FocusCut: Diving into a Focus View in Interactive Segmentation

Zheng Lin[1] Zheng-Peng Duan[1] Zhao Zhang[1,2] Chun-Le Guo[1*] Ming-Ming Cheng[1]
[1]TMCC, College of Computer Science, Nankai University [2]SenseTime Research
http://mmcheng.net/focuscut/

Abstract

Interactive image segmentation is an essential tool in pixel-level annotation and image editing. To obtain a high-precision binary segmentation mask, users tend to add interaction clicks around the object details, such as edges and holes, for efficient refinement. Current methods regard these repair clicks as the guidance to jointly determine the global prediction. However, the global view makes the model lose focus from later clicks, and is not in line with user intentions. In this paper, we dive into the view of clicks' eyes to endow them with the decisive role in object details again. To verify the necessity of focus view, we design a simple yet effective pipeline, named FocusCut, which integrates the functions of object segmentation and local refinement. After obtaining the global prediction, it crops click-centered patches from the original image with adaptive scopes to refine the local predictions progressively. Without user perception and parameters increase, our method has achieved state-of-the-art results. Extensive experiments and visualized results demonstrate that FocusCut makes hyper-fine segmentation possible for interactive image segmentation.

1. Introduction

Interactive image segmentation aims to obtain an accurate binary mask of the target object with the least interaction cost. It has developed into an indispensable tool in serving pixel-level data annotation and image editing. The research mainly focuses on two aspects. One is a more efficient mode of user interaction, and the other is to make more efficient use of the interaction provided by users. For the former, the interactive modes are widely explored and mainly based on the bounding box [50], the polygon [1, 6, 32], clicks [2, 29, 36], scribbles [3, 48], and some combinations [34, 52]. Among them, the click-based method has become the mainstream because of its simplicity. For the latter, researchers have explored the interaction ambiguity [9, 26, 30], input information [31, 35], backpropagating [20, 41], *etc*. These methods provide better segmentation results without changing the user input.

*C.L. Guo is the corresponding author.

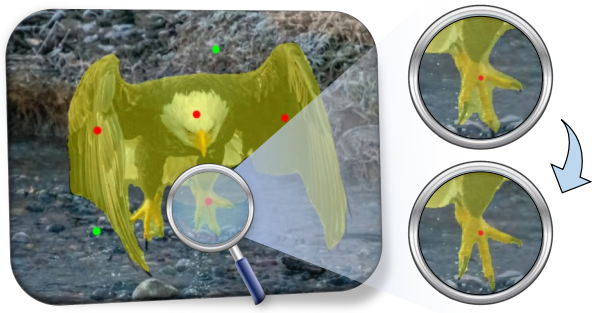

Figure 1. Visual display of FocusCut. The details of the eagle claw are refined with an additional focus view. The red and green clicks are provided by users to indicate the foreground and background in interactive segmentation. The yellow mask is the prediction.

In recent years, with the increase of large screen equipments and the improvement of aesthetic, both image annotation and image editing need more delicate segmentation masks. In high-precision interactive segmentation, the refinement of object details, such as edges and holes, often requires more interactive clicks and time. When users click in mislabeled regions, they tend to focus on the detail region for efficient repairing. However, current methods consider previous clicks together to determine the global prediction. In a new round of interaction, a join predicting process may weaken the decisive effect of the newly-input click on its surrounding details and feed back disagreeable results.

For more effective refinement, we dive into the view of a click to consider its surrounding information, which is called the focus view. In the paper, we design a concise pipeline, FocusCut, to verify the importance of the focus view. The original function of the interactive segmentation network has been changed, and instead, we endow it with a new role, allowing it to not only segment the target object but also repair local details. Specifically, after global segmentation, which is called the global view in our paper, it crops a local patch from original images centered by the newly-clicked point as the focus view to further refine object details using the same network. The progressive crop-

ping scopes are adjusted dynamically according to prediction variation in the global view. Then the crop scope will gradually decrease according to our progressive focus strategy. To keep it fair with other methods and better prove our opinion, almost no parameters and specific modules have been inserted into the common-used architecture of interactive segmentation. Comprehensive experiments have been carried out on GrabCut [40], Berkeley [37], SBD [15], and DAVIS [39] datasets to prove the effectiveness of FocusCut.

The contributions can be summarized as follows:

- We introduce the focus view to grasp user intentions by considering the local segmentation from clicks' eyes.
- Based on our opinion, we propose the FocusCut, a simple yet effective pipeline to strengthen the local refinement.
- Without additional parameters, the FocusCut achieves state-of-the-art performance, and the visualized results reflect its effectiveness in fine segmentation.

2. Related Work

2.1. Interactive Image Segmentation

Most traditional methods of interactive segmentation build models on the low-level features of the image, such as intelligent scissors [38] and lazy snapping [25]. On the basis of GraphCut [5], Rover *et al*. propose a method called GrabCut [40] to make it more convenient. Grady *et al*. develop the random walker [14] algorithm to determine the probability for each unlabeled pixel. Kim *et al*. [22] improve it by introducing a restart simulation. However, due to too much attention on the low-level features, these methods may become invalid in complex environments.

Thanks to the neural network's ability to comprehensively consider global and local features, although there are also some works [21, 46, 47] to further improve traditional methods, deep-learning-based methods have recently become the mainstream in this field. Except that some works are based on recurrent neural network [1, 6], graph convolutional network [32], and reinforcement learning [27, 42], most researches are conducted on the traditional convolutional neural network. Multiple interaction modes have been explored in this task. For example, the extreme points have been used in the segmentation for common objects [36], thin objecst [29], and the full image [2]. The boundary clicks [19, 24] are also adopted as an effective interaction. The combination of interactions, such as the bounding box and clicks [4, 52], is also popular in the field. Among these, the way of providing points in foreground and background has gradually become the mainstream, which is also the interaction mode studied in this paper.

