comes to pixel-level segmentation tasks. Although semi-supervised or even unsupervised algorithms have been proposed to relieve label dependence, there is a great gap in accuracy between them and full supervision.

Therefore, interactive segmentation appears to be an attractive and efficient way, which allows the human annotators to quickly extract the object-of-interest [52]. Unlike model-centric methods, interactive segmentation methods

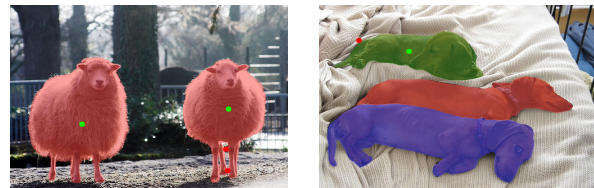


Figure 1. Examples of interactive clicks. Green dots denote positive clicks, red dots denote negative clicks. Best viewed in colors.

take interactive information into account. Therefore, they simplify the annotation process and improve the quality progressively. In general, interactive information could be various inputs, such as scribbles [4], clicks [47, 29, 46], bounding boxes [50] and so on.

The characteristics of the different interactions have been studied well by previous works, where click-based methods are the most promising, because they provide sufficient selected object information with minimal interaction time. In practice, click-based methods usually use two types of user clicks, i.e. positive clicks and negative clicks. Positive clicks aim to emphasize the target object (foreground) and negative clicks isolate non-target areas (background). Usually, such methods only need a few clicks to complete an object segmentation task, which is shown in Fig. 1.

In recent years, there are a couple of works [24, 21, 46] on click-based interactive segmentation, in which deep learning methods surpass traditional ones in terms of accuracy. However, most of them require extra post-processing in the evaluation process, which is time-consuming in practice. More recently, end-to-end interactive algorithms [29, 47] were proposed to speed up the clicking interaction, but they have common problems. The clicks are the only input to the first layers, so that the specific spatial and semantic information would be diluted through early layers. The other problem is that the relation of consecutive clicks is not modelling properly, resulting in unstable annotations, e.g. the segmentation annotation dramatically changes between two consecutive clicks.

In this work, we propose a novel interactive segmentation architecture that fully utilizes clicks of users and the

\*PaddlePaddle Developers Experts (PPDE)

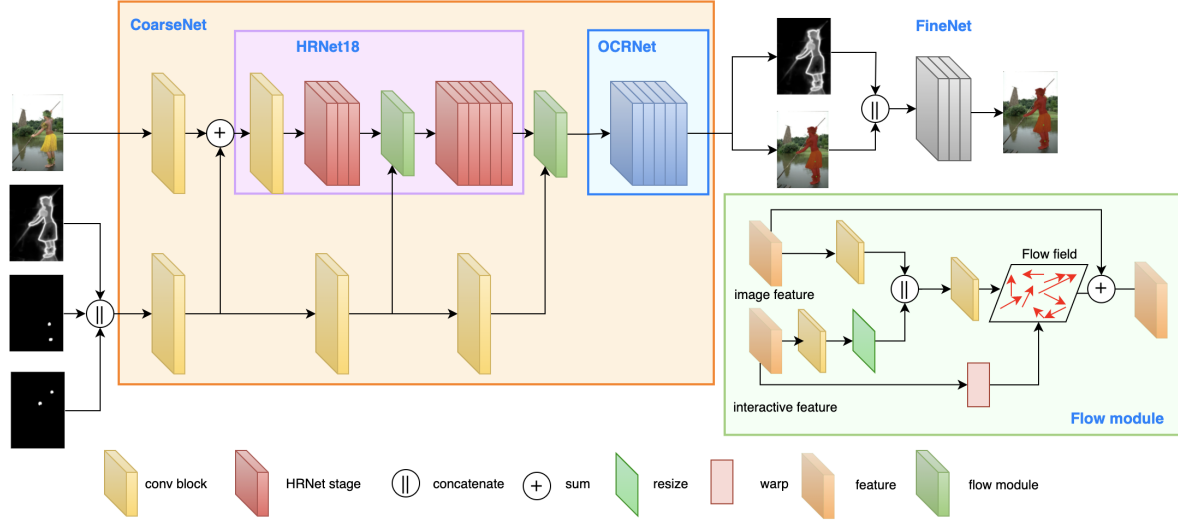


Figure 2. Overview of the EdgeFlow architecture. We design a coarse-to-fine network including CoarseNet and FineNet. For CoarseNet, We utilize HRNet-18+OCR as the base segmentation model and append the edge-guided flow to deal with interactive information. For FineNet, we utilize three atrous convolution blocks to refine the coarse masks.

relation of consecutive clicks. To enhance the interactive information, the feature of user clicks are embedded into both early and late layers and image features are efficiently integrated with an early-late fusion strategy. To establish the relationship between two consecutive clicks, the edge mask generated by the previous clicks is taken as an input together with the current click. It improves the stability of segmentation results significantly. In addition, we adopt a coarse-to-fine network design to further obtain the fine-grained segmentation. The comprehensive evaluations show our state-of-the-art performance on well-known benchmarks.

Furthermore, upon the proposed interactive model, we develop an efficient interactive segmentation tool for practical segmentation tasks, e.g. image labelling. The tool not only generates the segmentation mask, but also allows the user to adjust the polygon vertexes of the mask to further improve accuracy. Thus, the tool provides flexible options for annotation accuracy according to different practical tasks.

