FocusCut：深入聚焦视角的交互式分割方法

林铮 [1]　段正鹏 [1]　张钊 [1,2]　郭春乐 [1*]　程明明 [1]
[1] 南开大学，计算机学院　[2] 商汤科技
http://mmcheng.net/focuscut/

Abstract

交互式图像分割在像素级别的标注和图像编辑中是一项必不可少的工具。为了获得高准确率的二值分割掩膜，用户倾向于在边缘、孔洞等对象细节周围添加交互点来实现高效的细化。目前的方法把这些修复点当作指导，共同确定全局预测。但是，全局视角使得模型丢失来自较后交互点的关注，并且与用户的意图不相符。在这篇论文中，我们深入聚焦每一个交互点，并赋予它们在分割物体细节时决定性的作用。为了验证聚焦视角的必要性，我们设计一个简单但有效的网络管道，称作 *FocusCut*。该网络同时集成全局分割和局部细化的功能。在获得对目标的全局预测后，它以自适应的范围在原图上裁剪以交互点为中心的区块，来逐渐细化局部的分割结果。我们的方法在不需要用户感知和参数增加的情况下取得了目前最高水平的性能。大量的实验和可视化结果证明，*FocusCut* 使得交互式分割研究中的超精细分割成为了可能。

1. 引言

交互式图像分割旨在以最小的交互成本获得目标对象的准确二值分割掩膜，现已发展成为提供像素级数据标注和图像编辑必不可少的工具。现在该领域的研究主要集中在两个方面：一是更高效的用户交互模式，二是更有效地利用用户提供的交互。对于前者，各种交互模式被广泛探索，常见的方法主要有基于边界框 [50]、基于多边形 [1,6,32]、基于交互点 [2,29,36]、基于线条 [3,48] 和一些交互方式的组合 [34,52]。其中，基于点击的方法因其简洁性已经成为主流。对于后者，研究人员探索了交互歧义 [9,26,30]，输入信息 [31,35]，网络反向传播 [20,41] 等。这些方法都在不改变用户输入的前提下，取得了更好的分割结果。

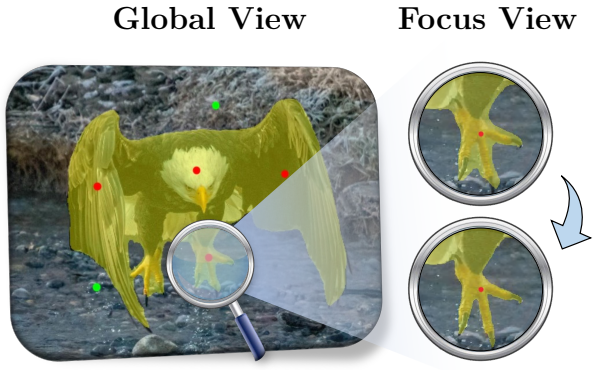

图 1. FocusCut 的可视化展示。鹰爪的细节通过一个额外的聚焦视角进行优化。用户提供的红色和绿色的交互点分别表示交互分割中的前景和背景。黄色的掩膜表示预测结果。

近年来，随着大屏幕设备的增加和人们审美水平的提高，图像标注和图像编辑对精细化分割掩膜的需求量都在增加。在高精度交互分割中，边缘、孔洞等目标细节的精细化改进通常需要更多的交互点和交互时间。当用户在标记错误的区域添加交互点，他们倾向于关注细节区域来实现高效的修复。但是，目前的方法把之前的交互点考虑到一起来确定全局预测。在一轮新的交互中，一个共同的预测过程可能会弱化新输入的交互点对其周围细节的决定性影响，并且返回不合意的结果。

为了实现更加高效的优化，我们深入一个交互点的视角来考虑其周围的信息，本文称之为聚焦视角。在本文中，我们设计了一个名为 FocusCut 的简洁管道来验证聚焦视角的重要性。交互分割网络的原始功能已经被改变，取而代之的是，我们给它赋予了一个新的作用，允许它不仅能够分割目标对象，而且可以修复局

*郭春乐是通讯作者

部细节。具体而言，经过全局分割（本文称之为全局视角）以后，它从原始图像中裁剪出一个以新交互点为中心的局部区块作为聚焦视角，使用同一个网络来进一步优化目标的细节。渐进的裁剪范围根据全局视角中的预测变化进行动态调整。之后，裁剪范围依据本文提出的渐进聚焦策略逐步减小。为了公平地与其他方法做对比，并且证明我们观点的有效性，几乎所有无参和特定的模块都被插入到交互分割任务常用的网络结构中。我们在 GrabCut [40], Berkeley [37], SBD [15] 和 DAVIS [39] 等数据集上开展的所有实验都证明了 FocusCut 的有效性。

本文的贡献总结如下:

