Reviving Iterative Training with Mask Guidance for Interactive Segmentation

Konstantin Sofiiuk, Ilia A. Petrov* and Anton Konushin**

Visual Understanding lab., AI Center Moscow, Samsung Electronics Co., Lesnaya 5C, Moscow, Russia

ARTICLE INFO

Keywords:
interactive segmentation
segmentation
mask refinement

ABSTRACT

Recent works on click-based interactive segmentation have demonstrated state-of-the-art results by using various inference-time optimization schemes. These methods are considerably more computationally expensive compared to feedforward approaches, as they require performing backward passes through a network during inference and are hard to deploy on mobile frameworks that usually support only forward passes. In this paper, we extensively evaluate various design choices for interactive segmentation and discover that new state-of-the-art results can be obtained without any additional optimization schemes. Thus, we propose a simple feedforward model for click-based interactive segmentation that employs the segmentation masks from previous steps. It allows not only to segment an entirely new object, but also to start with an external mask and correct it. When analyzing the performance of models trained on different datasets, we observe that the choice of a training dataset greatly impacts the quality of interactive segmentation. We find that the models trained on a combination of COCO and LVIS with diverse and high-quality annotations show performance superior to all existing models. The code and trained models are available at `https://github.com/saic-vul/ritm_interactive_segmentation`.

1. Introduction

Interactive segmentation algorithms allow users to explicitly control the predictions using interactive input at several iterations, in contrast to common semantic and instance segmentation algorithms that can only input an image and output a segmentation mask in one pass. Such interaction makes it possible to select an object of interest and correct prediction errors. Another important feature of this group of algorithms is the capability to segment objects of previously unseen classes.

User input can be formalized via various representations: scribbles, clicks, extreme points, etc. Click-based interactive segmentation is one of the most well-studied topics among other deep learning-based interactive segmentation approaches, as it has well-established protocols of training and evaluation [1, 2, 3, 4]. It also employs quite an intuitive and simple way to specify the desired object. Scribble-based methods often use heuristic and complicated procedures for simulating user input, which makes it hard to fairly evaluate them [5, 6, 7]. Approaches based on extreme points are not intuitive for users and are not flexible enough due to a limited number of user interactions [8]. In our work, we focus on click-based interactive segmentation.

In general, the development of deep learning-based algorithms for semantic and instance segmentation requires a huge amount of annotated data. Annotating data with segmentation masks is very time-consuming and therefore expensive. Interactive segmentation can significantly simplify and speed up this process [9, 10, 11], which is one of the most important applications of interactive segmentation. It can also be used in photo editing, allowing users to select objects easily, which is especially important for smartphone applications where user input is often limited by finger or stylus activities.

Recent works on click-based interactive segmentation propose complicated inference-time optimization procedures to improve the quality of interactive segmentation even further [2, 3, 4]. The deployment of these models is substantially limited, for instance, on mobile devices, as it requires the implementation of backward passes with gradients that are not provided by popular frameworks. Surprisingly, we find that properly chosen baselines of click-based interactive segmentation models without any without any explicit refinement techniques and other specific modifications can be

*Corresponding author
**Principal corresponding author
k.sofiiuk@samsung.com (K. Sofiiuk); ilia.petrov@samsung.com (I.A. Petrov); a.konushin@samsung.com (A. Konushin)
ORCID(s): 0000-0002-8900-1071 (I.A. Petrov); 0000-0002-6152-0021 (A. Konushin)

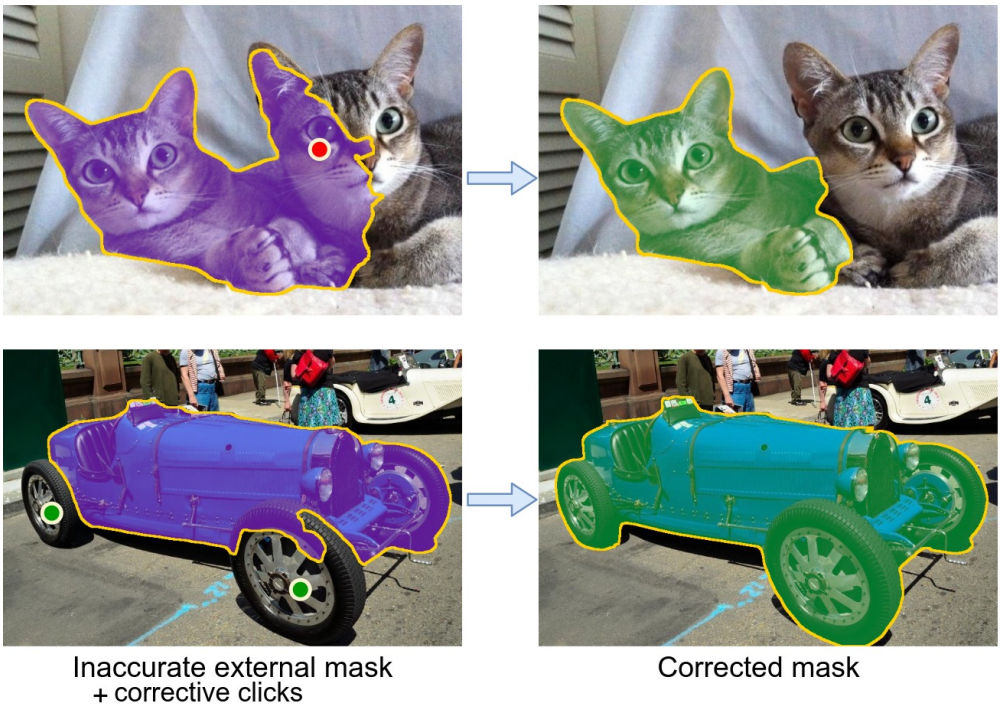

Figure 1: Besides segmenting new objects, proposed method allows to correct external masks, *e.g.*produced by other instance or semantic segmentation models. A user can fix false negative and false positive regions with positive (green) and negative (red) clicks, respectively.

trained using the standard random points sampling procedure and show state-of-the-art performance. We observe that modifications of poorly tuned baselines can provide false performance improvements, while strong baselines with the same modifications do not show any improvement at all. At the same time, datasets with coarse masks or containing a small amount of images may become a bottleneck for the performance of interactive segmentation models. Our experiments show that usage of diverse large datasets with fine masks for training plays a crucial role for the performance of discussed models. Therefore, we choose a strong baseline model for implementing and evaluating new modifications. We train models on a combination of LVIS and COCO datasets [12, 13], that, to the best of our knowledge, are the most suitable for training interactive segmentation models.

We propose an extension of click-based interactive segmentation that allows to modify existing instance segmentation masks interactively. We revive an iterative training procedure, and make a network aware of the mask from a previous step [14]. We show that such awareness improves the models' stability, i.e. allows to avoid accuracy dropping when adding new clicks. We propose a new training dataset obtained by combining the LVIS and COCO datasets.