For this interaction mode, Xu *et al*. [51] first propose a deep-learning-based algorithm, along with a click map transformation and several random sampling strategies. In order to make the user interactions into full use, Liew *et al*. [28] propose RIS-Net to exploit the local region from click pairs to refine the segmentation results. Hu *et al*. [17] provide a two-branch architecture for this task. Majumder *et al*. [35] improve the transformation of user clicks by generating content-aware guidance maps. Jang *et al*. [20] develop BRS to correct the mislabeled pixels in the initial results, which has been improved in f-BRS [41]. Kontogianni *et al*. [23] employ user corrections as training samples and update the model parameters instantly. To deal with the ambiguity of user interactions, Li *et al*. [26] couple two convolutional networks to train and select the proper result. Liew *et al*. [30] introduce scale diversity into the model to help users quickly locate their desired target. Lin *et al*. [31] emphasize the critical role of the first click and take it as the special guidance. Chen *et al*. [9] introduce a non-local method to fully exploit the user cues. Most methods transform user interactions into a guidance map sharing the same size as the whole image. However, we view each click extra in a focus view, utilizing them to the full potential.

2.2. Local View in Segmentation

Local information has been made full use of in many segmentation tasks. HAZN [49] can adaptively adjust the scale of view to the object or the part to refine the segmentation. GLNet [8] aggregates feature maps captured by local and global branches. Moreover, for semantic segmentation, AWMF-CNN [44] assigns weights to different magnification of local patches separately. CascadePSP [11] feeds image patches from the original image through the refinement module. Similarly, MagNet [18] refines segmentation results of local patches with different scales in a progressive way. However, for semantic segmentation tasks, sliding window strategy is inevitable, resulting in a large cost of calculation and time. Due to the specificity of interactive segmentation, the local views can be decided by the interaction, thus avoiding the shortcoming.

In interactive segmentation, RIS-Net [28] has proved the significance of local refinement. It generates the local patch by finding the nearest negative click for each positive click and constructs a bounding box. The local feature is extracted by using the ROI pooling layer from the main branch, whose input is the image concatenated with the transformed clicks. That is to say, the local refinement is still under the influence of the entire image and other clicks, weakening the dominant role of local clicks to some extent. Additionally, the local features are somewhat lost due to the down-sampling operation of the network. We go a step further and adopt a purer focus view for local refinement, directly feeding the local patches centered by each click into the network and completely ignoring the influence of the whole image and other distant clicks.

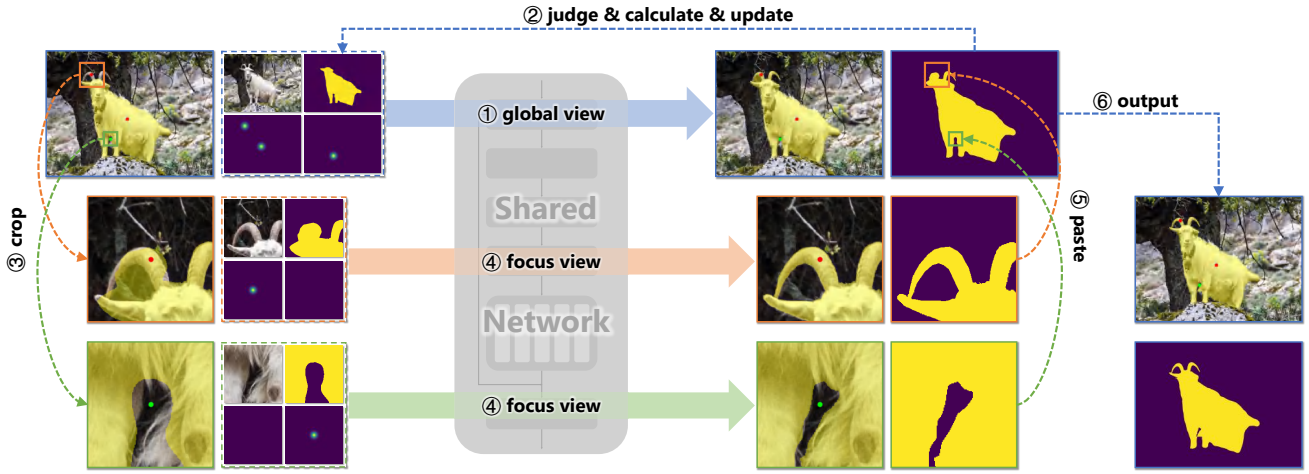

Figure 2. The pipeline of the FocusCut. The process is divided into six steps: (1) Feed the 6-channel input of the whole target into the shared network to generate the prediction in global view; (2) For current click, judge whether to focus and calculate the focus scope based on the variation between the current and previous predictions, then update the previous prediction with current one; (3) Crop the patches from the original image for each focus click with corresponding scopes respectively; (4) Feed the inputs of focus patches into the network to generate local predictions in focus view; (5) Paste the patch predictions to the global one; (6) Output the final prediction.

3. Proposed Method

3.1. Revisiting Classic Pipeline

With the development of neural networks, most of the works about interactive segmentation in recent years have been carried out by introducing the convolutional neural network. Since interactive segmentation can be regarded as a specific type of segmentation task, many methods are designed based on the classic network for semantic segmentation, DeepLab series, especially the DeepLab v3+ [7]. The network architecture contains a backbone network, an Atrous Spatial Pyramid Pooling (ASPP) part and a decoder part. For the backbone network, the ResNet [16] is mostly adopted in interactive segmentation. The ASPP part contains four dilated convolution branches and a global average pooling branch. The decoder part refines the ASPP module's output by fusing the backbone's low level features to generate the final prediction. For interactive segmentation, the input should contain the information of the interactions. The click locations will be transformed into two click maps, such as distance, disk, and our used Gaussian maps, representing the positive and negative points. Most works in interactive segmentation modify the input part of the network and take a 5-channel map as input, which concatenates the RGB image and two click maps. It can be implemented by adding another head to encode a 5-channel map to a 3-channel map for satisfying the standard architecture or directly changing the first convolutional layer like us. The output will be supervised by the ground truth with the binary cross-entropy loss and binarized to the final prediction.