- 我们介绍了聚焦视角，通过考虑来自每个交互点的局部分割来理解用户的意图。
- 基于以上观点，我们提出了 FocusCut，一种简单但有效的网络管道来加强这种局部优化。
- FocusCut 在不增加额外参数的情况下取得了目前最优异的性能，并且可视化结果反映了该模型在精细分割上的有效性。

2. 相关的工作

2.1. 交互式图像分割

大多数传统方法使用图像的低层次特征来构建模型，例如智能剪刀 [38] 和懒人抠图 [25]。基于 GraphCut [5]，Rover 等人提出了一种名为 GrabCut [40], 的方法使得模型更加实用。Grady 等人提出了随机游走算法 [14] 来确定每一个未标记像素的概率。Kim 等人 ~ [22] 通过引入一个重开始模拟来提高模型的性能。但是，由于过多地关注低层次特征，这些方法在复杂环境中可能会失效。

近年来，尽管还有一些工作 [21, 46, 47] 继续提升传统方法，基于深度学习的方法因其综合考虑全局和局部特征的能力已经成为了这个领域的主流。除了一些基于循环神经网络 [1, 6]，图卷积神经网络 [32] 和强化学习 [27, 42] 的方法，大多数研究开展在传统的卷积神经网络上。在这个任务中，人们已经探索了多种交互模式。例如，极值点已经被用于常见对象 [36]、薄对象 [29] 和全图 [2] 的分割。边界交互点 [19, 24] 也被采纳作为一种高效的交互。融合交互方式，例如边界框和交互点 [4, 52]，在这个领域中也很受欢迎。在这些方法中，在前景和背景中提供交互点已经成为了主流，本文亦研究了这种交互模式。

对于这种交互模式，Xu 等人 [51] 首次提出了一种基于深度学习的算法，该算法带有交互点图谱转换和多种随机采样策略。为了充分利用用户的交互，Liew 等人 [28] 提出了 RSI-Net，利用来自交互点对的局部区域来优化分割结果。Hu 等人 [17] 为这个任务提出了一种双分枝结构。Majumder [35] 等人通过生成内容感知的导航图谱来提升用户交互点的转换。Jang 等人 [20] 提出了 BRS 来纠正初始结果中错误标记的像素，这个作用在 f-BRS [41] 中得到了提升。Kontogianni 等人 [23] 采用用户的修正作为训练样本，并且立即更新模型的参数。为了解决用户交互的歧义，Li 等人 [26] 连接两个卷积神经网络来训练和挑选出合适的结果。Liew 等人 [30] 将尺度多样性引入到模型中来帮助用户快速定位他们想要的目标。Lin 等人 [31] 强调第一个交互点的重要作用，并且将其作为特俗的指导。Chen 等人 [9] 引入一种非局部的方法来充分利用用户的线索。大多数方法将用户的交互点转换为一个同整幅图像尺寸相同的指导图谱。但是，我们在一个聚焦视角中查看每一个额外的交互点，充分地利用它们的潜力。

2.2. 分割中的局部视角

局部信息已经被很多分割任务所充分利用。HAZN [49] 能够自适应地调整对对象或部分对象的视角范围来优化分割结果。GLNet [8] 聚合局部和全局分支捕获的特征图谱。此外，对于语义分割，AWMF-CNN [44] 分别为局部区块的不同放大率赋予权重。CascadePSP [11] 通过强化模块从原始图像中提取图像区块。相似地，MagNet [18] 以一种渐进的方式来优化不同尺度局部区块的分割结果。但是，对于语义分割任务，滑动窗口策略会不可避免地造成计算量和时间的成本。由于交互式分割的特殊性，聚焦视角可以通过交互来决定，因此避免了这个缺点。

在交互式分割中，RIS-Net [28] 已经证明了局部优化的重要性。它通过为每个正交互点寻找最近的负交互点来生成局部区块，并且构建一个边界框。局部特征通过主要分支的 ROI 池化层来提取，该分支以图像和转换的交互点的拼接结果作为输入。也就是说，局部优化依然受到整幅图像和其他交互点的影响，这会在某