2. Related Work

Interactive segmentation methods. Interactive segmentation is a longstanding problem in computer vision. Early methods [15, 16, 17, 18] tackle the problem using optimization-based approaches minimizing a specifically constructed cost function defined on a graph over image pixels. GrabCut proposed in [18] is a classic approach based on iterative energy minimization of a cost function, that is modeled using a Gaussian mixture. Xu *et al.* [1] first propose a CNN-based model for interactive segmentation and introduced a clicks simulation strategy for training that is adopted in some further works. Later, [19, 20, 21, 22] propose various CNN-based methods for interactive segmentation that aimed at the refinement of predictions by increasing the diversity of predicted masks, and using the attention mechanism. The novel subgroup of methods emerges with an introduction of the Backpropagating Refinement Scheme (BRS) in [2]. The authors propose an optimization procedure that minimizes a discrepancy between the predicted mask and the map of input clicks after each click with respect to the input distance maps. This refinement technique improves the segmentation quality at the cost of increasing the runtime. Sofiiuk *et al.* [4] address this issue by proposing f-BRS,

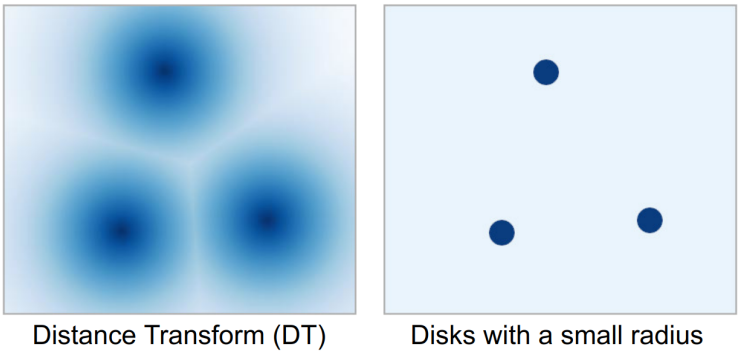

Figure 2: Visualization of two different approaches for encoding user clicks.

a lightweight version of BRS that uses a similar optimization procedure, though with respect to internal parameters introduced to the higher levels of a network, reducing runtime on an order of magnitude. Kontogianni *et al.* [3] introduce another method of test time refinement targeting the optimization process on network parameters.

Different types of interactive feedback. While clicks are the mainstream form of input in interactive segmentation, a lot of works explore other variations of interactive feedback from the user. For example, one of the simplest types of interactions is a bounding box, that is used in [23, 18, 24]. The main drawbacks of these approaches are lack of specific object reference inside the selected area and lack of interface for correction of the predicted mask. The limitations of bounding box-based interaction are addressed in [25]. The authors propose to combine clicks with bounding boxes to provide more specific object guidance and allow corrections of the predicted mask. DEXTR [8] uses extreme points of the target object, i.e. left-most, right-most, top, and bottom pixels as an input. On the one hand, such an interaction is compact and is limited by 4 clicks. On the other hand, placing extreme points in the right locations is harder than making an arbitrary click on an object and there is no support for corrections, similar to interaction via bounding boxes. Scribbles are used in many early works [5, 15, 16, 17, 26]. This type of feedback provides richer prior information for the algorithm, compared to the others. In contrast, putting a stroke requires more effort from the user, compared to simpler forms of interactions. Another drawback that refrains CNN-based methods from the wide adoption of scribbles is that realistic strokes simulation for training the neural networks is a rather hard task. Therefore, the vast majority of scribble-based methods employ graphical models or similar training-free techniques. Apart from the aforementioned forms of interaction, some works use a combination of traditional input representation with other modalities, e.g. PhraseClick [27] combines clicks with text input to better infer the attributes of the target object, thus requiring fewer clicks.

Segmentation mask refinement. The majority of segmentation mask refinement techniques are targeted to refine the boundaries of a mask by making local corrections without changing the mask globally, i.e. excluding or including large parts of an object. These methods can either be a part of the network architecture [28, 29] or serve as a post-processing step [30, 31]. Local and boundary refinement tasks are often addressed without any additional input from the user, as all the required features are automatically extracted from the input image and mask. Global refinement, on the contrary, may require additional feedback from the user to make improvements to the mask. We propose to combine interactive segmentation and mask refinement allowing to refine external segmentation mask with user clicks as guidance.

3. Proposed Method

3.1. Revising Network Architecture

The task of interactive segmentation is very similar to instance or semantic segmentation in terms of the network architecture. In these tasks, networks take high-resolution input and produce high-resolution segmentation masks aligned with the input. The key difference is in the user input: its main aspects are the encoding and processing of the encoded input inside the network. Therefore, there is no need to reinvent general segmentation architectures. Instead, it is reasonable to rely on time-tested state-of-the-art segmentation networks and focus on interactive-specific parts.

We consider the DeepLabV3+ [32] and HRNet+OCR [33, 34] semantic segmentation architectures as a backbone

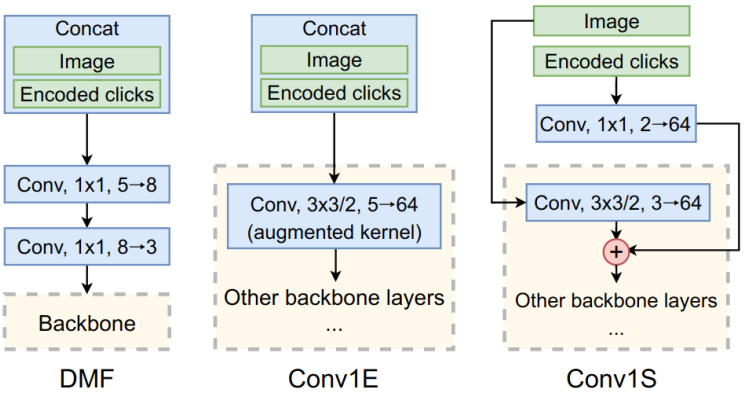

Figure 3: Different architecture choices of feeding encoded clicks to a backbone, described in Section 3.1.

for our interactive segmentation model. While DeepLabV3+ is a well-studied segmentation architecture that proved efficient in many segmentation-related tasks, HRNet is a relatively new promising one that was specially designed to produce high-resolution output. According to our experiments, HRNet is a more preferable architecture for this task. We provide the ablation study for the backbone architecture in Section 5.2 and show the results in Tables 1, 3 and 4.

Clicks encoding. There are positive and negative clicks in click-based interactive segmentation. These clicks are represented by their coordinates in an image. In order to feed them to a convolutional network, we should encode them in a spatial form first. Encoding the clicks via a distance transform from Xu *et al.* [1] is the most common approach to clicks encoding. However, they can also be represented by gaussians or disks with a fixed radius. Benenson *et al.* [10] perform a detailed ablation study on clicks encoding and find that disks with a small radius surpass the other encodings in terms of model performance.

We conduct our ablation study to compare the distance transform encoding with the disks encoding and find that the latter outperforms the former. We assume that the following observation explains the superiority of disks. The changes in disk encoding caused by adding new points or moving existing ones are always local and only slightly affect the encoding map. At the same time, a distance transform map can change drastically when a new point is added, especially if there are only a few points. In turn, such sudden considerable changes might confuse a network. We show a visualization of both the distance transform and disk encodings in Figure 2.

Feeding encoded clicks to a backbone. Semantic segmentation backbones are usually pre-trained on ImageNet [35] and can take only RGB images as an input. The most common way to handle additional input, e.g. encoded user clicks, is to augment the weights of the first convolutional layer of a pre-trained model to accept N-channel input instead of only an RGB image [1, 10]. In our work, we denote this modification of network architecture by Conv1E. Sofiiuk *et al.* [4] propose the Distance Maps Fusion (DMF) module which transforms an image concatenated with additional user input channels into 3-channel input.

We propose a new simple approach to solving this problem, which is described as follows. We introduce a convolutional block that outputs the tensor of exactly the same shape as the first convolutional block in the backbone does. This tensor is then summed element-wise with the output of the first backbone convolutional layer, which usually has 64 channels. We denote this modification of network architecture by Conv1S. While being similar to Conv1E, it allows to choose a different learning rate for new weights without affecting the weights of a pre-trained backbone. We show the schemes of these architectural modifications in Figure 3 and provide an ablation study in Section 5.2.