Our contributions are summarized as follow:

- We propose a novel interactive architecture that fully utilizes interactive and image information with the early-late fusion. The enhancement of interactive clicks prevents feature dilution over the network and then enables it to respond to the clicks efficiently.
- We utilize the object edges produced by network to improve the segmentation stability. With the coarse-to-fine network design, comprehensive experiments show our method achieves state-of-the-art performance on several benchmarks.
- We develop an efficient interactive segmentation tool

that supports interactive annotation and polygon frame editing. Our tool also supports multi-scenes and various labelling formats.

## 2. Related Works

The interactive segmentation task aims to obtain an accurate mask of an object with minimal user interaction. Interactive information can be clicks, scratches, contours, bounding boxes, phrases and so on. According to the type of information modelling, there are two research branches for interactive image segmentation algorithms.

**Optimization based methods:** Most of them are traditional methods which are divided into four categories: 1) contour based methods [39, 40, 11, 3, 25], 2) graph-cut based methods [9, 8, 31], 3) random walk based methods [15, 14, 5] and 4) region merging methods [2, 44, 43]. As a contour-based method, the active contours model (ACM) [41] constructs and optimizes an energy equation until the force on the closed curve decreases to zero. Graph-cut based methods (GC) [9, 8] utilize min-cut/max-flow algorithm to minimize the energy function. Random walk based methods (RW) [15, 14] construct an undirected graph by taking pixels as vertices and the relationship of neighbourhoods as edges. Region merging methods (RM) are initialized by user-interactive seeds and then gather the similar points and regions by homogeneity criterion.

The traditional methods have common drawbacks. The generalization ability of the methods is poor, where they only work well on specific scenes. In addition, they are sensitive to initial interactive information and require high-quality interactions without noisy input.

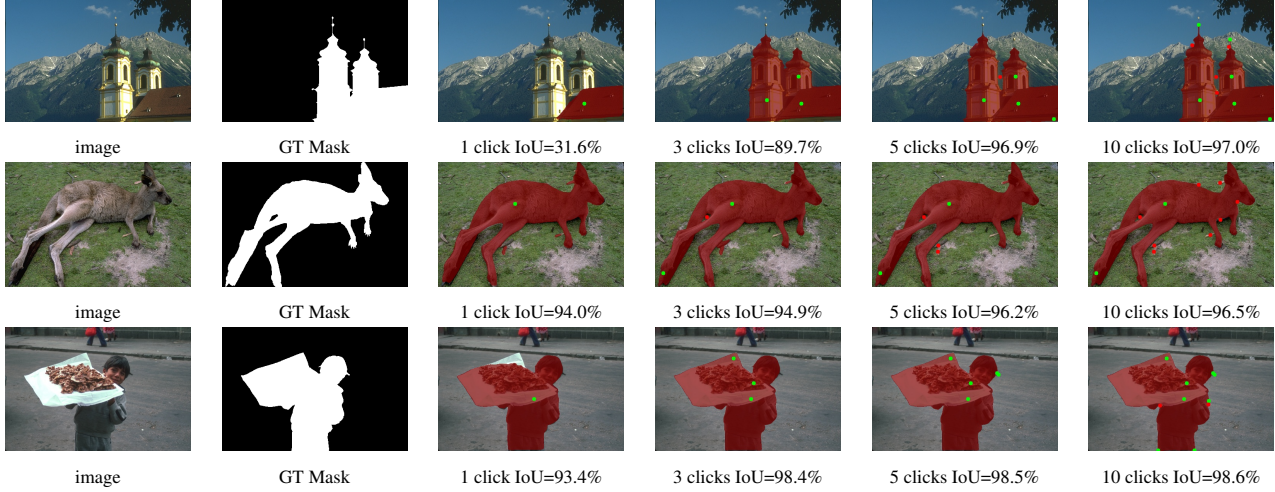


Figure 3. Qualitative results of our method on Berkeley dataset. The first two columns represent original images and ground-truth masks, respectively. Other columns show segmentation results with the different numbers of clicks. Green dots denote positive clicks, red dots denote negative clicks.

**Deep learning based methods:** Xu *et al.* [49] firstly introduced deep learning to solve interactive segmentation problems. They converted the interactive clicks to a distance map and then took the distance map together with the original image as an input to fine-tune FCN [32]. After that, Maninis *et al.* [37] took extreme points of an object as interactive information and utilized DeepLabv2 [12] as a segmentation model. Optimization for activation has been applied on the back-propagating refinement scheme (BRS) [21], which corrected mislabeled pixels by employing the L-BFGS algorithm. However, the optimization is time-consuming. Soon after, Feature-BRS [46] was proposed to improve the optimization scheme and speed up the interaction process. The mislabeled pixels would be rectified by auxiliary scales and biases which modify features from the middle of the network. Kontogianni *et al.* [24] adjusted the mask of the target object by optimizing the model parameters at test time. Sofiiuk *et al.* [47] applied an iterative training process by employing previous mask outputs, which improved the segmentation accuracy.

However, these methods highly rely on post-processes or additional optimization schemes in the evaluation process, which requires extra time and computations. Besides, they are sensitive to a new click, resulting in unstable segmentation masks.