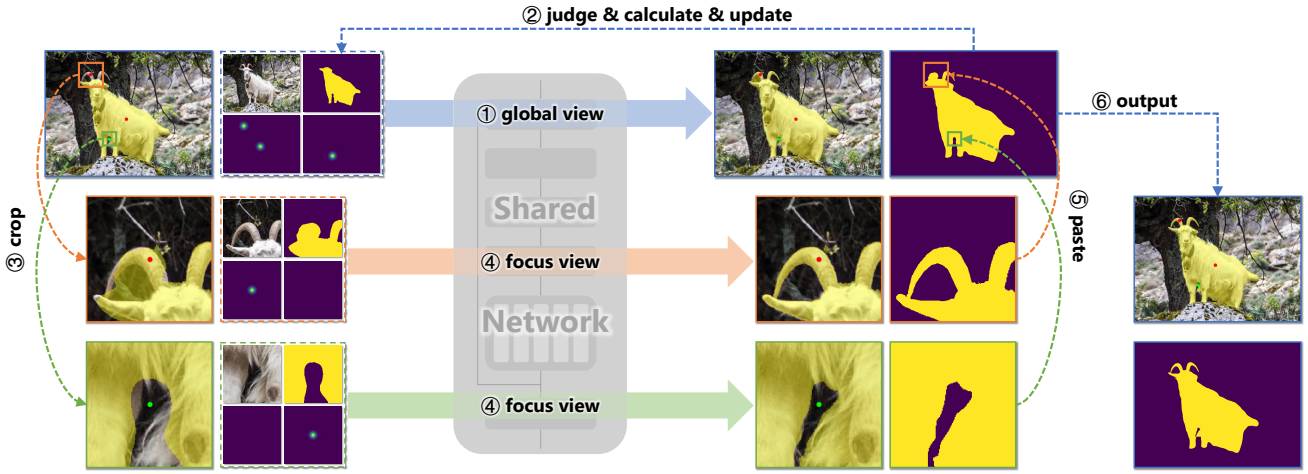

图 2. FocusCut 网络的管线。这个过程被划分为 6 个阶段：（1）把整幅图像的 6 通道输入输入到共享的网络中来生成全局视角的预测；（2）对于当前的交互点，判断是否聚焦并且基于当前预测和之前预测的变化计算聚焦的范围，然后使用当前的预测更新之前的预测；（3）分别为每个不同聚焦范围的交互点从原始图像中裁剪区块；（4）把聚焦区块的输入送入到网络中生成聚焦视角下的局部预测；（5）把区块预测粘贴到全局预测中；（6）输出最终的预测结果。

种程度上弱化局部交互点的主导作用。此外，由于网络的下采样操作，局部特征有些丢失。我们更进一步，采用了一种更纯粹的焦点视图进行局部优化，直接把以每个交互点为中心的局部区块送入网络并且完全地忽略了整个图像和其他远距离交互点的影响。

3. 提出的方法

3.1. 回顾经典的网络管道

随着神经网络的发展，近年来大多数交互式分割研究是通过引入卷积神经网络而开展的。因为交互式分割可以被视为一种特殊类型的分割任务，很多方法是基于语义分割中以 DeepLab v3+ [7] 为代表的 DeepLab 系列网络来设计的。这个网络结构包含一个骨干网络，一个空洞空间卷积池化金字塔（Atrous Spatial Pyramid Pooling，ASPP）部分和一个解码器部分。对于骨干网络，在交互式分割中大多采用 ResNet [16]。ASPP 部分包含 4 个空洞卷积分支和一个全局平均池化分支。解码器部分通过融合骨干网络的低层次特征优化 ASPP 模块的输出来生成最终的预测。对于交互式分割任务，输入还应该包含交互点的信息。交互点位置将会被转换为两个交互点图谱，例如距离、磁盘和我们使用的高斯图谱，分别表示正点和负点。交互式分割中的大多数工作修改了网络的输入部分，采用一个由 RGB 图像和两个交互点图谱拼接而成的 5 通道图谱作为输入。在具体实现时，可以添加额外的头将一个 5 通道图谱编码为一个 3 通道图谱来满足标准的结构或者像我们那样直接改变第一个卷积层。输出将会被标注结果使用二值交叉熵损失来监督并且被二值化成最终的预测。

3.2. FocusCut 网络管道

在交互式分割的过程中，用户经常通过提供更多的前景和背景点来修正不正确的分割区域。随着交互点数目的增加，较后的交互点逐渐被用于修正更加局部的区域。特别是在较后的阶段，可能会聚集很多交互点来修正一个小区域。由于感受野的尺寸和网络的下采样操作，很难同时分割出整个对象和细节区域。

Fig. 2展示了本文提出的交互式图像分割模型 FocusCut 的管道。该网络管道包含两个交互式视图。一种是全局视角，对整个对象进行分割，另一种是聚焦视角，根据交互点附近的全局分割结果进行细化分割。为了体现我们方法的有效性，我们决定尽可能不改变常用网络的架构。我们以输出步幅为 16 的 DeepLab v3+ 作为基础网络。不同的是，我们将其视为共享网络，不仅可以学习整个对象的分割，还可以学习局部区域的细化。为此，我们需要统一这两个输入。由于聚焦视图中的细化是基于粗糙的掩膜生成的，我们为输入添加一个先前预测的额外通道。我们希望我们的网络除了目标分割之外，还学习基于先前的预测和交互点生成更准确的分割。为了实现这个目标，我们交替使用全局