3.2. Iterative Sampling Strategy

Most recent works on click-based interactive segmentation use the sampling strategy for simulating user clicks during training, where a set of positive and negative clicks is randomly generated without considering any relations between them [4, 22, 10, 1]. In practice, every new click is placed in the erroneous region of a prediction produced by a network using the set of previous clicks. This fact is completely ignored in the random sampling strategy. It also makes it impossible to integrate masks from previous interactions into a model, as we need to have ordered interactions and a sequence of corresponding predictions to successfully train such a model.

The iterative sampling strategy, which resembles the interaction with a real user, is employed to a certain extent

Table 1
Ablation studies of the network architecture choices described in Section 3.1. Each cell consists of two results "X/Y", where "X" and "Y" correspond to evaluation without and with f-BRS-B[4], respectively. "DT" stands for the distance transform clicks encoding. All models are trained on SBD.

Backbone	Input Scheme	Clicks Encoding	NoC_{20}@90	
			Berkeley	DAVIS
ResNet-34	DMF [4]	DT	5.50/4.32	8.45/8.34
	Conv1E	DT	4.79/4.43	7.56/7.60
	Conv1S	DT	4.98/4.16	7.41/7.28
	Conv1S	Disk3	4.52/4.04	7.27/7.18
	Conv1S	Disk5	4.09/3.89	6.92/7.22
HRNet-18	DMF [4]	DT	4.93/4.35	8.59/8.00
	Conv1E	DT	4.41/3.95	7.50/7.43
	Conv1S	DT	3.99/3.81	7.16/7.24
	Conv1S	Disk3	3.63/3.47	7.14/7.04
	Conv1S	Disk5	3.52/3.50	6.90/6.97

in several works [14, 36, 3]. As full iterative sampling is very computationally expensive, in these works random sampling is used for initialization and then a few clicks are added using the iterative sampling procedure.

We adopt the iterative sampling procedure proposed by Mahadevan *et al.* [14] with the following changes. First, we sample each point not just from the center of a mislabelled region, but from the region obtained by applying morphological erosion operation to the mislabelled region, so that the eroded region has 4 times less area. We observe that choosing center points leads to overfitting to the NoC evaluation metric (see Section 5 for the details). In practice, however, these models demonstrate worse performance and unstable behavior when a user places a click near the borders of an object or a mislabelled region. Second, we do not save simulated clicks for dataset samples during training and simulate user clicks for each batch individually. For that reason, we limit the maximum number of sampling iterations to N_{iters}, and each batch can uniformly get from 0 to N_{iters} iterations.

We use a combination of the random and iterative sampling strategies for training our iterative models. First, we simulate user clicks with the random sampling strategy [1] just as we do for our non-iterative baselines. Then, we add from 0 to N_{iters} iteratively simulated clicks. We provide an ablation study of the number of iteratively sampled clicks in Section 5.2.

3.3. Incorporating Masks From Previous Steps

In interactive segmentation, it seems natural to incorporate output segmentation masks from previous interactions as an input for the next correction, providing additional prior information that can help improve the quality of prediction. In case of incorporating the mask from a previous interaction, it is necessary to use the iterative sampling for simulating user interactions during training. Along with the iterative sampling, Mahadevan *et al.* [14] propose passing an output mask from a previous iteration into the model as an optional extension of their method.

To train a model with the mask from a previous step, we use a combination of the random and iterative user interactions simulation described in Section 3.2. Our model takes this mask as the third channel together with two channels for positive and negative encoded clicks, respectively. For the first interaction, as well as for the batches with skipped iterative sampling, we feed an empty mask to our model.

3.4. Normalized Focal Loss

Binary cross entropy (BCE) loss is one of the standard loss functions for training semantic segmentation algorithms. Besides, some state-of-the-art interactive segmentation methods [2, 21, 36, 8, 25] adopt it for training the network. The main drawback of BCE is that it treats all examples equally, slowing the training during later epochs as the gradient from almost correctly segmented areas propagates similarly to the gradient from erroneous regions. Focal loss was introduced in [37] to alleviate this problem. Let $\hat{M}$ denote the output of the network and $p_{i,j}$ denote the confidence of prediction at the point (i, j). Then the FL is formulated as follows:

$$FL(i,j) = -(1 - p_{i,j})^{\gamma} \log p_{i,j} \tag{1}$$

One can notice that the total weight $P(\hat{M}) = \sum_{i,j}(1-p_{i,j})^{\gamma}$ decreases when the accuracy of the prediction increases.

Table 2
Evaluation of the non-iterative baseline method with the HRNet-18+OCR backbone trained on COCO+LVIS (proposed in Section 4.2) with different loss functions. We report NoC@90 on four datasets.

Method	NoC_{20}@90			
	GrabCut	Berkeley	SBD	DAVIS
BCE	1.82	3.13	7.58	6.31
Soft IoU	2.02	3.03	7.94	6.45
FL	1.80	3.28	7.56	6.40
NFL	**1.70**	**2.48**	**6.72**	**5.90**

This means that the total gradient of the FL fades over time, slowing the training process. To mitigate this problem, [38] proposed normalized focal loss (NFL) that is formulated as follows:

$$NFL(i,j,\hat{M}) = -\frac{1}{P(\hat{M})}(1-p_{i,j})^{\gamma}\log p_{i,j} \tag{2}$$

The gradient of NFL does not fade over time due to normalization and remains equal to the total gradient of BCE. This allows for faster convergence and better accuracy compared to training with BCE. We choose NFL as a loss function for training the proposed methods and provide an ablation study with respect to loss function choice in Section 5.2 (results are provided in Table 2).

4. Dataset for Interactive Segmentation

4.1. Reviewing Existing Datasets

The vast majority of recent CNN-based interactive segmentation methods [2, 19, 20, 21, 36, 4, 1] are trained either using Semantic Boundaries Dataset [39], or the Pascal VOC dataset [40], or the combination of these two datasets, as they share the same set of images. The augmented dataset contains a total of 10582 images with 25832 instance-level masks. The total number of classes in this dataset is twenty: 7 categories of transportation, 6 species of animals, 6 types of indoor objects, and a separate class for persons. These classes cover only general types of objects, implying limitations on the variety of predictable classes. Recently, large-scale segmentation datasets OpenImages [10, 41] and LVIS [13] were introduced. These datasets stimulated the development of state-of-the-art instance segmentation algorithms [42, 43, 44, 45], providing a large variety of labeled classes with a sufficient number of examples. LVIS contains around 1.2M instances on 100k images embracing more than a thousand object classes, while OpenImages has 2.6M instances on 944k images covering 350 object categories. We believe that having such a diverse dataset is one of the key components in training of the state-of-the-art interactive segmentation model. Another important characteristic of a segmentation dataset is annotation quality. Gupta *et al.* [13] present a study of annotation quality in some of the instance segmentation benchmarks, comparing a subset of masks from each dataset to experts' annotations. According to this study, the LVIS dataset has the highest annotation quality among reported datasets, presumably allowing to achieve higher prediction quality.

4.2. Combination of COCO and LVIS

Considering the observations from Section 4.1, the LVIS dataset appears to be the best choice for training the models except for one drawback: it is long-tailed and therefore lacking general object categories, which can affect the accuracy and generalization of the trained model. As a solution to this problem, we propose to augment LVIS labels with masks from the COCO [12] dataset, as these two datasets share the same set of images. The COCO segmentation dataset contains a total of 1.2M instance masks on 118k training images with 80 object classes. These categories represent more common and general objects, complementing the long-tailed object classes from the LVIS dataset. We propose the following procedure to construct the combined COCO+LVIS dataset with diverse object classes from the most common to less frequent ones. All masks from both datasets are joined together except for those masks from COCO that have a corresponding mask from LVIS with an intersection over union (IoU) score between them larger than 80%. In that case, we only keep the mask from LVIS, since it presumably has better overall and especially boundary quality. As a result of the described procedure, we obtained a dataset with 104k images and 1.6M instance-level masks.