### 3. Proposed Method

In this work, we propose a novel interactive segmentation architecture that fully utilizes clicks and the relation of consecutive clicks. Compared with previous approaches that combine image and clicks together at the first layers, our proposed method 1) utilizes a separate branch to enhance the features of clicks, which enables the network to

respond to interactive information better; 2) applies edge mask as prior information of the network, which stabilizes annotations.

#### 3.1. Network architecture

As shown in Fig. 2, the proposed architecture consists of two parts, input base segmentation model (CoarseNet) and refinement model (FineNet). For network inputs, the edge mask and the positive/negative clicks are encoded by the interactive branch.

**CoarseNet.** We utilize HRNet-18 [48] as the backbone and OCRNet [51] as segmentation head. The architecture has been proved to be excellent for semantic segmentation tasks. Unlike the single image branch in the generic segmentation model, we add an interaction branch into the backbone to handle the positive clicks, negative clicks, as well as the edge mask. In the beginning, the edge mask is initialized to a zero map with the same size as the input image. After receiving user clicks, the edge mask is estimated through the segmentation result and then is adopted as a part of input. To obtain an accurate edge mask, we add an auxiliary block for the edge constraint in the segmentation head. The fusion details of the image and interaction branch will be introduced in section 3.2.

**FineNet.** We utilize a FineNet module to further refine the coarse segmentation mask. The module takes three parts as the input, the outputs of CoarseNet, the original image and the user clicks, where the image and the user clicks are the same as partial inputs in CoarseNet. Note that for better visualization, we do not connect them to FineNet in Fig. 2, while they are connected in our implementation. To improve computation efficiency, we use three atrous convolution blocks with lightweight operations in the FineNet.

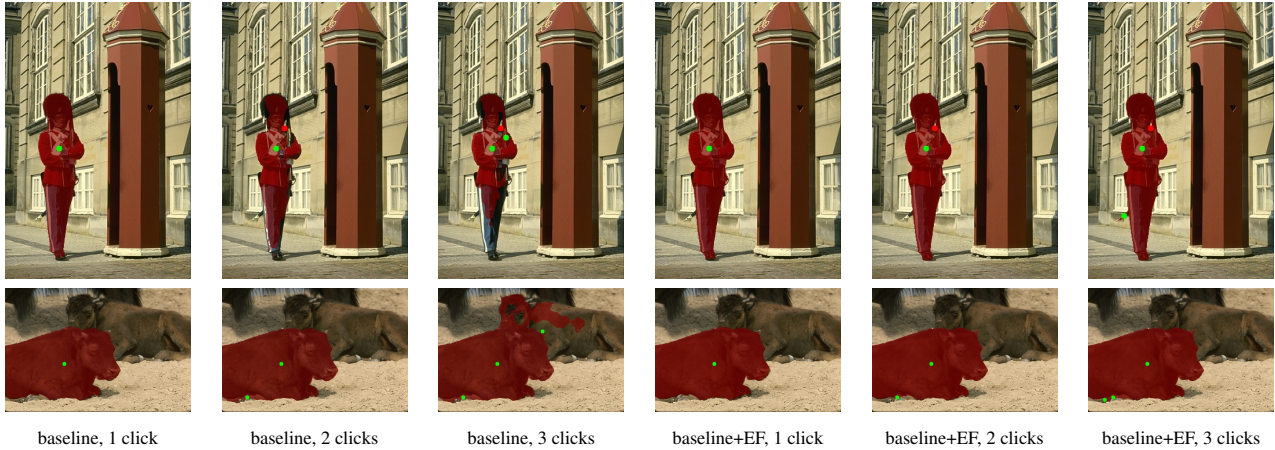


Figure 4. The qualitative results of edge-guided flow on Berkeley dataset. The first three columns show the output of baseline on different clicks. The last three columns show the output of baseline with edge-guided flow on different clicks. Edge-guided flow produces more stable segmentation results.



Figure 5. The qualitative results of early-late fusion. We visualize the feature map before OCR decoder. The second and fifth column utilize model with early fusion strategy. The third and sixth column utilizes early-late fusion strategy. The architecture of edge-guided flow utilizes early-late fusion strategy and prevents the network forgetting interactive information.

The atrous convolutions also obtain large receptive fields on high-resolution features, so that they can obtain more context information and improve the quality of outputs.

### 3.2. Edge-guided Flow

**Clicks encoding:** Interactive information contains the coordinates of positive and negative clicks. The positive clicks emphasize the target objects and the negative clicks isolate abandoned areas. To achieve the intuition and feed to the network, the clicks need to be encoded as feature maps. Usually, interactive segmentation methods utilize Gaussian algorithm or  $\mathcal{L}_2$  distance to generate distance maps as the feature maps. However, such distance maps change dramatically when adding a new click. Benenson *et al.* [6] found that the disk map has more stable performance than other

encoding methods. Inspired by [6, 47], in our implementation, we utilize the disk map with a radius of 5 pixels to encode both positive and negative clicks.