视角和聚焦视角的数据来训练我们的网络。对于全局视角，我们采用迭代训练策略 [33]。如果是迭代步骤，则将粗略预测设置为先前的分割。对于其他情况，则将其设置为空图谱。RGB 图像包含整个对象，并且交互点也是根据对象掩膜模拟的，其中将包含至少一个正点来指示对象的位置。在全局视图中，网络将这个 6 通道图作为输入来生成整个目标的预测。对于我们的聚焦视角，则使用表示目标局部信息的区块样本来训练网络。我们将会在 Sec. 3.3中详细描述生成区块样本的过程。如 Fig. 2所示，这个阶段的输入图谱依然包含 6 个通道。但是，RGB 图像会是从原始图像中裁剪出来的局部区域，不代表物体，会更加注重细节。与全局视角中的交互点图谱不同，这些交互点图谱必须包含图谱的中心点，这可能是正点也可能是负点。我们将通过处理正确的分割结果来生成粗分割以降低其精细度。这些图谱将会被计算并且送入到网络中。

Fig. 2详细地展示了推理阶段。在这个阶段，用户会不断地点击，直到结果满足需求。由于第一个交互点必然会分割整个对象，我们从第二个交互点开始引入聚焦视角。添加当前交互点时，将首先采用全局视角的管道，如 Fig. 2的顶部所示。根据当前点击的位置以及当前预测 $\mathbf{P}$ 与之前预测 $\mathbf{P}'$ 在全局视图中的差异，判断交互点是否应该经过额外的聚焦视角路径。如果采用聚焦视角，则计算当前点击的聚焦范围 r，这些将会在 Sec. 3.4中被介绍。然后，根据聚焦范围，从原图、交互点图和当前预测结果上裁剪下相应的局部区块，输入到聚焦视图的路径中来生成局部预测 $\widehat{\mathbf{P}}$，如 Fig. 2的底部所示。此外，这里的图像区块是从原始图像中裁剪出来的。对于高分辨率图像，这有助于避免信息丢失并获得更清晰的 RGB 区块。最后，局部预测将被粘贴回原始预测。如果区块之间存在重叠，则重叠部分采用它们的平均值。在 Sec. 3.4中，我们还提供了一种渐进式聚焦策略，以迭代方式关注局部区域以取得更好的效果。

3.3. 聚焦区块模拟

在这个部分，我们将会介绍我们旨在生成交互点周围聚焦区快的模拟算法，用于模型的训练。我们发现在交互式图像分割的中后期，用户通常单击对象边界以使边界更准确，并且对象细节通常在边界附近。我们生成区块以模拟这种情况。我们在对象的边界上选择一个点，并且基于 $\beta \in [\beta_{min}, \beta_{max}]$ 赋予该点一个随机的偏移作为这个区块的中心点。聚焦范围 r 是一个与对象尺寸相关的随机数字。对象尺寸由从标注结果 $\mathbf{G}$ 和随机系数 $\alpha \in [\alpha_{min}, \alpha_{max}]$ 计算出的 k 来反映。详细的计算过程在 Algorithm 1中描述。在我们的实验中，α_{min}、α_{max}、β_{min} 和 β_{max} 的默认值分别设置为 0.2、0.8、-0.3 和 0.3。

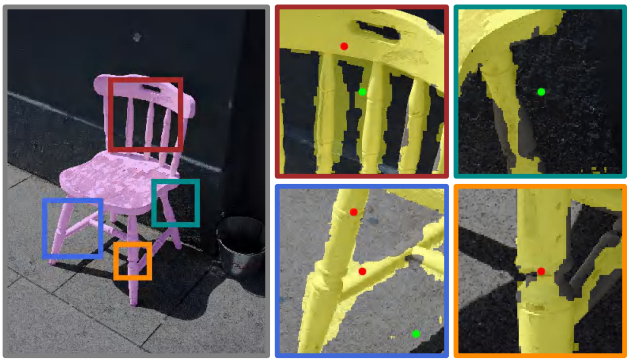

图 3. 聚焦区块模拟的样例。右边区块中的边界颜色表示左边对应的裁剪。左边的粉红色掩膜表示标注结果，右边的黄色表示生成的粗糙掩膜。这些模拟的交互点也会显示在区块中。

Algorithm 1 聚焦区块模拟

Input: 正确的分割结果 $\mathbf{G}$, 常数 $\alpha_{\min}$, $\alpha_{\max}$, $\beta_{\min}$, $\beta_{\max}$;

目标物体大小 $k = \sqrt{\sum_{i,j} \mathbf{G}_{i,j}}$, $\mathbf{G}_{i,j} \in \{0, 1\}$;
选择随机数 α 在 $[\alpha_{\min}, \alpha_{\max}]$;
聚焦范围 $r = \alpha \cdot k$;
生成边界图 $\mathbf{B}$ 根据 $\mathbf{G}$, $\mathbf{B}_{i,j} \in \{0, 1\}$;
选择随机的边界点 $\tilde{\mathbf{p}}(x, y)$, $\mathbf{B}_{x,y} = 1$;
选择随机数 β_x, β_y 在 $[\beta_{\min}, \beta_{\max}]$;
$\mathbf{p} = (\tilde{\mathbf{p}}_x + \beta_x \cdot r, \tilde{\mathbf{p}}_y + \beta_y \cdot r)$;

Output: 区块中心点 $\mathbf{p}$, 聚焦范围 r.