Table 3
Evaluation results for the non-iterative baseline method with the ResNet-34 and HRNet-18+OCR backbones trained on various datasets. NoC@90 is reported.

Train Dataset	Backbone	NoC_{20}@90			
		GrabCut	Berkeley	SBD	DAVIS
ADE20k	ResNet-34	2.70	5.09	8.27	8.56
	HRNet-18	2.68	4.78	8.02	8.32
OpenImages	ResNet-34	2.20	4.68	8.16	7.47
	HRNet-18	2.02	4.47	7.95	8.08
SBD	ResNet-34	2.94	4.73	6.94	7.56
	HRNet-18	2.41	3.95	6.66	7.17
Pascal VOC+SBD	ResNet-34	2.77	4.53	6.92	6.48
	HRNet-18	2.25	3.63	6.63	6.16
LVIS	ResNet-34	2.59	3.61	7.99	6.98
	HRNet-18	2.44	3.13	8.14	7.18
COCO	ResNet-34	1.80	3.34	6.29	6.11
	HRNet-18	1.77	2.90	6.32	5.85
COCO+LVIS	ResNet-34	1.74	2.91	6.53	6.01
	HRNet-18	1.70	2.48	6.86	6.00

We argue that the further development of interactive segmentation algorithms relies heavily on the training data. Thus, we provide the comparative study of the training datasets in Section 5.2, results are provided in Table 3. Based on our experiments and the aforementioned observations, we conclude that the proposed COCO+LVIS dataset is the best choice for training interactive segmentation methods.

5. Experiments

We perform an extensive evaluation of the proposed approach by conducting ablation studies for all sufficient parts of the method, exploring the convergence properties with an increasing number of clicks and comparing our method with current state-of-the-art works.

Datasets. We evaluate the performance of our method on five common benchmarks for interactive segmentation with instance-level annotations. The GrabCut [18] dataset contains 50 images with a single object mask for each image. We adopt the test subset of Berkeley [46] introduced in [47], which consists of 100 masks for 96 images. The DAVIS [48] dataset was introduced for the evaluation of video segmentation datasets. We use the subset of 345 randomly sampled frames of video sequences that was introduced in [2] for evaluation. We follow the common protocol and combine all instance-level masks for one image into one segmentation mask. We also use the validation part of the Pascal VOC [40] dataset that consists of 1449 images with 3417 instances. Each instance mask is used separately during evaluation. The Semantic Boundaries Dataset (SBD) [39] contains 6671 instance-level masks for 2820 images. This dataset has been used for evaluating the interactive segmentation algorithms since [1]. To test the algorithm, we use each of instance masks separately and do not combine them into one segmentation mask for one image.

Evaluation metric. We perform the evaluation using the standard Number of Clicks (NoC) measure, reporting the number of clicks required to achieve the predefined Intersection over Union (IoU) threshold between predicted and ground truth masks. We denote NoC with IoU threshold set to 85% and 90% as NoC@85 and NoC@90, respectively. To generate clicks during the evaluation procedure, we follow the strategy used in [19, 1]. The next click is placed at the center of the region with any type of prediction error (false positive or false negative) with the largest area among other erroneous regions. The region center is defined as the point farthest from the boundaries of the corresponding region.

Segmentation backbones. We consider two different segmentation architectures: DeepLab-V3+ [32] with ResNet and HRNet+OCR [33, 34]. To implement DeepLab-V3+ for interactive segmentation we follow [4]. We use the implementation of HRNet and models pre-trained on ImageNet presented in the official repository[1]. There are several versions of HRNet which differ in terms of their capacity: HRNet-W18-C, HRNet-W30-C, HRNet-W32-C, etc. In our work we employ HRNet-W18-C-Small-v2, HRNet-W18-C, HRNet-W32-C, referring to them as HRNet-18s, HRNet-

[1] https://github.com/HRNet/HRNet-Image-Classification

Table 4
Comparison of different models in terms of the number of FLOPs and parameters. All models take an image with a resolution of 400×400 to compute the number of FLOPs. We use exactly the same architecture of the DeepLab models as it was proposed in [4].

Model	#Params	Ratio-to-HRNet18	#FLOPs	Ratio-to-HRNet18
HRNet18s+OCR	4.22M	0.4x	17.84G	0.6x
HRNet18+OCR	10.03M	1.0x	30.80G	1.0x
HRNet32+OCR	30.95M	3.1x	82.84G	2.7x
DeepLab-ResNet-34	19.17M	1.9x	122.28G	4.0x
DeepLab-ResNet-50	31.40M	3.1x	170.13G	5.5x

18, HRNet-32 for the sake of brevity. The numbers of parameters and FLOPs of our models is shown in Table 4. All the networks are initialized by the weights pre-trained on ImageNet weights.

We use the OCR module proposed in [34] for all HRNets. The original OCR module always produces 512 output channels regardless of the size of a backbone. We proportionally reduce the width of all the OCR layers, so that it produces 48, 64 and 128 channels for HRNet-18s, HRNet-18 and HRNet-32, respectively.

Implementation details. The training task is binary segmentation with normalized focal loss, described in Section 3.4, as an objective. We use image crops with a size of 320×480 and randomly resize images with a scale factor from 0.75 to 1.40 before cropping. We use horizontal flip and random jittering of brightness, contrast, and RGB values as augmentations during training. We adopt test time augmentations from f-BRS method [4], using Zoom-In technique and averaging predictions from original and horizontally flipped image during evaluation.

In all our experiments, we use Adam optimizer with $\beta_1 = 0.9, \beta_2 = 0.999$ and train the models for 55 epochs on the proposed COCO+LVIS dataset (49 epochs with learning rate 5×10^{-4}, then learning rate is decreased by 10 times on 50th and 53rd epochs). The learning rate for backbone networks is set 10 times lower than the learning rate for the rest of the network. We denote one pass through each image of the dataset as an epoch. The batch size is set to 32. We train the networks based on HRNet-18s, HRNet-18 and HRNet-32 on 1, 2 and 4 Tesla P40, respectively. To train the model based on ResNet-34, we use 1 Tesla P40.

We implement the proposed method using the PyTorch [49] framework.

5.1. Convergence Analysis

One of the key properties of an interactive segmentation algorithm is convergence to sufficient accuracy with an increasing number of clicks. Previous works [2, 3, 4] improve the convergence using inference time optimization schemes, that force the predictions to match with input clicks. Our method is free from any refinement schemes and makes use of the proposed architecture and the iterative training scheme to achieve the desired level of convergence.

Sofiiuk *et al.* [4] introduced the evaluation protocol for analysis of convergence for interactive segmentation methods. The main aspect of this protocol is the NoC_{100} evaluation metric, which is similar to the NoC described earlier but has the clicks limit set to 100. Such size of interactive feedback should give an algorithm enough information to converge. In case the algorithm can not reach the IoU threshold, the image is supposedly too hard for the method to handle, therefore increasing the click limit is unlikely to change it.

Table 5 shows the comparison of the the results of BRS [2], f-BRS-B [4] and the proposed method with HRNet18+OCR backbone, reporting NoC_{100}, number of images for which the 90% threshold on IoU was not achieved after 20 and 100 clicks. All models are trained on the SBD [39] dataset for a fair comparison.

5.2. Ablation Studies

Network architecture ablations. In Section 3.1, we discuss different architectural choices for interactive segmentation networks. First, we explore different strategies of feeding encoded clicks to the model (DMF, Conv1E and Conv1S). We find that the HRNet-18 and ResNet-34 models with Conv1S show better performance in general and we use it in all further experiments. Then, we evaluate different clicks encoding strategies. We find that changing the distance transform encoding to the disk encoding significantly improves results of both HRNet-18 and ResNet-34, which confirms the findings of Benenson *et al.* [10] whose model performed best with disks with a radius of 3. However, our experiments show that a radius of 5 is better than 3. The results of all ablations can be found in Table 1.