3.2. FocusCut Pipeline

In the process of interactive segmentation, the user often repairs the incorrectly segmented region by providing more foreground and background points. As the number of clicks increases, the later points are used for repairing more local areas gradually. Especially in the later stage, it is likely that many interaction points are gathered together for repairing a small area. Due to the size of the receptive field and down-sampling operations of the neural network, it is difficult to segment the whole object and detail areas simultaneously.

As shown in Fig. 2, the provided FocusCut is a pipeline for interactive segmentation which contains two interactive views. One is the global view to segment the whole object, and the other is the focus view to refine the segmentation according to the previous coarse mask around clicks. To reflect the effectiveness of our method, we decide not to change the architecture of the common-used network as much as possible. We take the DeepLab v3+ with output stride of 16 as the basic network. The difference is that we regard this as a shared network, which can not only learn the segmentation of the whole object, but also learn the refinement for local areas. To achieve this, we need to unify the two inputs. Since the refinement in focus view is generated based on the coarse mask, we add an extra channel of the previous prediction for the input. We hope that our network can learn to generate a more accurate segmentation based on the previous prediction and interaction points besides object segmentation. To achieve this goal, we use the data of global view and focus view alternately to train our network. For the global view, we adopt the iterative train-

ing strategy [33]. The coarse prediction is set to the previous segmentation if it is the iterative step. For the other situations, it will be set to an empty map. The RGB image contains the whole object, and the clicks are also simulated according to the object mask, which will include at least one positive point to indicate the location of the object. In the global view, the network takes this 6-channel map as the input to generate the prediction of the whole object. For the focus view, the core of our method, we train the network with patch samples which represent the local information of the target. In Sec. 3.3, we will describe the process of generating patch samples in detail. As shown in Fig. 2, the input map is also a 6-channel map for this phase. However, the RGB images will be local areas cropped from the original image, which does not represent the object and will pay more attention to the fine details. Unlike the click maps in the global view, these click maps must contain the center point of the map, which may be either positive or negative. We will generate the coarse mask by processing the local ground truth to reduce its fineness. These maps will be concatenated and fed into the network.

Fig. 2 shows the inference phase in detail. In this phase, the user will click continually until the result meets the user's needs. Since the first click is bound to segment the whole object, we introduce our focus view from the second click and later. When the current click is added, the pipeline of global view will be firstly adopted, as shown in the top part of Fig. 2. According to the position of the current click and the difference between current prediction $\mathbf{P}$ and previous prediction $\mathbf{P}'$ in the global view, the judgment is made to determine whether the click should go through an additional path of focus view. If the focus view is adopted, we will calculate the focus scope r for the current click. This will be introduced in Sec. 3.4. Then the original image, clicks, and the current prediction will be cropped to a local patch according to the focus scope, which will be fed into the path of focus view to generate the local prediction $\hat{\mathbf{P}}$, as shown in the bottom part of Fig. 2. It is worth mentioning that the image patch here is cropped from the original image. For high-resolution images, this helps to avoid information missing and get a clearer RGB patch. Finally, the local prediction will be pasted back to the original prediction. If there are overlaps between patches, the overlapping part adopts their mean value. In Sec. 3.4, we also provide a progressive focus strategy to pay attention to local areas iteratively to achieve better results.

3.3. Focus Patch Simulation

In this section, we will introduce our simulation algorithm to generate the focus patches around the clicks for training. We find that in the middle and later stage of interactive segmentation, users often click around the object boundary to make the boundary more accurate, and the object details are often near the boundary. We generate the patch to simulate this situation. We select a point on the boundary of the object and give it a random offset based on β within $[\beta_{min}, \beta_{max}]$ to serve as the center point of the patch. The focus scope r is a random number related to the object size. The object size is reflected by k and calculated from the ground truth $\mathbf{G}$ and the random coefficient is α within $[\alpha_{min}, \alpha_{max}]$. The detailed calculation process is described in Algorithm 1. The default α_{min}, α_{max}, β_{min}, and β_{max} are 0.2, 0.8, -0.3, and 0.3 in our experiments.

With the patch center $\mathbf{p}$ and focus scope r, we crop the image and the corresponding ground truth as a square patch from $(\mathbf{p}_x - r, \mathbf{p}_y - r)$ to $(\mathbf{p}_x + r, \mathbf{p}_y + r)$. With the patch data, we generate a coarse mask as the previous prediction through dilating and eroding randomly as in [11]. The center point will always be included as a user click. We will also select $0 \sim 3$ positive and negative points in the patch to simulate these clicks around the center one. These patch clicks will be transformed into click maps and fed into the network with the RGB image and the coarse mask.

In Fig. 3, we illustrate simulated patches from an image of a chair and its ground truth. It can be seen that our algorithm simulates the user's interaction positions and crops different parts. At least one interaction point is included in

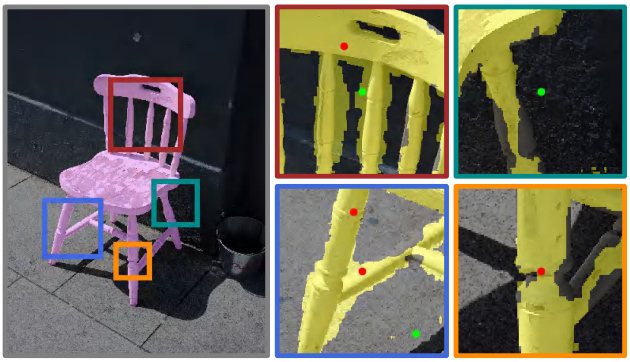

Figure 3. The example of focus patch simulation. The border colors of right patches represent the corresponding cropping in the left. The pink mask in the left represents the ground truth and the yellow ones in the right are the generated coarse masks. These simulated clicks are also shown on the patches.