基于区块中心 $\mathbf{p}$ 和聚焦范围 r，我们在图像和对应标注结果的 $(\mathbf{p}_x - r, \mathbf{p}_y - r)$ 到 $(\mathbf{p}_x + r, \mathbf{p}_y + r)$ 区间中裁剪出一个正方形区块。基于区块数据，我们通过随机膨胀和侵蚀方法 [11] 生成一个粗糙的掩膜作为之前预测。中心点将会作为用户点击始终包含其中。我们还将在区块中选出 0 ∼ 3 个正点和负点来模拟区块中心周围的交互点。这些区块交互点将被转换为交互点图谱，并和原图像，粗糙分割结果一起输入到网络中。

在图 Fig. 3中，我们展示了从一个包含椅子的图片和对应的标注结果中模拟裁剪的区块。可以看出，我们的算法模拟用户的交互位置，并且裁剪了不同的部分。在区块中心中至少包含一个交互点。这些粗糙掩膜的分割质量低，但是保留了宏观的信息，使得我们的神经网络关注分割的细化。

3.4. 聚焦范围计算

在聚焦视角的推理阶段，如何选出聚焦范围对细化具有重要意义。我们发现局部交互点尽管不足以优化细节，但是在全局视角中依然有一定的影响。因此，可以通过比较当前和之前的预测可以估计出当前交互点的影响范围。根据不同的预测区域和对象的尺寸，我们可以决定是否深入观察这个交互点周围的聚焦视角。上述过程基于模拟用户点击在预测错误的区域。在实践中，用户有时点击在已经正确预测的区域，例如他们在预测的前景上添加正交互点来细化小组件或者在预测的背景上添加负交互点来限制边界。为了应对这种状况，我们将始终以交互点和上一个边界之间的距离作为焦点范围来查看聚焦视图。因为我们的裁剪基于一个正方形，我们在实际计算时使用切比雪夫距离。我们定义函数 η 来计算交互点 $\mathbf{a}$ and $\mathbf{b}$ 间的切比雪夫距离：$\eta(\mathbf{a},\mathbf{b}) = \max(|\mathbf{a}_x - \mathbf{b}_x|, |\mathbf{a}_y - \mathbf{b}_y|)$. 这个过程如 Algorithm 2所示。$\lambda$ 和 ω 的默认值分别为 0.2 和 1.75。

Algorithm 2 聚焦范围计算

Input: 之前和现在的全局预测, $\mathbf{P}', \mathbf{P}$,
区块中心 $\mathbf{p}$ 在全局视角中, 常数 λ, ω;

预测中变化区域 $\Delta\mathbf{P} = |\mathbf{P} - \mathbf{P}'|$;
if $\Delta\mathbf{P}_\mathbf{p}$=1 **then**
  生成区域 $\mathbf{A}$ 在 $\mathbf{p}$ 周围用 flood fill 在 $\Delta\mathbf{P}$;
  得到聚焦判定根据 $\sum\mathbf{A} < \lambda \cdot \sum\mathbf{P}$;
  $\tilde{r} = \max_{\forall\{\mathbf{a}|\mathbf{A}_\mathbf{a}=1\}} \eta(\mathbf{p},\mathbf{a})$;
else
  $\tilde{r} = \min_{\forall\{\mathbf{a}|\mathbf{P}'_\mathbf{a}=1-\mathbf{P}'_\mathbf{p}\}} \eta(\mathbf{p},\mathbf{a})$;
  聚焦判定设为 true;
end if
生成 r 通过余量系数, $r = \omega \cdot \tilde{r}$;

Output: 聚焦判定, 聚焦范围 r.

3.5. 渐进式聚焦策略

对于我们的聚焦视图，聚焦范围越小，越能关注更多的细节信息。基于这一点，我们提出了渐进式聚焦策略（Progressive Focus Strategy，PFS），逐渐地关注需要被更多修复的区域。PFS 与传统的多尺度方式不同，其尺度是根据之前和当前区块预测的变化而动态变化。渐进式聚焦视图中，每次获得新的预测，其相应部分将被用作下一次的输入。我们在 Algorithm 3中展示出渐进式聚焦的算法图。其中，T 的默认值设置为 3，$\hat{\omega}$ 设置为 1.1，ε 设置为 2。