Table 5
Convergence analysis on Berkeley, SBD and DAVIS. We report the number of images that were not correctly segmented after 20 and 100 clicks and the NoC_{100}@90 metric. All models are trained on SBD.

Dataset	Model	#images ≥20	#images ≥100	NoC_{100} @90
Berkeley	BRS [2]	10	2	8.77
	f-BRS-B [4]	**2**	**0**	**4.47**
	HRNet-18	7	3	7.10
	HRNet-18 ITER-M	3	2	4.89
DAVIS	BRS [2]	77	51	20.98
	f-BRS-B [4]	78	50	20.70
	HRNet-18	67	50	19.83
	HRNet-18 ITER-M	**57**	**44**	**18.42**
SBD	f-BRS-B [4]	1466	265	14.98
	HRNet-18	1051	450	13.62
	HRNet-18 ITER-M	**671**	**215**	**9.52**

Table 6
Ablation studies on the maximum number of iterations N_{iters} in the iterative sampling. All the results are reported for the model with the HRNet-18+OCR backbone trained on COCO+LVIS.

N_{iters}	Prev Mask	NoC_{20}@90		
		Berkeley	DAVIS	SBD
3	–	2.38	5.92	6.49
3	+	2.26	5.74	6.06
1	+	2.57	5.81	6.15
2	+	2.48	**5.70**	6.10
3	+	**2.26**	5.74	6.06
4	+	2.52	6.03	**6.04**
5	+	2.49	5.98	6.24
6	+	2.55	6.11	6.82

In addition, these experiments clearly demonstrate the superiority of HRNet-18 over DeepLabV3+ with ResNet-34, especially when the former has several times fewer parameters and FLOPs than the latter (the number of parameters and FLOPs can be found in Table 4). We use HRNet with the Conv1S input scheme and encode clicks with disks of radius 5 in all other experiments.

Training datasets comparison. To further research the findings from Section 4, we evaluate the models trained on each of the six common segmentation datasets: ADE20k [50], OpenImages [10, 41], SBD [39], Pascal VOC [40] augmented with labels from SBD (denoted as Pascal VOC+SBD), LVIS [13], COCO [12] and also the COCO+LVIS dataset, proposed in Section 4.2. We provide results for non-iterative baseline methods with HRNet-18+OCR and ResNet-34 backbones, trained with NFL loss. We report NoC@90 on Berkeley, SBD and DAVIS in Table 3. The performance on COCO and COCO+LVIS is relatively close, but the model trained on the proposed dataset generalizes better due to wider class distribution in the training set. Another observation is that the models trained on SBD and Pascal VOC+SBD show the best performance on the SBD dataset in terms of NoC@90, which is most likely caused by the similar distribution of training and testing sets. Nonetheless, these models are inferior to the model trained on the COCO+LVIS dataset in terms of performance on other benchmarks. All models trained for dataset comparison share the same training parameters and augmentations, that are described at the beginning of Section 5. We train the models for the following number of epochs on each dataset: 180 for ADE20k, 5 for OpenImages, 120 for SBD and Pascal VOC+SBD, 55 for LVIS and COCO+LVIS, 40 for COCO. We denote one pass through each image of the dataset as an epoch.

Loss functions comparison. We compare four loss functions that were used for training segmentation methods in recent works: binary cross entropy (BCE) loss, focal loss (FL) [37], soft IoU loss [51] and normalized focal loss (NFL) [38]. We evaluate the performance of the baseline HRNet-18+OCR model trained on the COCO+LVIS dataset, described in Section 4.2. The results are presented in Table 2. We provide only NoC@90 on all standard datasets for simplicity. The evaluation results support the reasoning in Section 3.4, demonstrating that training with NFL leads to

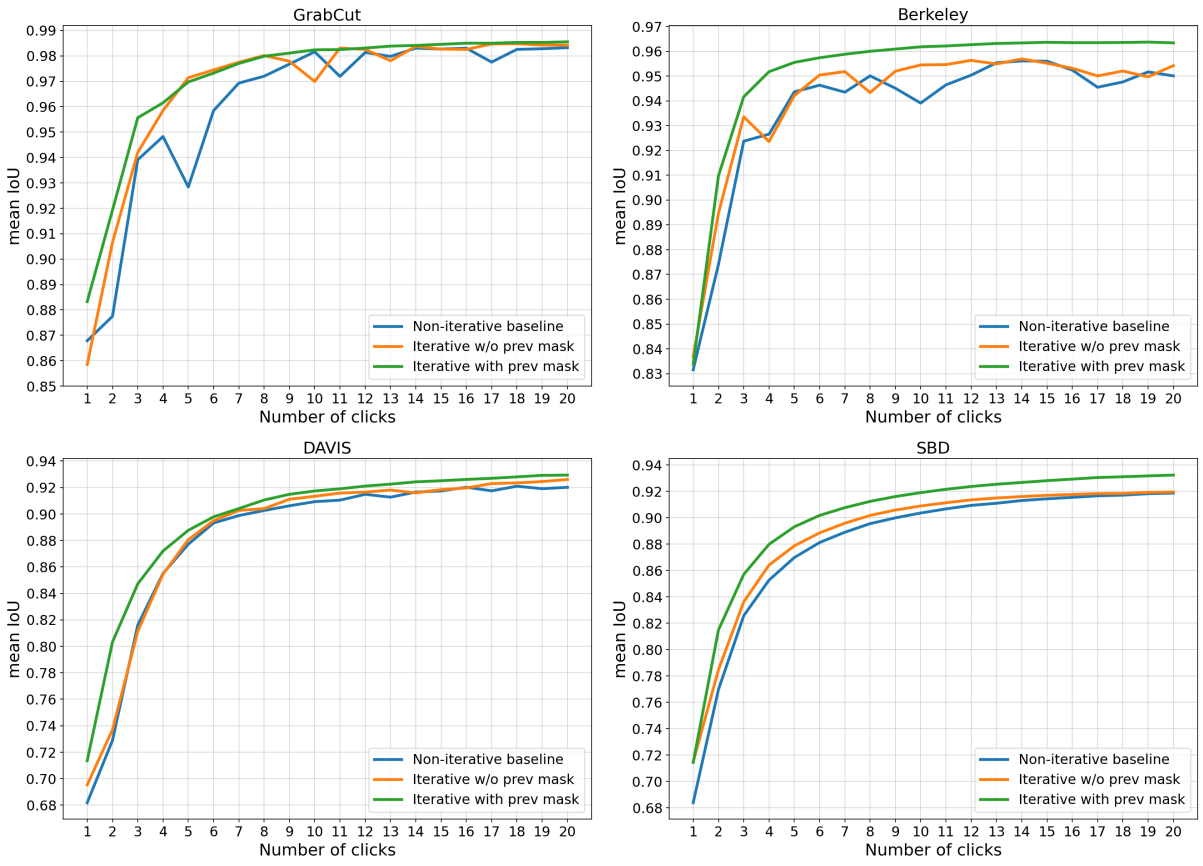

Figure 4: Mean IoU@k for varying number of clicks k on GrabCut, Berkeley, DAVIS and SBD. The iterative model that takes a mask from a previous step is much more stable and converges to a better IoU. All the results are reported for the model with the HRNet-18+OCR backbone trained on COCO+LVIS, iterative models are trained with $N_{iters} = 3$.

better accuracy and convergence on all 4 datasets.