**Early-late fusion:** In general, a segmentation model only supports three-channel images as the inputs. Thus, the generic model should be modified to adapt to the additional interaction inputs. Some methods [46, 37, 35] introduce additional convolution layers in front of the backbone to accept N-channels input, which is shown in Fig. 6. Both of them fuse interaction and image features before the backbone network, which is so-called early fusion. The early fusion methods have common problems in that interaction information is not extracted properly. The interaction features are much sparser than image features and contain high-level information, e.g. position information. The early layers in



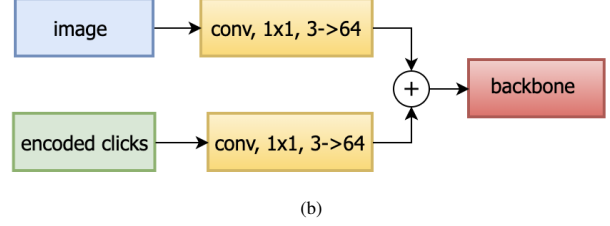
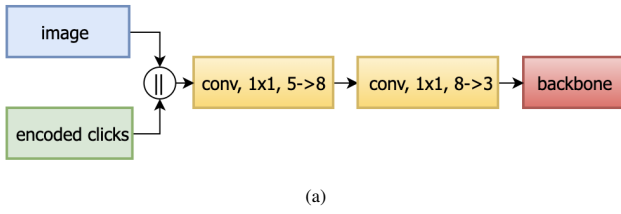


Figure 6. Different fusion blocks in front of backbone.

the backbone network are focusing on low-level feature extraction, so that the interaction features would be diluted through the early layers and the network can not respond to user clicks timely.

To prevent feature dilution, we propose an early-late fusion strategy to integrate interaction and image features. Instead of only fusing the features at the beginning of the network, we design the multi-stage feature fusion as shown in Fig. 2. The first fusion is the same as Fig. 6(b). The second fusion is between the first transition block and the second stage block of the backbone. The last fusion is after the fourth stage block. The multi-stage fusion promotes the propagation of interaction information over the network and also enables the network to respond to the user clicks precisely.

**Edge Guidance:** The key idea of interactive segmentation is improving segmentation masks progressively with user clicks. Due to the large spatial variance of the user clicks, the features of consecutive clicks would be quite different, resulting in dramatic segmentation masks. Previous methods [47] introduced the segmentation mask of previous clicks as an input, which alleviates this problem to a certain extent. However, the full mask could make the model fall into local optima, e.g. a poor previous mask usually leads to poor segmentation results. To improve the stability of the segmentation mask, we propose an edge mask scheme, which takes the object edges estimated from the previous iteration as prior information, instead of direct mask estimation. Edge estimation is sparser and less fluctuating than the full mask on input, so it can improve the stability and efficiency of segmentation.

In the interactive segmentation model, the interaction image and edge mask features are heterogeneous, resulting in a large spatial bias. Thus, it is necessary to align these properly. The optical flow method is originally used to align features from two adjacent frames in a video [33]. In semantic segmentation, it is effective for multi-scale features alignment while fusing different layers. Inspired by optical flow, we take a flow module to align image and interaction features such that spatial information can be represented precisely. The details of the flow module are shown in Fig. 2.

### 3.3. Loss Function

We expect that the loss function is more focusing on the wrong pixels rather than well-classified pixels. Inspired by [47], we utilize normalize focal loss to calculate the discrepancy between prediction mask and ground truth mask. It can be denoted as Eq 1:

$$\mathcal{L}_m(i, j) = -\frac{1}{\sum_{i,j} (1 - p_{i,j})^\gamma} (1 - p_{i,j})^\gamma \log p_{i,j}, \quad (1)$$

where  $\gamma$  is the hyper-parameter for focal loss.  $p_{i,j}$  denotes the confidence of the prediction of pixel  $(i, j)$ , it is denoted as :

$$p_{i,j} = \begin{cases} p, & y = 1 \\ 1 - p, & otherwise \end{cases}, \quad (2)$$

where  $p$  is the prediction probability on the location  $(i, j)$ ,  $y$  is the corresponding ground truth on the location  $(i, j)$ .

To minimize the difference between edge estimation from the last iteration and edges derived from ground truth masks, we employ balanced BCE loss  $\mathcal{L}_e$  to assign more attention on edge than the background. Besides, we utilize BCE as auxiliary loss  $\mathcal{L}_a$  to constrain the backbone outputs.

## 4. Experiments

### 4.1. Experiment Settings

**Datasets:** Following [47], we use the combination of COCO [28] and LVIS [20] as the training data. LVIS contains 164k images and 2M high-quality instance masks with over 1000 classes. Due to the long-tailed object classes from LVIS dataset, following [47], we use 10582 COCO images and corresponding 25832 instance masks as the augmented dataset.

We evaluate our method on four popular benchmarks, including GrabCut [45], Berkeley [38], Pascal VOC [16], DAVIS [42]. The GrabCut dataset contains 50 images and 50 masks. The Berkeley dataset contains 96 images and 100 masks. For Pascal VOC dataset, we only use the validation set, which contains 1449 images and 3417 instances.

Table 1. Evaluation results of GrabCut, Berkeley, DAVIS and Pascal VOC. Lower value is better. The best result is marked in bold.