Algorithm 1 Focus Patch Simulation

Input: The ground truth $\mathbf{G}$, constant $\alpha_{\texttt{min}}$, $\alpha_{\texttt{max}}$, $\beta_{\texttt{min}}$, $\beta_{\texttt{max}}$;

The object size $k = \sqrt{\sum_{i,j} \mathbf{G}_{i,j}}$, $\mathbf{G}_{i,j} \in \{0, 1\}$;
Select a random α from $[\alpha_{\texttt{min}}, \alpha_{\texttt{max}}]$;
The focus scope $r = \alpha \cdot k$;
Generate boundary map $\mathbf{B}$ from $\mathbf{G}$, $\mathbf{B}_{i,j} \in \{0, 1\}$;
Select a random boundary point $\tilde{\mathbf{p}}(x, y)$, $\mathbf{B}_{x,y} = 1$;
Select random β_x, β_y from $[\beta_{\texttt{min}}, \beta_{\texttt{max}}]$;
$\mathbf{p} = (\tilde{\mathbf{p}}_x + \beta_x \cdot r, \tilde{\mathbf{p}}_y + \beta_y \cdot r)$;

Output: The patch center $\mathbf{p}$, focus scope r.

the center of the patch. These coarse masks are with low segmentation quality, but retain macroscopic information, making our neural network pay attention to the refinement.

3.4. Focus Scope Calculation

In the inference phase for the focus view, how to choose the focus scope is of great significance for the refinement. We find that these local clicks can still have a certain effect in the global view, although they are insufficient for detail refinement. Therefore, by comparing the current and previous predictions, the influence scope of the current point can be estimated. According to the size of varied prediction areas and the object, we can decide whether to dive into a focus view around this point. The above process is based on the situation that the user clicks on the area where the prediction is wrong. In practice, the users sometimes click in the area where the prediction is already correct, *e.g.*, they put positive clicks on the predicted foreground for refining small components or negative clicks on the predicted background to constrain the boundary. For this situation, we will always go through a focus view with the focus scope as the distance between the click and the previous boundary. Because our clipping is based on a square, we use Chebyshev distance in the practical calculation. The function η is defined to calculate the Chebyshev distance between points $\mathbf{a}$ and $\mathbf{b}$: $\eta(\mathbf{a}, \mathbf{b}) = \max(|\mathbf{a}_x - \mathbf{b}_x|, |\mathbf{a}_y - \mathbf{b}_y|)$. Algorithm 2 shows the process. The default λ and ω are 0.2 and 1.75.

Algorithm 2 Focus Scope Calculation

Input: The previous and current global predictions, $\mathbf{P}', \mathbf{P}$, the patch center $\mathbf{p}$ in global view, constant λ, ω;

The prediction variation $\Delta\mathbf{P} = |\mathbf{P} - \mathbf{P}'|$;
if $\Delta\mathbf{P}_\mathbf{p}$=1 **then**
  Generate area $\mathbf{A}$ around $\mathbf{p}$ using flood fill in $\Delta\mathbf{P}$;
  Get focus judgement by $\sum\mathbf{A} < \lambda \cdot \sum\mathbf{P}$;
  $\tilde{r} = \max_{\forall\{\mathbf{a}|\mathbf{A}_\mathbf{a}=1\}} \eta(\mathbf{p}, \mathbf{a})$;
else
  $\tilde{r} = \min_{\forall\{\mathbf{a}|\mathbf{P}'_\mathbf{a}=1-\mathbf{P}'_\mathbf{p}\}} \eta(\mathbf{p}, \mathbf{a})$;
  The focus judgement is set to true;
end if
Generate the r by relax coefficient, $r = \omega \cdot \tilde{r}$;

Output: The focus judgement, focus scope r.

3.5. Progressive Focus Strategy

For our focus view, the smaller the focus scope is, the more detailed information may be focused on. Based on this, we propose Progressive Focus Strategy (PFS), which gradually focuses on areas that need to be repaired more. This is different from the traditional multi-scale way, and the scale is changing dynamically according to the variation of the previous and current patch predictions. And each time the new prediction is obtained, its part will be used as the next input in the progressive focus view. We show this iterative process in Algorithm 3. The default T is set to 3, $\hat{\omega}$ is set to 1.1, ε is set to 2.

Algorithm 3 Progressive Focus Strategy

Input: The previous patch prediction $\hat{\mathbf{P}}'$, the patch center $\hat{\mathbf{p}}$ in focus view, constant $T, \hat{\omega}, \varepsilon$;

for $t = 1, 2, \cdots, T$ and $\hat{r} \neq 0$ **do**
  Generate new patch prediction $\hat{\mathbf{P}} = \texttt{Network}(\hat{\mathbf{P}}')$;
  The prediction variation $\Delta\hat{\mathbf{P}} = |\hat{\mathbf{P}} - \hat{\mathbf{P}}'|$;
  Generate area $\hat{\mathbf{A}}$ by erode ε pixels from $\Delta\hat{\mathbf{P}}$;
  if $\sum\hat{\mathbf{A}} > 0$ **then**
    $\tilde{r} = \max_{\forall\{\mathbf{a}|\hat{\mathbf{A}}_\mathbf{a}=1\}} \eta(\hat{\mathbf{p}}, \mathbf{a})$;
    Generate the $\hat{r}$ by relax coefficient, $\hat{r} = \hat{\omega} \cdot \tilde{r}$;
    Update the previous prediction $\hat{\mathbf{P}}' \leftarrow \hat{\mathbf{P}}$;
    Crop the new patch according to $\hat{r}$;
  else
    $\hat{r} = 0$;
  end if
end for

Output: The final patch prediction $\hat{\mathbf{P}}$.

The standard PFS needs to iteratively take use of the current prediction to repair the next patch. Parallel operations cannot be realized among these multiple iterative processes. Therefore, we also propose a fast version to alleviate this problem and improve the speed by sacrificing a little performance. For each turn, we use 0.8 times the previous focus scope as the current one. At the same time, the previous prediction of the cropped patch comes from the original global prediction. In this way, the three turns can be conducted in parallel, accelerating the calculation process.