Iterative training ablations. The main hyperparameter of our iterative sampling scheme is the maximum number of iteratively sampled clicks N_{iters} in addition to some set of randomly sampled clicks. Our experiments show that $N_{iters} = 3$ is an optimal value of that parameter. Surprisingly, too high values (> 4) lead to instability during training and to worse results. We had to train models with $N_{iters} = 5, 6$ for several times, as they collapsed after 10-20 epochs of training on COCO+LVIS and showed poor results. We provide the ablation study in Table 6.

We study the impact of feeding a mask from the previous click to the iterative model with $N_{iters} = 3$. Apart from metrics improvement, we observe substantial improvement of the stability of the model when new clicks are added. As shown in Figure 4, both the iterative model without a previous mask and the non-iterative model sometimes have drops of mean IoU when the number of clicks increases. It indicates that adding a new click during the process of interactive segmentation can even make the mask worse, contradicting user expectations. This effect is better illustrated on mean IoU plots for small datasets such as GrabCut or Berkeley, as the smoothness of the curve is proportional to the dataset size. When the model relies on a mask from a previous iteration, it can take into account the segmentation result of a previous iteration and avoid unexpected collapse of the current mask. Moreover, it allows to modify existing masks without additional effort. We discover that the trained model can be successfully initialized with an external inaccurate mask without the history of previous clicks to correct the errors of this mask. Several examples of applying our to this use case are shown in Figure 1.

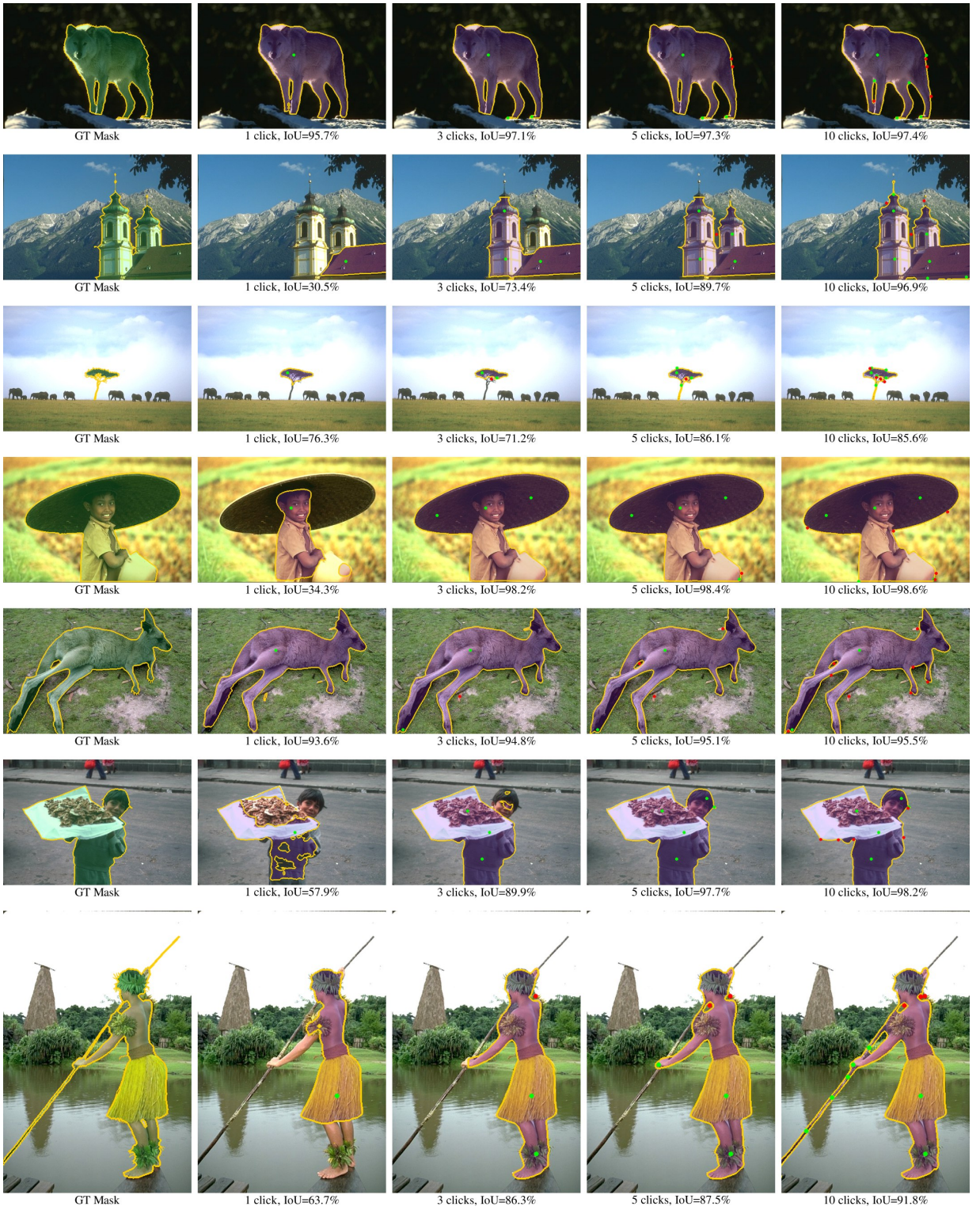

Figure 5: Visualization of interactive segmentation for the Berkeley images with a different number of clicks fed to the HRNet-18 ITER-M model and obtained by the NoC evaluation procedure [1]. Green and red dots denote positive and negative clicks, respectively. There are only 2 images from Berkeley on which our model does not converge to 90% IoU in 20 clicks. One of them is shown in the third row.

Table 7
Evaluation results on GrabCut, Berkeley, SBD, DAVIS and Pascal VOC datasets. The best and the second best results are set in bold and underlined, respectively. "H18s", "H18" and "H32" stand for "HRNet-18s", "HRNet-18" and "HRNet-32", respectively. Our models with the "IT-M" suffix are iterative models that take a mask from a previous step with $N_{iters} = 3$, otherwise they are our non-iterative baselines. The name of the training dataset is indicated below the word "Ours". "C+L" stands for COCO+LVIS.

Method		GrabCut		Berkeley	SBD		DAVIS		Pascal VOC
		NoC@85	NoC@90	NoC@90	NoC@85	NoC@90	NoC@85	NoC@90	NoC@85
GC [15]		7.98	10.00	14.22	13.60	15.96	15.13	17.41	–
GM [17]		13.32	14.57	15.96	15.36	17.60	18.59	19.50	–
RW [16]		11.36	13.77	14.02	12.22	15.04	16.71	18.31	–
ESC [17]		7.24	9.20	12.11	12.21	14.86	15.41	17.70	–
GSC [17]		7.10	9.12	12.57	12.69	15.31	15.35	17.52	–
DIOS with GC [1]		–	6.04	8.65	–	–	–	–	6.88
Latent diversity [19]		3.20	4.79	–	7.41	10.78	5.05	9.57	–
RIS-Net [20]		–	5.00	6.03	–	–	–	–	5.12
ITIS [14]		–	5.60	–	–	–	–	–	3.80
CAG [36]		–	3.58	5.60	–	–	–	–	3.62
BRS [2]		2.60	3.60	5.08	6.59	9.78	5.58	8.24	–
FCA-Net (SIS) [22]		–	2.08	3.92	–	–	–	7.57	2.69
IA+SA [3]		–	3.07	4.94	–	–	5.16	–	3.18
f-BRS-B [4]		2.50	2.98	4.34	5.06	8.08	5.39	7.81	–
Ours SBD	H18	1.96	2.41	3.95	4.12	6.66	5.08	7.17	2.94
	H18 IT-M	1.76	2.04	3.22	**3.39**	**5.43**	4.94	6.71	<u>2.51</u>
Ours C+L	H18	1.54	1.70	2.48	4.26	6.86	4.79	6.00	2.59
	H18s IT-M	1.54	1.68	2.60	4.04	6.48	4.70	5.98	2.57
	H18 IT-M	**1.42**	**1.54**	<u>2.26</u>	3.80	6.06	<u>4.36</u>	<u>5.74</u>	**2.28**
	H32 IT-M	<u>1.46</u>	<u>1.56</u>	**2.10**	<u>3.59</u>	<u>5.71</u>	**4.11**	**5.34**	2.57

5.3. Comparison with Previous Works

The quantitative results are summarized in Table 7. We notice that even the proposed baseline model with HRNet18+OCR backbone outperforms all previous methods. The proposed iterative method that uses a mask from the previous click sets a new state-of-the-art in interactive segmentation on all five benchmarks. Another notable fact is that the smallest among the proposed models with HRNet18s+OCR backbone performs on par with heavier ones, making it possible to use the proposed method on devices with low computational capability.