Method	GrabCut		Berkeley	DAVIS		Pascal VOC
	NoC@85	NoC@90	NoC@90	NoC@85	NoC@90	NoC@85
GC [10]	7.98	10.00	14.22	15.13	17.41	-
GM [18]	13.32	14.57	15.96	18.59	19.50	-
RW [17]	11.36	13.77	14.02	16.71	18.31	-
ESC [19]	7.24	9.20	12.11	15.41	17.70	-
GSC [19]	7.10	9.12	12.57	15.35	17.52	-
DOS [49]	-	6.04	8.65	-	-	6.88
LD [26]	3.20	4.79	-	5.05	9.57	-
RIS-Net [27]	-	5.00	6.03	-	-	5.12
ITIS [35]	-	5.60	-	-	-	3.80
CAG [36]	-	3.58	5.60	-	-	3.62
BRS [21]	2.60	3.60	5.08	5.58	8.24	-
FCA [29]	-	2.08	3.92	-	7.57	2.69
IA+SA [24]	-	3.07	4.94	5.16	-	3.18
f-BRS-B [46]	2.50	2.98	4.34	5.39	7.81	-
RITM-H18 [47]	<b>1.54</b>	<b>1.70</b>	2.48	4.79	6.00	2.59
Our method	1.60	1.72	<b>2.40</b>	<b>4.54</b>	<b>5.77</b>	<b>2.50</b>

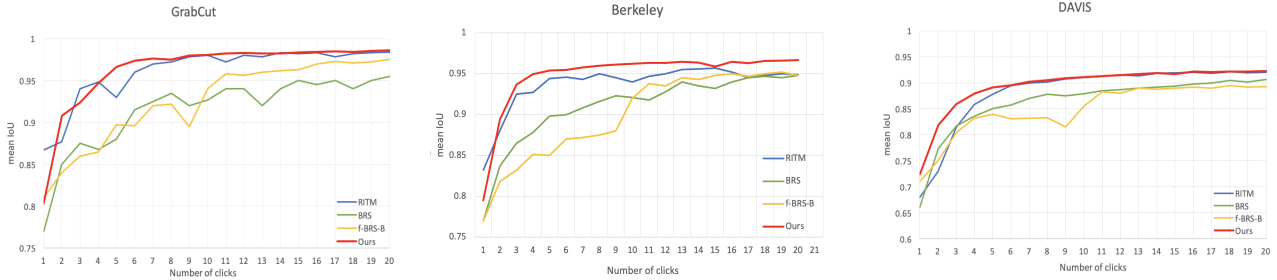


Figure 7. Evaluation of different interactive segmentation methods. The plots represent the mean IoU of different methods for the first 20 clicks iteratively on GrabCut, Berkeley and DAVIS.

The DAVIS dataset is randomly sampled from the video object segmentation dataset. We adopt 345 images and corresponding 345 masks which were introduced in [21, 47].

**Evaluation metrics:** Mean intersection over union (IoU) and the standard number of clicks (NoC) are used as evaluation metrics. NoC indicates the number of interactive clicks for achieving over the specified IoU threshold, which is commonly set as 85% and 90% [47, 37, 46, 21], i.e. NoC@85 and NoC@90. During the evaluation procedure, we build a simulation to generate clicks, following [46, 21, 47]. The first click is positive to the object of interest. The next click is selected as the centre of the largest error region. Note that in this work, the maximum number of clicks is 20, which is the same as [46, 21, 47].

**Implementation details:** In this work, we utilize COCO+LVIS dataset to train our model. For data augmentation, we adopt random resize with a scale from 0.75 to 1.4, horizontal flip, random crop image with the resize of (320, 480). For colour distortion, we utilize random distribution on contrast, brightness and pixel value for the original image.

In our experiments, the weight of focal loss  $\mathcal{L}_m$ , balanced BCE loss  $\mathcal{L}_e$  and BCE loss  $\mathcal{L}_a$  is set as 1, 0.4, 0.4, respectively. Firstly, we train the CoarseNet for 40 epochs with the learning rate of  $5 \times 10^{-4}$ . Then, we train the whole network for the last 30 epochs, where the learning rate of  $5 \times 10^{-4}$  for FineNet and  $5 \times 10^{-5}$  for CoarseNet. We utilize Adam optimizer and polynorm decay to reduce the learning rate to 1 percent of the initial learning rate value. Our model is based on PaddlePaddle [34, 1]. We train the model on 2 GPUs (Tesla V100), and the batch size is set to 32.

## 4.2. Comparison with SOTA methods

**NoC@k.** We calculate NoC@85 and NoC@90 as the metrics on GrabCut, Berkeley, DAVIS and Pascal VOC. The evaluation results are shown in table 1. Our method achieves the best performance on all datasets except for GrabCut. GrabCut has a limited number of objects with clear boundaries, so that all deep learning methods perform very well, and our method also achieves competitive results. Whereas Berkeley images have complicated object bound-

Table 2. Evaluation of different model architecture on Berkeley, DAVIS and Pascal VOC. ‘baseline’ represents coarse model without edge-guided flow branch. ‘EF’ represents edge-guided flow. ‘F’ represents FineNet. Lower value means less interaction and better performance.

model	Berkeley		DAVIS		Pascal VOC	
	NoC@85	NoC@90	NoC@85	NoC@90	NoC@85	NoC@90
baseline	2.06	2.93	4.83	6.27	2.68	3.26
baseline+EF	1.83	2.79	4.82	6.14	2.66	3.19
baseline+EF+F	<b>1.60</b>	<b>2.40</b>	<b>4.54</b>	<b>5.77</b>	<b>2.50</b>	<b>3.07</b>

Table 3. Evaluation of different input setting on Berkeley, DAVIS and Pascal VOC. ‘SI’ represents sobel edge as prior information. ‘EI’ represents edge mask from segmentation result. ‘F’ represents FineNet. Lower value means less interaction and better performance.