4. Experiments

4.1. Settings

Datasets. We adopt the following widely used datasets for our experiments:

- **GrabCut [40]:** The dataset contains 50 images whose background and foreground have a clear difference.
- **Berkeley [37]:** The dataset contains 96 images with 100 object masks, among which some are challenging for the task of interactive image segmentation.
- **SBD [15]:** The dataset contains 8498 images for training and 2857 images for test. In this paper, we train our model on the training set and evaluate it on the validation set, which includes 6671 object masks.
- **DAVIS [39]:** The dataset contains 50 videos, which is initially proposed for video image segmentation. As previous works [9, 20, 41] did, we use the same 345 frames for evaluation, whose masks are of high quality.

#	Candidate	Berkeley				DAVIS			
		NoC@90 ↓	IoU&5 ↑	ASSD&5 ↓	BIoU&5 ↑	NoC@90 ↓	IoU&5 ↑	ASSD&5 ↓	BIoU&5 ↑
ResNet-50	GV	4.510	0.917	2.451	0.785	7.899	0.862	9.711	0.771
	GV + FV	3.560	0.923	2.365	0.793	6.649	0.870	9.424	0.785
	GV + FV + PFS	3.440	0.929	2.170	0.804	6.377	0.870	9.338	0.787
ResNet-101	GV	4.280	0.922	2.787	0.792	7.713	0.868	9.547	0.777
	GV + FV	3.350	0.930	2.272	0.805	6.475	0.876	9.038	0.793
	GV + FV + PFS	3.010	0.933	2.050	0.811	6.223	0.879	8.840	0.796

Table 1. The core ablation study of the FocusCut. We use the metric 'NoC@90' and 'IoU&5' to evaluate the segmentation of the whole object and the 'ASSD&5' and 'BIoU&5' to evaluate the segmentation of details. The ↑ and ↓ mean that the performance is better when the metric is larger or smaller. The experiments of taking ResNet-50 and ResNet-101 as the backbone are shown in this table.

Evaluation Metrics. Following the previous works [9,20, 23, 26, 28, 30, 31, 35, 41, 51], we adopt the same robot user to simulate the clicks. To be brief, the next click will be placed at the center of the largest error region by comparing the ground truth and prediction. We adopt the Number of Click (NoC) as the evaluation metric, which counts the average number of clicks needed to achieve a fixed Intersection over Union (IoU). We set the target IoU to 85% and 90%, denoted as NoC@85 and NoC@90 respectively. The default maximum number of clicks is limited to 20 for each instance and the Number of Failure (NoF) that could not reach the target IoU will also be reported. We also use the IoU metric at the N-th clicks (IoU&N) to represent the segmentation quality. The IoU-NoC curves are also adopted to represent the convergence trend in the later phase when interacting. Since our method is more helpful for detail refinement, we also introduce two indicators for details. The boundary IoU (BIoU) [10] focuses on the IoU metric near the object boundary. The Average Symmetric Surface Distance (ASSD), which is used to evaluate the similarity of the boundaries of the prediction and ground truth, has also been used in interactive medical image segmentation [45]. For these two metrics, we also adopt that at the N-th clicks ('BIoU&20' and 'ASSD&20') to evaluate the performance. The larger the IoU and BIoU are, the better the performance is, while NoC and ASSD are on the contrary.

Implementation Details. The ResNet [16] pre-trained on ImageNet [12] is adopted as the feature extractor. The training process lasts for 40 epochs with the batch size of 8. The exponential learning rate decay strategy with the initial learning rate of 7×10^{-3} and gamma of 0.9 for each epoch is used. We take stochastic gradient descent with a momentum of 0.9 and weight decay of 5×10^{-4} for parameters optimization. We use random flipping and cropping with 384 pixels to augment the data. For annotation simulation in the global view, we follow the strategy in [31]. The Zoom-In strategy [41] is also adopted in the inference phase from the initial click. The experiments are implemented with the PyTorch [43] framework on a GPU of NVIDIA Titan XP.

Speed Analysis. The inference time is convenient to calculate because our method is composed of two branches with a shared network. We take the speed of the network, as '$1\times$'. When introducing the focus view, since these clicks with focus view can be calculated in parallel, the speed is '$2\times$'. When introducing the progressive focus strategy, the speed of fast version is still '$2\times$' because all turns can be still in parallel. The speed of standard version becomes '$4\times$' when we adopt our default T. For images with different resolutions, the input will always be resized to the fixed length as short side. In our environment, the '$1\times$' speed of ResNet-50 and ResNet-101 are 0.0295 and 0.0346 Seconds Per Click (SPC) with 384 pixels as the fixed length. Even for our standard version, the inference speed is enough to meet the needs of practical applications.

4.2. Ablation Study

As shown in Tab. 1, we carry out the core ablation studies to demonstrate the necessity of each component adopted in FocusCut. We choose the Berkeley and DAVIS datasets to do these experiments because the Berkeley dataset is similar but larger than GrabCut dataset, and the annotation quality of the SBD dataset is poor. We use four metrics for these experiments in Tab. 1, among which the first two are for the whole object segmentation, and the last two are for the details. For progressive focus strategy, we also ablate the strategy with different turns and settings.

Introducing Focus View. For the core part of introducing the focus view, the performance improvement is significant, whether for the whole object or the details. As for the core metric, the NoC decreases by about one click in Berkeley and DAVIS datasets. We compare the other three metrics at the 5-th click. The improvement of IoU metric shows that the focus view brings a more complete object. The increase of BIoU and decrease of ASSD also imply that our method improves the details obviously and provide a more accurate boundary. Whether ResNet-50 or ResNet-101 and no matter which metric, their improvements are obvious. The introducing of the focus view is undoubtedly useful.

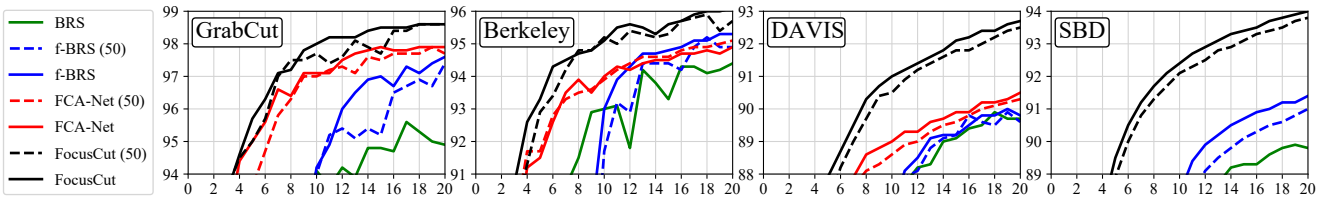

Figure 4. The local IoU-NoC curves to represent the convergence trend in four datasets. '(50)' means taking ResNet-50 as the backbone.