We provide plots of mean IoU with respect to the number of clicks for the GrabCut, Berkeley, DAVIS and SBD datasets in Figure 6. The proposed method outperforms previous state-of-the-art works and shows great improvement in accuracy and stability while avoiding accuracy drops after any click and converging to a better result.

We also visualise evaluation process of the proposed iteratively trained HRNet-18 model on some images from the Berkeley dataset in Figure 5.

6. Conclusion

Most of recent studies on interactive segmentation propose models that heavily rely on additional inference-time optimization schemes. In our work, we have demonstrated that a pure feedforward model with a modern backbone architecture can achieve or even surpass current state-of-the-art results. We have introduced a new model able to modify existing segmentation masks as well as segment new objects without any prior masks. It sets a new state-of-the-art on all common interactive segmentation benchmarks.

Our experiments have proved that a training dataset has a major impact on the model performance. We have proposed to combine the two existing instance segmentation datasets, COCO [12] and LVIS [13], and use it for interactive segmentation. The resulting large and diverse dataset with high-quality annotations has significantly improved the generalization ability of our model. Training on this dataset allows to push the state-of-the-art results achieved by our model even further.

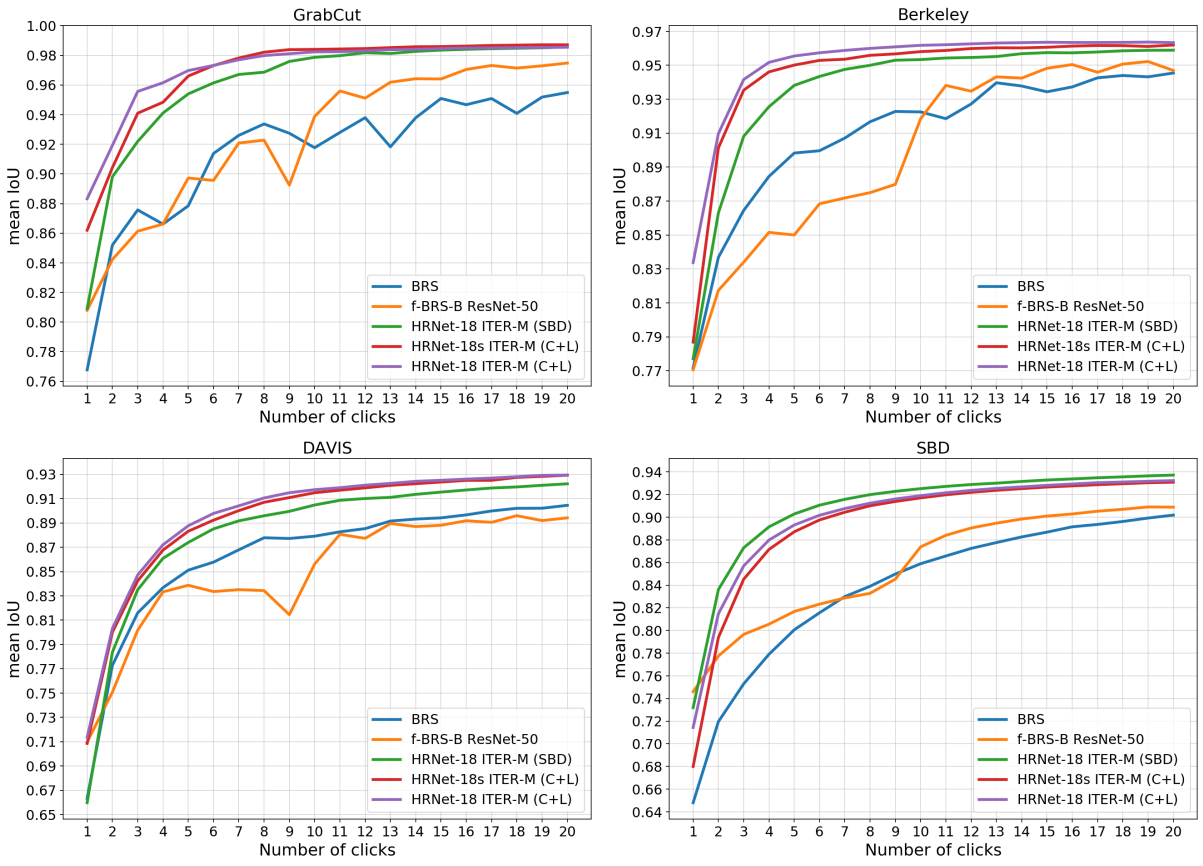

Figure 6: Mean IoU@k for varying number of clicks k on GrabCut, Berkeley, DAVIS and SBD. The iterative models (denoted as ITER-M) show stable performance without accuracy drops and converge to a better IoU. Names of the training datasets are enclosed in parentheses. "C+L" stands for COCO+LVIS.

Acknowledgment. We thank Julia Churkina for her assistance with editing and for comments that greatly improved the manuscript.

References

[1] N. Xu, B. Price, S. Cohen, J. Yang, T. Huang, Deep interactive object selection, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2016, pp. 373–381. doi:10.1109/cvpr.2016.47.

[2] W.-D. Jang, C.-S. Kim, Interactive image segmentation via backpropagating refinement scheme, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2019, pp. 5297–5306. doi:10.1109/cvpr.2019.00544.

[3] T. Kontogianni, M. Gygli, J. Uijlings, V. Ferrari, Continuous adaptation for interactive object segmentation by learning from corrections, in: Computer Vision – ECCV 2020, Springer International Publishing, 2020, pp. 579–596. doi:10.1007/978-3-030-58517-4_34.

[4] K. Sofiiuk, I. Petrov, O. Barinova, A. Konushin, F-BRS: Rethinking backpropagating refinement for interactive segmentation, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2020, pp. 8623–8632. doi:10.1109/cvpr42600.2020.00865.

[5] J. Bai, X. Wu, Error-tolerant scribbles based interactive image segmentation, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2014, pp. 392–399. doi:10.1109/cvpr.2014.57.

[6] D. Lin, J. Dai, J. Jia, K. He, J. Sun, ScribbleSup: Scribble-supervised convolutional networks for semantic segmentation, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2016, pp. 3159–3167. doi:10.1109/cvpr.2016.344.

[7] D. Freedman, T. Zhang, Interactive graph cut based segmentation with shape priors, in: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), IEEE, 2005, pp. 755–762. doi:10.1109/cvpr.2005.191.

[8] K.-K. Maninis, S. Caelles, J. Pont-Tuset, L. V. Gool, Deep extreme cut: From extreme points to object segmentation, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, 2018, pp. 616–625. doi:10.1109/cvpr.2018.00071.