Algorithm 3 渐进式聚焦策略

Input: 之前的区块预测 $\widehat{\mathbf{P}}'$,
区块中心点 $\widehat{\mathbf{p}}$ 在聚焦视角中, 常数 $T, \hat{\omega}, \varepsilon$;

for $t = 1, 2, \cdots, T$ and $\hat{r} \neq 0$ **do**
  生成新的区块预测 $\widehat{\mathbf{P}} = \texttt{Network}(\widehat{\mathbf{P}}')$;
  预测的变化区域 $\Delta\widehat{\mathbf{P}} = |\widehat{\mathbf{P}} - \widehat{\mathbf{P}}'|$;
  生成区域 $\widehat{\mathbf{A}}$ 通过腐蚀 ε 像素数在 $\Delta\widehat{\mathbf{P}}$;
  if $\sum\widehat{\mathbf{A}} > 0$ **then**
    $\tilde{r} = \max_{\forall\{\mathbf{a}|\widehat{\mathbf{A}}_\mathbf{a}=1\}} \eta(\widehat{\mathbf{p}},\mathbf{a})$;
    生成 $\hat{r}$ 通过余量系数, $\hat{r} = \hat{\omega} \cdot \tilde{r}$;
    更新之前的预测结果 $\widehat{\mathbf{P}}' \leftarrow \widehat{\mathbf{P}}$;
    裁剪新的区块根据 $\hat{r}$;
  else
    $\hat{r} = 0$;
  end if
end for

Output: 最终区块预测 $\widehat{\mathbf{P}}$.

标准的 PFS 需要迭代地使用当前预测来修复下一个区块。在多个迭代过程中不能实现并行操作。因此，我们也提出了一个快速版本来缓解这个问题，在略微牺牲性能的情况下提升速度。对于每一个轮次，我们使用之前聚焦范围的 0.8 倍作为当前的聚焦范围。与此同时，裁剪的区块的之前预测来自原始的全局预测。在这个方法中，三个轮次可以并行执行，加速计算过程。

4. 实验

4.1. 设置

数据集. 我们在实验中采用以下被广泛使用的数据集：

- **GrabCut [40]:** 该数据集包含 50 张背景和前景存在明显差异的图片。
- **Berkeley [37]:** 该数据集包含 96 张带有 100 个对象掩膜的图片，其中的一些样本对交互式图像分割任务是有挑战性的。
- **SBD [15]:** 该数据集包含 8498 张用于训练的图片和 2857 张用于测试的图片。在本文中，我们在训练集上训练模型，在包含 6671 张图片的验证集上测试模型。
- **DAVIS [39]:** 该数据集起初用于视频图像分割，包含 50 个视频。遵循之前的工作 [9,20,41]，我们将相同的 345 个具有高质量掩膜的帧用于测试。

#	候选项	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

表 1. FocusCut 的核心消融研究。我们使用指标 'NoC@90' 和 'IoU&5' 来评价整个目标的分割，使用指标 'ASSD&5' 和 'BIoU&5' 来评价细节的分割。↑ and ↓ 分别表示指标值越大模型性能越好和指标值越小模型性能越好。实验分别采用 ResNet-50 和 ResNet-101 作为骨干网络，结果如此表格所示。

评价指标. 遵循先前工作 [9,20,23,26,28,30,31,35,41,51]，我们采样相同的机器人用户来模拟点击，即通过对比标注结果和预测，下个交互点将被置于最大误差区域的中心。我们采用交互点数目（Number of Click，NoC）作为评价指标，该指标记录达到一个固定交并比（IoU）所需的交互点数目。我们把目标 IoU 设置为 85 和 90，分别表示为 NoC@85 和 NoC@90。每个实例的默认最大点击次数限制为 20，并且还会报告无法达到目标 IoU 的失败次数 (Number of Failure，NoF)。我们使用第 N 次点击时的 IoU 指标来表示分割质量，还采用 IoU-NoC 曲线来表示交互后期阶段的收敛趋势。因为我们方法对细节优化更有作用，我们还详细介绍了两个评价指标。边界交并比（boundary IoU，BIoU）[10] 关注靠近对象边界的交并比指标。旨在评价预测边界和标注边界相似性的平均对称表面距离（Average Symmetric Surface Distance，ASSD）也被用于交互式医学图像分割 [45]。对于这两个指标，我们也采用第 N 次点击时的指标值，即 'BIoU&20' 和 'ASSD&20'，来评价模型的性能。IoU 和 BIoU 的指标值越大，表明模型的性能越好，NoC 和 ASSD 则与之相反。

实现细节. 在 ImageNet [12] 上预训练的 ResNet [16] 被用作特征提取器。训练批次大小为 8，轮次数目为 40。对于每个训练轮次，采用初始学习率为 7×10^{-3} 和 γ 值为 0.9 的指数学习率衰减策略。我们采用 momentum 值为 0.9 和权重衰减值为 5×10^{-4} 的随机梯度下降优化模型的参数。我们使用 384 个像素的随机翻转和裁剪来增强数据。对于全局视角的标注模拟，我们遵循 Lin 等人 [31] 使用的策略。从初始点击开始，Zoom-In 策略 [41] 也被应用到推理阶段。本文实验是在 NVIDIA Titan XP 的一个 GPU 上使用 PyTorch [43] 实现的。