Method		GrabCut		Berkeley	SBD		DAVIS	
		NoC@85	NoC@90	NoC@90	NoC@85	NoC@90	NoC@85	NoC@90
§ DOS w/o GC [51] $_{CVPR16}$		8.02	12.59	-	14.30	16.79	12.52	17.11
§ DOS with GC [51] $_{CVPR16}$		5.08	6.08	-	9.22	12.80	9.03	12.58
§ RIS-Net [28] $_{ICCV17}$		-	5.00	6.03	-	-	-	-
† Latent diversity [26] $_{CVPR18}$		3.20	4.79	-	7.41	10.78	5.05	9.57
§ CM guidance [35] $_{CVPR19}$		-	3.58	5.60	-	-	-	-
† BRS [20] $_{CVPR19}$		2.60	3.60	5.08	6.59	9.78	5.58	8.24
§ MutiSeg [30] $_{ICCV19}$		-	2.30	4.00	-	-	-	-
§ Continuous Adaptation [23] $_{ECCV20}$		-	3.07	4.94	-	-	5.16	-
§ FCANet [31] $_{CVPR20}$	ResNet-50	2.18	2.62	4.66	-	-	5.54	8.83
	ResNet-101	1.88	2.14	4.19	-	-	5.38	7.90
† f-BRS [41] $_{CVPR20}$	ResNet-50	2.50	2.98	4.34	5.06	8.08	5.39	7.81
	ResNet-101	2.30	2.72	4.57	4.81	7.73	5.04	7.41
† CDNet [9] $_{ICCV21}$	ResNet-50	2.22	2.64	3.69	4.37	7.87	5.17	6.66
	ResNet-101	2.42	2.76	3.65	4.73	7.66	5.33	6.97
† FocusCut* $_{Ours}$	ResNet-50	1.58	1.78	3.48	3.76	5.86	5.18	6.59
	ResNet-101	1.48	1.68	3.22	3.54	5.55	4.98	6.32
† FocusCut $_{Ours}$	ResNet-50	1.60	1.78	3.44	3.62	5.66	5.00	6.38
	ResNet-101	1.46	1.64	3.01	3.40	5.31	4.85	6.22

Table 2. Comparison of the NoC metric with other methods in four evaluation datasets. Symbol † means adopting SBD [15] dataset for training. § means adopting the Augmented PASCAL VOC [13, 15] dataset for training. * means the fast version of the FocusCut.

Progressive Focus Strategy. As shown in Tab. 1, the progressive focus strategy can assist our method and further improve its performance. In the standard version, the channel of the previous prediction will be updated iteratively according to the output in the last turn. Tab. 3 shows the results without iterative prediction, from which we can find that the performance will have a certain degree of degradation in this situation. In Fig. 5, we also show the metric of NoC@90 with different turns for the strategy. The performance improvement of the first few turns is obvious, while that of the latter turns fluctuates because the patch size is too small. Since the operation of iterative prediction and step-by-step determination of the focus scope need to be carried out according to the previous result, it cannot be implemented in parallel on devices. The standard version may sacrifice a certain speed, but we also provide a fast version shown in Tab. 2, in which the reduction factor of the focus scope is set as a constant. In this way, the time used for updating prediction can be saved, but it can still achieve excellent performance. Users can choose whichever version they want according to their demand and environment.

4.3. Comparison & Discussion

Performance Evaluation. As shown in Tab. 2, we compare our method with others on the most common NoC metrics. The GrabCut, Berkeley, SBD, and DAVIS datasets are all evaluated like others. All the performances of ResNet-50 and ResNet-101 are available in this table. We can find that our proposed method has achieved the state-of-the-art performance in all datasets. In Tab. 2, we also provide the fast version of our method, which is slightly worse than our standard version, but still performs well compared to other methods. It is worth noting that almost no parameters or modules are inserted into the baseline network, which strongly reflects the validity of the FocusCut. To reflect the convergence trend, we crop and enlarge the IoU-NoC curves and show them in Fig. 4. In the figure, we choose some recent methods with available codes. The FCA-Net is not in SBD subfigure because it use Augmented PASCAL for training. We can find that in the later interactive phase, our method still has a certain upward trend. The results at 20-th click shows that our method has a higher upper bound, reflecting that it can segment the objects more finely.

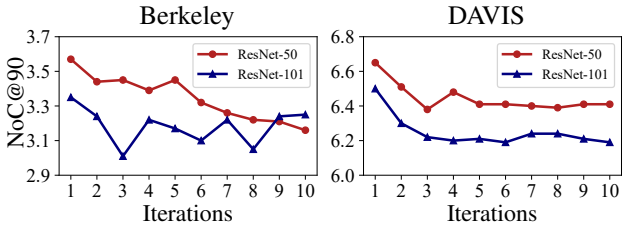

Figure 5. NoC@90 *vs*. iterations for progressive focus strategy.

Setting	Berkeley		DAVIS	
	ResNet-50	ResNet-101	ResNet-50	ResNet-101
w/o IP	3.51	3.11	6.56	6.38
w/ IP	3.44	3.01	6.38	6.22

Table 3. The comparison of the NoC@90 metric for the progressive focus strategy with or without iterative prediction (IP).

Method	Berkeley		DAVIS	
	ASSD	BIoU	ASSD	BIoU
DOS [51]	4.150	0.594	7.402	0.741
LD [26]	2.218	0.773	7.186	0.776
BRS [20]	1.099	0.866	6.188	0.829
f-BRS [41]	1.218	0.866	6.318	0.825
FCA-Net [31]	1.147	0.861	6.051	0.834
Ours	0.928	0.892	4.427	0.874

Table 4. The comparison of the detail metrics (ASSD and BIoU) at the 20-th click for the methods with available codes. The last four are based on ResNet-101. LD is short for latent diversity [26].