Model	Berkeley		DAVIS		Pascal VOC	
	NoC@85	NoC@90	NoC@85	NoC@90	NoC@85	NoC@90
SI + w/o F	1.83	3.20	5.04	6.32	2.88	3.41
EI + w/o F	1.83	2.79	4.82	6.14	2.66	3.19
SI + F	<b>1.58</b>	2.73	4.68	6.21	2.9	3.46
EI + F	1.60	<b>2.40</b>	<b>4.54</b>	<b>5.77</b>	<b>2.5</b>	<b>3.07</b>

aries, e.g. bicycle wheel springs and parachute ropes. In this dataset, our method achieves the best performance compared with other methods. Since our method utilizes edge masks as prior information, it provides more accurate details of segmentation objects.

For DAVIS and Pascal VOC datasets, they contain more than 1.5k images with various scenes, which are more suitable for practical tasks. Our method achieves the best results compared with other methods. Analysing the result, we consider that early-late fusion strategy can prevent features of interactive clicks dilution over the network. Therefore, our method responds to the clicks efficiently and is robust in various scenes.

**mIoU per click.** Fig. 7 shows the mIoU change over the first 20 clicks on different datasets. There are three observations: 1) Our method has the most stable performance compared with other methods, where mIoU improves gradually as the clicks increases. However, other methods have the degradation problem. For example, mIoU of RITM decreases dramatically when the clicks increase from 4 to 5, and 10 to 11 on GrabCut dataset. As described above, the proposed edge estimation can improve the segmentation stability. 2) Our method requires fewer clicks to achieve higher mIoU compared with other methods. In the Berkeley dataset, NoC@95 of our method is around 4, while others are more than 10. 3) Using all 20 clicks, our method still achieves the highest mIoU on all datasets. The experiment demonstrates the superior segmentation ability of our method.

### 4.3. Ablation Study

**Edge Flow.** We perform ablation experiments on Berkeley, DAVIS and Pascal VOC datasets to validate the effectiveness of the edge-guided flow. As shown in table 2, the baseline with edge-guided flow outperforms the baseline obviously on all three datasets. For a better understanding, we visualize the intermediate feature maps of baseline and

baseline+EF, as shown in Fig. 5. The model with the early-late fusion shows the stronger activation on coordinate information, while the early fusion has the feature dilution problem. Then, we visualize the output difference between baseline and baseline+EF, which is shown in Fig. 4. The result demonstrates that edge-guided flow improves interactive segmentation performance and stabilize the output when adding a new click. We apply edge masks as prior information of the network, which makes the segmentation results more stable.

**FineNet.** As shown in table 2, we find that NoC@85 and NoC@90 improve significantly after FineNet is added. As described above, FineNet adopts atrous convolution blocks which obtain large receptive fields on high-resolution features, so that they can obtain more context information. The best result is the combination of baseline+EF+F, which demonstrates the effectiveness of the early-late fusion strategy and the coarse-to-fine architecture.

**Prior information** Table 3 shows the effect of prior information on different datasets. Fig 8 shows the mIoU change over the first 20 clicks using different prior information. We study two types of edge masks, 1) the edge mask estimated from the previous segmentation result, i.e. EI mask, 2) the edge mask directly calculated by the Sobel operator, i.e. SI mask. We find the model with the EI mask outperforms the other one. In the segmentation model, prior information can be dynamically adjusted with the network and user interaction clicks. Therefore, the EI mask is more related to the object of interest. However, the SI mask calculation is only based on the original image, where it contains a lot of redundant information which is irrelevant to the object of interest. Therefore, it causes network performance degradation.

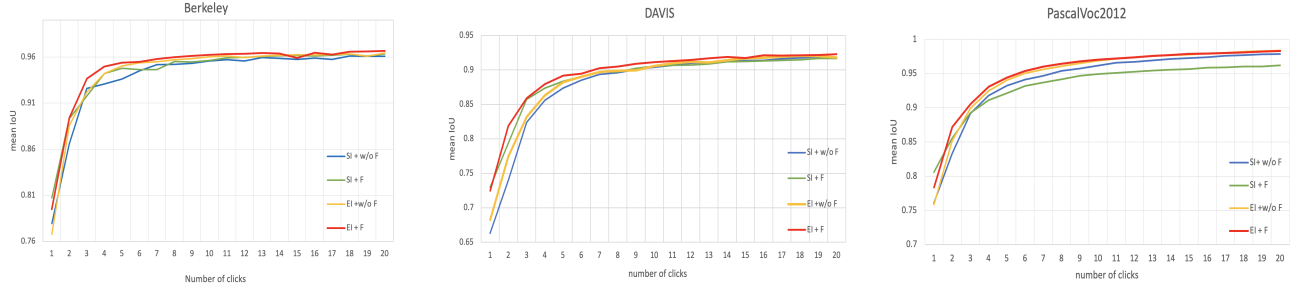


Figure 8. Effects of prior information. The plot represents the value of mean Intersection over Union for first 20 clicks iteratively on Berkeley, DAVIS and Pascal VOC, respectively.