速度分析. 我们的方法由一个共享网络的两个分支组成，推理时间是便于计算的。我们用 '1×' 表示这个网络的速度。当引入聚焦视角时，由于交互点可以被并行计算，因此速度是 '2×'。当引入渐进式聚焦策略时，如果采用默认的 T 值，则标准版本的速度为 '4×'；而快速版本的所有轮次依然可以并行计算，故速度依然为 '2×'。对于不同分辨率的图像，输入将始终调整为固定长度作为短边。在固定长度为 384 个像素的环境中，ResNet-50 和 ResNet-101 的 '1×' 速度分别为每次点击 0.0295 秒和 0.0346 秒（Seconds Per Click，SPC）。即便是标准版本，推理速度也足以满足实际应用的需要。

4.2. 消融分析

我们开展核心的消融研究来证明 FocusCut 中采用的各个组件的必要性，实验结果如 Tab. 1所示。由于 Berkeley 数据集规模较大并且与 GrabCut 数据集相似，而 SBD 数据集的标注质量较差，因此我们选择在 Berkeley 和 DAVIS 数据集开展消融实验。我们在这些实验使用了四个指标，其中前两个用于评价整个对象分割，后两个用于细节分割。对于渐进式聚焦策略，我们还针对不同的轮次和设置开展消融实验。

引入聚焦视角. 对于引入焦点视图的核心部分，无论是对整个对象还是细节，性能的提升都是显著的。作为核心的指标，NoC 在 Berkeley 和 DAVIS 数据集中大致较少了一次点击。我们在第 5 次点击时对比其他三个指标。IoU 指标的提升表明聚焦视图带来了一个更完整的对象。BIoU 的增加和 ASSD 的减少表明此方法明显改善了细节，并提供了一个更精确的边界。无论是 ResNet-50 / ResNet-101，还是某个指标，性能的提升都是明显的。因此，聚焦视角的引入无疑是有用的。

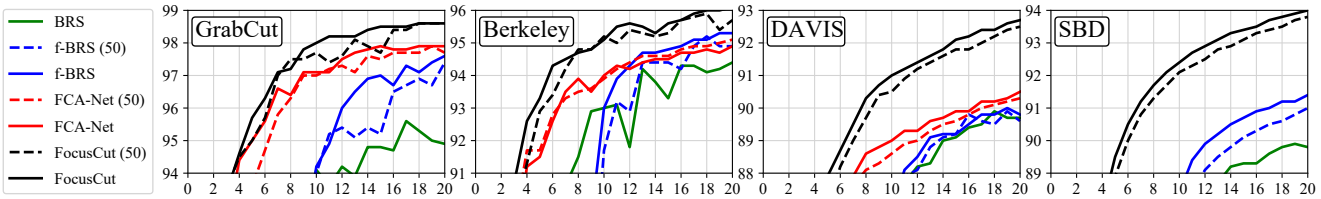

图 4. 代表四个数据集中收敛趋势的局部 IoU-NoC 曲线。‘(50)’ 表示采用 ResNet-50 作为骨干网络。

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

表 2. 在 4 个测评数据集中不同方法的 NoC 指标对比结果。符号 † 表示将 SBD [15] 数据集用于模型训练。§ 表示将增强的 PASCAL VOC [13, 15] 数据集用于模型训练。* 表示 FocusCut 的快速版本。

渐进式聚焦策略. 如 Tab. 1所示，交互式聚焦策略能够帮助我们的方法，并且进一步提高其性能。在标准版本中，之前预测的通道会根据最后一轮的输出进行迭代更新。Tab. 3 展示了不使用迭代预测的实验结果。可以发现，在这种情况下，模型的性能会有一定程度的下降。在 Fig. 5中，我们还展示了使用该策略时，不同轮次的 NoC@90 值。前几个轮次的性能提升很明显，而后几轮的性能提升则因区块尺寸太小而出现波动。由于迭代预测和分步确定聚焦范围的操作需要根据之前的结果进行，因此无法在设备上并行实现。该策略的标准版本可能会牺牲一定的速度，所以我们也提供了一个快速版本，如 Tab. 2所示，其中聚焦范围的缩减因子设置为常数。这个快速版本方法可以在达到出色性能的同时，节省更新预测所用的时间，用户可以根据自己的需求和环境选择自己想要的版本。

性能评价. 基于最常用的指标 NoC，我们的方法和其他方法的对比结果如 Tab. 2 所示。同其他方法一样，我们的方法在 GrabCut、Berkeley、SBD 和 DAVIS 数据集都进行评估。此表中提供了 ResNet-50 和 ResNet-101 的所有性能。可以发现，我们提出的方法在所有数据集中都取得了最先进的性能。此外，FocusCut 快速版本的性能略差于标准版本，但与其他方法相比，仍然具备良好表现。值得注意的是，基准网络中几乎没有插入任何参数或模块，这强烈反映了 FocusCut 的有效性。为了反映收敛趋势，我们裁剪并放大了 IoU-NoC 曲线，并将它们显示在 Fig. 4 中。在该图中，我们选择了一些最近的具有开源代码的方法。此外，由于 FCA-Net 使用增强 PASCAL 进行训练，故它不在 SBD 子图中。可以发现，在交互的较后阶段，我们的方法还是有一定的上升趋势。第 20 次点击的结果表明我们的方法具有更高的上限，这反映了它可以更精细地分割对象。