Segmentation Quality. Fig. 6 shows some cases where our FocusCut can play a dominant role. For example, in the position of small parts such as the wheels of an aircraft, the FocusCut can generate excellent prediction with only one click in the foreground. In some places where there are many gaps, such as the region between dog legs or people's fingers in the picture, although background points are provided, the neural network is likely to over suppress them from the global view, while our FocusCut can deal with this situation well. Like previous works [9,20,41], we also show the metric of Number of Failure (NoF) images for recent methods in Tab. 5. The result with 100 clicks as maximum can reflect the performance for segmenting details. Whether the metric is NoF or NoC, we have exceeded all the latest methods and achieved a new state-of-the-art performance. What's more, we also compare our method to other methods whose codes are available with BIoU and ASSD metrics, and the experimental results are available in Tab. 4. Obviously, our method is superior to other methods in both ASSD and BIoU, showing the effectiveness of our method for detail refinement. In practical use, we provide a small window serving as a magnifying glass on the user interface to show the area near the mouse, which helps users click on the more accurate location in small areas.

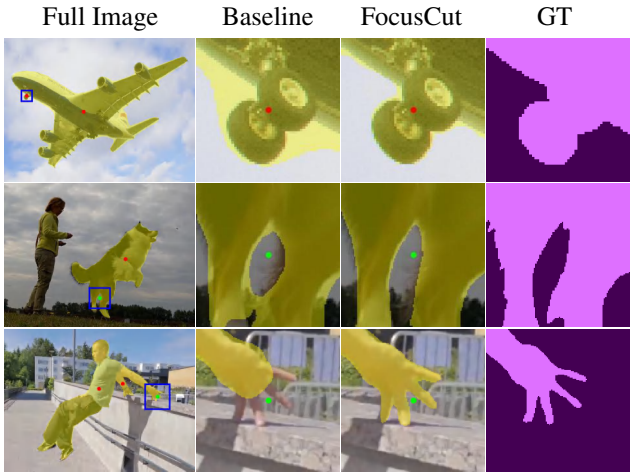

Figure 6. The quality results of the FocusCut and the comparison with the baseline. The predictions and clicks are shown above.

Method	NoF_{20}@90	NoF_{100}@90	NoC_{100}@90
BRS [20]	77	51	20.89
f-BRS [41]	78	50	20.70
FCA-Net [31]	87	54	22.56
CDNet [9]	65	48	18.59
Ours	57	43	17.42

Table 5. Comparison in different clicks setting in DAVIS dataset with ResNet-50. NoF_N@90 indicates the Number of Failure images that could not reach IoU 0.9 under N clicks. NoC_{100}@90 metric is the same as NoC@90 with maximum clicks as 100.

Limitation Analysis. Our method requires multiple runs of segmentation, the inference time will inevitably increase. Even for the fast version, the computational burden is actually the same. For some old equipments, the time consumption and computational burden might still be a bottleneck.

5. Conclusion

In this paper, we introduce the focus view to grasp the user's intention from the newly-input clicks' eyes. We concertize the focus view with a simple yet effective pipeline, FocusCut, in which the prediction of the cropped click-centered patch is updated by the shared network with the global view. The patch updating is progressive under multiple adaptive scopes. Extensive experiments on four datasets have demonstrated the superiority of our FocusCut, setting the new state-of-the-art performance.

Acknowledgment: This work is funded by the National Key Research and Development Program of China (NO. 2018AAA0100400), NSFC (NO. 61922046), S&T innovation project from Chinese Ministry of Education, China Postdoctoral Science Foundation (NO.2021M701780). We also gratefully acknowledge the support of MindSpore, CANN, and Ascend AI Processor used for this research.

References

[1] David Acuna, Huan Ling, Amlan Kar, and Sanja Fidler. Efficient interactive annotation of segmentation datasets with polygon-rnn++. In *CVPR*, 2018. 1, 2

[2] Eirikur Agustsson, Jasper RR Uijlings, and Vittorio Ferrari. Interactive full image segmentation by considering all regions jointly. In *CVPR*, 2019. 1, 2

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

[4] Rodrigo Benenson, Stefan Popov, and Vittorio Ferrari. Large-scale interactive object segmentation with human annotators. In *CVPR*, 2019. 2

[5] Yuri Y Boykov and M-P Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In *ICCV*, 2001. 2

[6] Lluis Castrejon, Kaustav Kundu, Raquel Urtasun, and Sanja Fidler. Annotating object instances with a polygon-rnn. In *CVPR*, 2017. 1, 2

[7] 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. 3

[8] Wuyang Chen, Ziyu Jiang, Zhangyang Wang, Kexin Cui, and Xiaoning Qian. Collaborative global-local networks for memory-efficient segmentation of ultra-high resolution images. In *CVPR*, 2019. 2

[9] Xi Chen, Zhiyan Zhao, Feiwu Yu, Yilei Zhang, and Manni Duan. Conditional diffusion for interactive segmentation. In *ICCV*, 2021. 1, 2, 5, 6, 7, 8

[10] Bowen Cheng, Ross Girshick, Piotr Dollar, Alexander C. Berg, and Alexander Kirillov. Boundary iou: Improving object-centric image segmentation evaluation. In *CVPR*, 2021. 6

[11] Ho Kei Cheng, Jihoon Chung, Yu-Wing Tai, and Chi-Keung Tang. Cascadepsp: toward class-agnostic and very high-resolution segmentation via global and local refinement. In *CVPR*, 2020. 2, 4

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 6

[13] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010. 7

[14] Leo Grady. Random walks for image segmentation. *IEEE TPAMI*, 2006. 2

[15] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *ICCV*, 2011. 2, 5, 7

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3, 6

[17] Yang Hu, Andrea Soltoggio, Russell Lock, and Steve Carter. A fully convolutional two-stream fusion network for interactive image segmentation. *NN*, 2019. 2