Figure 9. Visualization of interactive segmentation tool.

## 5. Interactive Segmentation Tool

In this work, we develop an interactive segmentation tool using the proposed model. Our tool aims to help users annotate segmentation datasets efficiently and accurately. The annotation pipeline includes three steps: data preparation, interactive annotation and polygon frame editing.

During data preparation, our tool supports data formats in multiple domains including natural images, medical imaging, remote sensing images and so on. After the image is loaded into the software, it can be zoomed, moved and pre-processed by adjusting brightness and contrast. The annotation process is highly flexible and configurable. Most operations in the application support keyboard shortcuts which can be changed by the user.

During interactive annotation, users add positive and negative points with left and right mouse clicks, respectively. The application runs model inference and shows the user prediction result, as shown in Fig. 9(a). The user can adjust the target border by changing the threshold to distinguish foreground and background pixels to get more accurate segmentation results. The tool also supports filtering the largest connected region, which is shown in Fig. 9(b). This feature is useful when there are multiple targets of the same type in an image. Suppressing small positive regions can free users from clicking negative points in each of these regions.

After finishing interactive segmentation, the tool generates a polygon frame around the target border. Users can adjust the polygon vertexes to further improve segmentation accuracy. It is flexible for practical segmentation tasks.

In some cases, adjusting the polygon frame could be faster than adding many clicks during interactive segmentation, so that it improves the overall annotation efficiency. Finally, the segmentation results can be saved in multiple formats including segmentation mask, PASCAL VOC, COCO, pseudo colour and so on.

## 6. Conclusion

A large number of labelled image data are usually essential for segmentation models. Due to the high cost of the pixel-level annotations, interactive segmentation becomes an efficient way to extract the object of interest. In this work, we propose a novel interactive architecture named EdgeFlow that fully utilizes the user interaction information without any post-processing or iterative optimization scheme. With the coarse-to-fine network design, our proposed method achieves state-of-the-art performance on common benchmarks. Furthermore, we develop an efficient interactive segmentation tool that helps the user to improve the segmentation result progressively with flexible options. In future work, we will work on lightweight models that can be deployed into various platforms. Another promising topic we are working on is multi-modality, which utilizes more input types, like audio and text. The different kinds of inputs can complement each other, so that it is vital to improve the quality further.

## 7. Acknowledgements

This work was supported by the National Key Research and Development Project of China (2020AAA0103500).