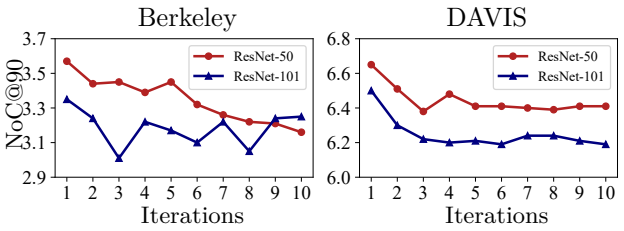

图 5. 渐进式聚焦策略的 NoC@90 *vs.* 迭代次数图。

设置	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

表 3. 有无渐进式预测（iterative prediction，IP）情况下的 NoC@90 指标对比。

方法	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

表 4. 基于开源代码，不同方法在第 20 次点击时的详细指标对比。最后四个模型基于 ResNet-101。LD 是 latent diversity [26] 的缩写。

分割质量. Fig. 6展示了 FocusCut 可以发挥主导作用的一些情况。例如，在飞机轮子等小部件的位置，FocusCut 只需在前景中提供一个交互点即可生成准确的预测。在一些有很多缝隙的地方，比如图片中小狗的腿之间或者人的手指之间的区域，虽然提供了背景点，但是神经网络很可能会从全局角度过度压制它们的作用，而我们的 FocusCut 可以很好地处理这种情况。与之前方法类似 [9, 20, 41]，我们还在 Tab. 5中展示了最近方法的 NoF 指标。最多 100 次点击的结果可以反映细分细节的性能。无论度量标准是 NoF 还是 NoC，我们都超越了所有最新的方法，并取得了目前最先进的性能。此外，我们还将我们的方法与其他 BioU 和 ASSD 指标代码已开源的方法进行了比较，实验结果如 Tab. 4所示。显然，我们的方法在 ASSD 和 BIoU 中都优于其他方法，显示了我们的方法在细节优化方面的有效性。在实际使用中，我们在用户界面上提供了一个小窗口作为放大镜来显示鼠标附近的区域，帮助用户在小范围内点击更准确的位置。

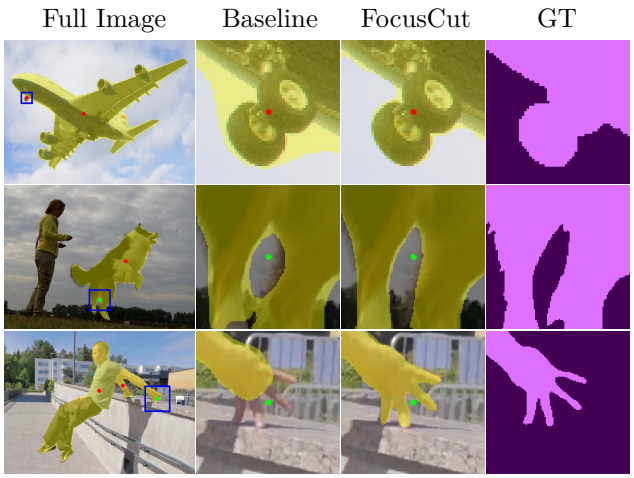

图 6. FocusCut 的分割结果及其同基准模型的对比。预测结果和交互点如上图所示。

方法	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

表 5. 基于 DAVIS 数据集和 ResNet-50，不同交互点设置情况下的对比。NoF_N@90 表示在 N 次点击下 IoU 未能达到 0.9 的失败图像数。NoC_{100}@90 指标与 NoC@90 具有相同的最大交互点数，即 100。

限制分析. 我们的方法需要多次运行分割，推理时间将不可避免地增加。事实上，即使对于快速版本，计算负担也是相同的。对于一些老设备来说，时间消耗和计算负担可能仍然是一个瓶颈。

5. 结论

在这篇论文中，我们引入了聚焦视角来理解用户新输入的交互点的意图。我们使用一个简单而有效的管道 FocusCut 来协调焦点视图，其中，以交互点为中心裁剪的区块的预测通过一个与全局视角共享的网络进行更新。在多个自适应范围下，区块更新是渐进式的。在 4 个数据集上的大量实验中，我们提出的 FocusCut 取得了目前最优异的性能，证明了该模型的优越性。

致谢: 这项工作得到了国家重点研发计划（NO.2018AAA0100400）、NSFC（NO.61922046）、中国教育部科技创新项目和中国博士后科学基金（NO.2021M701780）的资助。我们也非常感谢 MindSpore、CANN 和 Ascend AI Processor 对本研究的支持。

参考文献

[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