[18] Chuong Huynh, Anh Tuan Tran, Khoa Luu, and Minh Hoai. Progressive semantic segmentation. In *CVPR*, 2021. 2

[19] Suyog Dutt Jain and Kristen Grauman. Click carving: Interactive object segmentation in images and videos with point clicks. *IJCV*, 2019. 2

[20] Won-Dong Jang and Chang-Su Kim. Interactive image segmentation via backpropagating refinement scheme. In *CVPR*, 2019. 1, 2, 5, 6, 7, 8

[21] Meng Jian and Cheolkon Jung. Interactive image segmentation using adaptive constraint propagation. *IEEE TIP*, 2016. 2

[22] Tae Hoon Kim, Kyoung Mu Lee, and Sang Uk Lee. Generative image segmentation using random walks with restart. In *ECCV*, 2008. 2

[23] Theodora Kontogianni, Michael Gygli, Jasper Uijlings, and Vittorio Ferrari. Continuous adaptation for interactive object segmentation by learning from corrections. In *ECCV*, 2020. 2, 6, 7

[24] Hoang Le, Long Mai, Brian Price, Scott Cohen, Hailin Jin, and Feng Liu. Interactive boundary prediction for object selection. In *ECCV*, 2018. 2

[25] Yin Li, Jian Sun, Chi-Keung Tang, and Heung-Yeung Shum. Lazy snapping. *ACM TOG*, 2004. 2

[26] Zhuwen Li, Qifeng Chen, and Vladlen Koltun. Interactive image segmentation with latent diversity. In *CVPR*, 2018. 1, 2, 6, 7, 8

[27] Xuan Liao, Wenhao Li, Qisen Xu, Xiangfeng Wang, Bo Jin, Xiaoyun Zhang, Yanfeng Wang, and Ya Zhang. Iteratively-refined interactive 3d medical image segmentation with multi-agent reinforcement learning. In *CVPR*, 2020. 2

[28] JunHao Liew, Yunchao Wei, Wei Xiong, Sim-Heng Ong, and Jiashi Feng. Regional interactive image segmentation networks. In *ICCV*, 2017. 2, 6, 7

[29] Jun Hao Liew, Scott Cohen, Brian Price, Long Mai, and Jiashi Feng. Deep interactive thin object selection. In *WACV*, 2021. 1, 2

[30] 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*, 2019. 1, 2, 6, 7

[31] Zheng Lin, Zhao Zhang, Lin-Zhuo Chen, Ming-Ming Cheng, and Shao-Ping Lu. Interactive image segmentation with first click attention. In *CVPR*, 2020. 1, 2, 6, 7, 8

[32] Huan Ling, Jun Gao, Amlan Kar, Wenzheng Chen, and Sanja Fidler. Fast interactive object annotation with curve-gcn. In *CVPR*, 2019. 1, 2

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

[34] Soumajit Majumder, Abhinav Rai, Ansh Khurana, and Angela Yao. Two-in-one refinement for interactive segmentation. In *BMVC*, 2020. 1

[35] Soumajit Majumder and Angela Yao. Content-aware multi-level guidance for interactive instance segmentation. In *CVPR*, 2019. 1, 2, 6, 7

[36] Kevis-Kokitsi Maninis, Sergi Caelles, Jordi Pont-Tuset, and Luc Van Gool. Deep extreme cut: From extreme points to object segmentation. In *CVPR*, 2018. 1, 2

[37] Kevin McGuinness and Noel E O'connor. A comparative evaluation of interactive segmentation algorithms. *PR*, 2010. 2, 5

[38] Eric N Mortensen and William A Barrett. Intelligent scissors for image composition. In *SIGGRAPH*, 1995. 2

[39] 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*, 2016. 2, 5

[40] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM TOG*, 2004. 2, 5

[41] Konstantin Sofiiuk, Ilia Petrov, Olga Barinova, and Anton Konushin. f-brs: Rethinking backpropagating refinement for interactive segmentation. In *CVPR*, 2020. 1, 2, 5, 6, 7, 8

[42] Gwangmo Song, Heesoo Myeong, and Kyoung Mu Lee. Seednet: Automatic seed generation with deep reinforcement learning for robust interactive segmentation. In *CVPR*, 2018. 2

[43] Benoit Steiner, Zachary DeVito, Soumith Chintala, Sam Gross, Adam Paszke, Francisco Massa, Adam Lerer, Gregory Chanan, Zeming Lin, Edward Yang, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 6

[44] Hiroki Tokunaga, Yuki Teramoto, Akihiko Yoshizawa, and Ryoma Bise. Adaptive weighting multi-field-of-view cnn for semantic segmentation in pathology. In *CVPR*, 2019. 2

[45] Guotai Wang, Maria A Zuluaga, Wenqi Li, Rosalind Pratt, Premal A Patel, Michael Aertsen, Tom Doel, Anna L David, Jan Deprest, Sébastien Ourselin, et al. Deepigeos: a deep interactive geodesic framework for medical image segmentation. *IEEE TPAMI*, 2018. 6

[46] Tao Wang, Zexuan Ji, Quansen Sun, Qiang Chen, Qi Ge, and Jian Yang. Diffusive likelihood for interactive image segmentation. *PR*, 2018. 2

[47] Tao Wang, Jian Yang, Zexuan Ji, and Quansen Sun. Probabilistic diffusion for interactive image segmentation. *IEEE TIP*, 2018. 2

[48] 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*, 2014. 1

[49] Fangting Xia, Peng Wang, Liang-Chieh Chen, and Alan L Yuille. Zoom better to see clearer: Human and object parsing with hierarchical auto-zoom net. In *ECCV*, 2016. 2

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

[51] Ning Xu, Brian Price, Scott Cohen, Jimei Yang, and Thomas S Huang. Deep interactive object selection. In *CVPR*, 2016. 2, 6, 7, 8

[52] Shiyin Zhang, Jun Hao Liew, Yunchao Wei, Shikui Wei, and Yao Zhao. Interactive object segmentation with inside-outside guidance. In *CVPR*, 2020. 1, 2