## References

- [1] <https://github.com/PaddlePaddle/Paddle>.
- [2] Rolf Adams and Leanne Bischof. Seeded region growing. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(6):641–647, 1994.
- [3] Haider Ali, Lavdie Rada, and Noor Badshah. Image segmentation for intensity inhomogeneity in presence of high noise. *IEEE Trans. Image Process.*, 27(8):3729–3738, 2018.
- [4] Junjie Bai and Xiaodong Wu. Error-tolerant scribbles based interactive image segmentation, 2014.
- [5] Christos George Bampis and Petros Maragos. Unifying the random walker algorithm and the SIR model for graph clustering and image segmentation, 2015.
- [6] Rodrigo Benenson, Stefan Popov, and Vittorio Ferrari. Large-scale interactive object segmentation with human annotators, 2019.
- [7] Pengfei Bi and Xue Du. Application of locally invariant robust PCA for underwater image recognition. *IEEE Access*, 9:29470–29481, 2021.
- [8] Yuri Boykov and Gareth Funka-Lea. Graph cuts and efficient N-D image segmentation. *Int. J. Comput. Vis.*, 70(2):109–131, 2006.
- [9] Yuri Boykov and Marie-Pierre Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images, 2001.
- [10] Yuri Boykov and Marie-Pierre Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images, 2001.
- [11] Xavier Bresson, Selim Esedoglu, Pierre Vanderghenst, Jean-Philippe Thiran, and Stanley J. Osher. Fast global minimization of the active contour/snake model. *J. Math. Imaging Vis.*, 28(2):151–167, 2007.
- [12] 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. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(4):834–848, 2018.
- [13] Zhiyong Dai, Jianjun Yi, Lihui Jiang, Shikun Yang, and Xiaoci Huang. Cascade centernet: Robust object detection for power line surveillance. *IEEE Access*, 9:60244–60257, 2021.
- [14] Xingping Dong, Jianbing Shen, Ling Shao, and Luc Van Gool. Sub-markov random walk for image segmentation. *IEEE Trans. Image Process.*, 25(2):516–527, 2016.
- [15] Aurélien Ducournau and Alain Bretto. Random walks in directed hypergraphs and application to semi-supervised image segmentation. *Comput. Vis. Image Underst.*, 120:91–102, 2014.
- [16] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The pascal visual object classes (VOC) challenge. *Int. J. Comput. Vis.*, 88(2):303–338, 2010.
- [17] Leo Grady. Random walks for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(11):1768–1783, 2006.
- [18] Varun Gulshan, Carsten Rother, Antonio Criminisi, Andrew Blake, and Andrew Zisserman. Geodesic star convexity for interactive image segmentation, 2010.
- [19] Varun Gulshan, Carsten Rother, Antonio Criminisi, Andrew Blake, and Andrew Zisserman. Geodesic star convexity for interactive image segmentation, 2010.
- [20] Agrim Gupta, Piotr Dollár, and Ross B. Girshick. LVIS: A dataset for large vocabulary instance segmentation, 2019.
- [21] Won-Dong Jang and Chang-Su Kim. Interactive image segmentation via backpropagating refinement scheme, 2019.
- [22] Ren Jin, Junning Lv, Bin Li, Jianchuan Ye, and Defu Lin. Toward efficient object detection in aerial images using extreme scale metric learning. *IEEE Access*, 9:56214–56227, 2021.
- [23] Taehung Kim, Hoseong Kim, and Hyeran Byun. Localization-aware adaptive pairwise margin loss for fine-grained image recognition. *IEEE Access*, 9:8786–8796, 2021.
- [24] Theodora Kontogianni, Michael Gygli, Jasper R. R. Uijlings, and Vittorio Ferrari. Continuous adaptation for interactive object segmentation by learning from corrections, 2020.
- [25] Hao Li, Maoguo Gong, Qiguang Miao, and Bin Wang. Interactive active contour with kernel descriptor. *Inf. Sci.*, 450:53–72, 2018.
- [26] Zhuwen Li, Qifeng Chen, and Vladlen Koltun. Interactive image segmentation with latent diversity, 2018.
- [27] Jun Hao Liew, Yunchao Wei, Wei Xiong, Sim Heng Ong, and Jiashi Feng. Regional interactive image segmentation networks, 2017.
- [28] 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, 2014.
- [29] Zheng Lin, Zhao Zhang, Lin-Zhuo Chen, Ming-Ming Cheng, and Shao-Ping Lu. Interactive image segmentation with first click attention, 2020.
- [30] Yi Liu, Lutao Chu, Guowei Chen, Zewu Wu, Zeyu Chen, Baohua Lai, and Yuying Hao. Paddleseg: A high-efficient development toolkit for image segmentation. *CoRR*, abs/2101.06175, 2021.
- [31] Ekaterina Lobacheva, Olga Veksler, and Yuri Boykov. Joint optimization of segmentation and color clustering, 2015.
- [32] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation, 2015.
- [33] Tongwei Lu, Shihui Ai, Yongyuan Jiang, Yudian Xiong, and Feng Min. Deep optical flow feature fusion based on 3d convolutional networks for video action recognition, 2018.
- [34] Yanjun Ma, Dianhai Yu, Tian Wu, and Haifeng Wang. Paddlepaddle: An open-source deep learning platform from industrial practice. 1(1):105–115, 2019.
- [35] Sabarinath Mahadevan, Paul Voigtlaender, and Bastian Leibe. Iteratively trained interactive segmentation, 2018.
- [36] Soumajit Majumder and Angela Yao. Content-aware multi-level guidance for interactive instance segmentation, 2019.
- [37] Kevis-Kokitsi Maninis, Sergi Caelles, Jordi Pont-Tuset, and Luc Van Gool. Deep extreme cut: From extreme points to object segmentation, 2018.
- [38] David R. Martin, Charless C. Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, 2001.
- [39] Paulo A. V. Miranda, Alexandre X. Falcão, and Thiago Vallin Spina. Riverbed: A novel user-steered image segmentation method based on optimum boundary tracking. *IEEE Trans. Image Process.*, 21(6):3042–3052, 2012.
- [40] Thi Nhat Anh Nguyen, Jianfei Cai, Juyong Zhang, and

- Jianmin Zheng. Robust interactive image segmentation using convex active contours. *IEEE Trans. Image Process.*, 21(8):3734–3743, 2012.
- [41] Thi Nhat Anh Nguyen, Jianfei Cai, Juyong Zhang, and Jianmin Zheng. Robust interactive image segmentation using convex active contours. *IEEE Trans. Image Process.*, 21(8):3734–3743, 2012.
  - [42] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus H. Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation, 2016.
  - [43] Hongwu Qin, Jasni Mohamad Zain, Xiuqin Ma, and Tao Hai. Scene segmentation based on seeded region growing for foreground detection, 2010.
  - [44] Muhamad Farid Ramli and Khairul Nizam Tahar. Homogeneous tree height derivation from tree crown delineation using seeded region growing (SRG) segmentation. *Geo spatial Inf. Sci.*, 23(3):195–208, 2020.
  - [45] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "grabcut": interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, 2004.
  - [46] Konstantin Sofiiuk, Ilia A. Petrov, Olga Barinova, and Anton Konushin. F-BRS: rethinking backpropagating refinement for interactive segmentation, 2020.
  - [47] Konstantin Sofiiuk, Ilia A. Petrov, and Anton Konushin. Reviving iterative training with mask guidance for interactive segmentation. *CoRR*, abs/2102.06583, 2021.
  - [48] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation, 2019.
  - [49] Ning Xu, Brian L. Price, Scott Cohen, Jimei Yang, and Thomas S. Huang. Deep interactive object selection, 2016.
  - [50] Ning Xu, Brian L. Price, Scott Cohen, Jimei Yang, and Thomas S. Huang. Deep grabcut for object selection, 2017.
  - [51] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation, 2020.
  - [52] Shiyin Zhang, Jun Hao Liew, Yunchao Wei, Shikui Wei, and Yao Zhao. Interactive object segmentation with inside-outside guidance, 2020.