[9] D. Acuna, H. Ling, A. Kar, S. Fidler, Efficient interactive annotation of segmentation datasets with polygon-RNN++, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, 2018, pp. 859–868. doi:10.1109/cvpr.2018.00096.

[10] R. Benenson, S. Popov, V. Ferrari, Large-scale interactive object segmentation with human annotators, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2019, pp. 11700–11709. doi:10.1109/cvpr.2019.01197.

[11] E. Agustsson, J. R. Uijlings, V. Ferrari, Interactive full image segmentation by considering all regions jointly, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2019, pp. 11622–11631. doi:10.1109/cvpr.2019.01189.

[12] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft COCO: Common objects in context, in: Computer Vision – ECCV 2014, Springer International Publishing, 2014, pp. 740–755. doi:10.1007/978-3-319-10602-1_48.

[13] A. Gupta, P. Dollár, R. Girshick, LVIS: A dataset for large vocabulary instance segmentation, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2019, pp. 5356–5364. doi:10.1109/cvpr.2019.00550.

[14] S. Mahadevan, P. Voigtlaender, B. Leibe, Iteratively trained interactive segmentation, in: British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018, BMVA Press, 2018, p. 212.

[15] Y. Boykov, M.-P. Jolly, Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images, in: Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001, volume 1, IEEE Comput. Soc, 2001, pp. 105–112. doi:10.1109/iccv.2001.937505.

[16] L. Grady, Random walks for image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (2006) 1768–1783.

[17] V. Gulshan, C. Rother, A. Criminisi, A. Blake, A. Zisserman, Geodesic star convexity for interactive image segmentation, in: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 3129–3136. doi:10.1109/cvpr.2010.5540073.

[18] C. Rother, V. Kolmogorov, A. Blake, "GrabCut": interactive foreground extraction using iterated graph cuts, ACM Transactions on Graphics 23 (2004) 309–314.

[19] Z. Li, Q. Chen, V. Koltun, Interactive image segmentation with latent diversity, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, 2018, pp. 577–585. doi:10.1109/cvpr.2018.00067.

[20] J. Liew, Y. Wei, W. Xiong, S.-H. Ong, J. Feng, Regional interactive image segmentation networks, in: 2017 IEEE International Conference on Computer Vision (ICCV), IEEE, 2017, pp. 2746–2754. doi:10.1109/iccv.2017.297.

[21] J. H. Liew, S. Cohen, B. Price, L. Mai, S.-H. Ong, J. Feng, MultiSeg: Semantically meaningful, scale-diverse segmentations from minimal user input, in: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), IEEE, 2019, pp. 662–670. doi:10.1109/iccv.2019.00075.

[22] Z. Lin, Z. Zhang, L.-Z. Chen, M.-M. Cheng, S.-P. Lu, Interactive image segmentation with first click attention, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2020, pp. 13339–13348. doi:10.1109/cvpr42600.2020.01335.

[23] M. M. Cheng, V. A. Prisacariu, S. Zheng, P. H. S. Torr, C. Rother, DenseCut: Densely connected CRFs for realtime GrabCut, Computer Graphics Forum 34 (2015) 193–201.

[24] J. Wu, Y. Zhao, J.-Y. Zhu, S. Luo, Z. Tu, MILCut: A sweeping line multiple instance learning paradigm for interactive image segmentation, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2014, pp. 256–263. doi:10.1109/cvpr.2014.40.

[25] S. Zhang, J. H. Liew, Y. Wei, S. Wei, Y. Zhao, Interactive object segmentation with inside-outside guidance, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2020, pp. 12234–12244. doi:10.1109/cvpr42600.2020.01225.

[26] T. H. Kim, K. M. Lee, S. U. Lee, Generative image segmentation using random walks with restart, in: Computer Vision – ECCV 2008, Springer Berlin Heidelberg, 2008, pp. 264–275. doi:10.1007/978-3-540-88690-7_20.

[27] H. Ding, S. Cohen, B. Price, X. Jiang, PhraseClick: Toward achieving flexible interactive segmentation by phrase and click, in: Computer Vision – ECCV 2020, Springer International Publishing, 2020, pp. 417–435. doi:10.1007/978-3-030-58580-8_25.

[28] P. O. Pinheiro, T.-Y. Lin, R. Collobert, P. Dollár, Learning to refine object segments, in: Computer Vision – ECCV 2016, Springer International Publishing, 2016, pp. 75–91. doi:10.1007/978-3-319-46448-0_5.

[29] Y. Yuan, J. Xie, X. Chen, J. Wang, SegFix: Model-agnostic boundary refinement for segmentation, in: Computer Vision – ECCV 2020, Springer International Publishing, 2020, pp. 489–506. doi:10.1007/978-3-030-58610-2_29.

[30] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A. L. Yuille, DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs, IEEE Transactions on Pattern Analysis and Machine Intelligence 40 (2017) 834–848.

[31] P. Zhou, B. Price, S. Cohen, G. Wilensky, L. S. Davis, Deepstrip: High-resolution boundary refinement, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2020, pp. 10558–10567. doi:10.1109/cvpr42600.2020.01057.

[32] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-decoder with atrous separable convolution for semantic image segmentation, in: Computer Vision – ECCV 2018, Springer International Publishing, 2018, pp. 833–851. doi:10.1007/978-3-030-01234-2_49.

[33] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, B. Xiao, Deep high-resolution representation learning for visual recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence (2020) 1–1.

[34] Y. Yuan, X. Chen, J. Wang, Object-contextual representations for semantic segmentation, in: Computer Vision – ECCV 2020, Springer International Publishing, 2020, pp. 173–190. doi:10.1007/978-3-030-58539-6_11.

[35] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2009, pp. 248–255. doi:10.1109/cvprw.2009.5206848.

[36] S. Majumder, A. Yao, Content-aware multi-level guidance for interactive instance segmentation, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2019, pp. 11602–11611. doi:10.1109/cvpr.2019.01187.

[37] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: 2017 IEEE International Conference on Computer Vision (ICCV), IEEE, 2017, pp. 2980–2988. doi:10.1109/iccv.2017.324.

[38] K. Sofiiuk, O. Barinova, A. Konushin, O. Barinova, AdaptIS: Adaptive instance selection network, in: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), IEEE, 2019, pp. 7355–7363. doi:10.1109/iccv.2019.00745.

[39] B. Hariharan, P. Arbelaez, L. Bourdev, S. Maji, J. Malik, Semantic contours from inverse detectors, in: 2011 International Conference on Computer Vision, IEEE, 2011, pp. 991–998. doi:10.1109/iccv.2011.6126343.

[40] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, A. Zisserman, The pascal visual object classes (VOC) challenge, International Journal of Computer Vision 88 (2009) 303–338.

[41] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, A. Kolesnikov, T. Duerig, V. Ferrari, The open images dataset v4, International Journal of Computer Vision 128 (2020) 1956–1981.

[42] X. Hu, Y. Jiang, K. Tang, J. Chen, C. Miao, H. Zhang, Learning to segment the tail, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2020, pp. 14045–14054. doi:10.1109/cvpr42600.2020.01406.

[43] R. Jiawei, C. Yu, X. Ma, H. Zhao, S. Yi, et al., Balanced meta-softmax for long-tailed visual recognition, Advances in Neural Information Processing Systems 33 (2020).

[44] J. Tan, C. Wang, B. Li, Q. Li, W. Ouyang, C. Yin, J. Yan, Equalization loss for long-tailed object recognition, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2020, pp. 11662–11671. doi:10.1109/cvpr42600.2020.01168.

[45] X. Wang, R. Zhang, T. Kong, L. Li, C. Shen, Solov2: Dynamic and fast instance segmentation, Advances in Neural Information Processing Systems 33 (2020).

[46] D. Martin, C. Fowlkes, D. Tal, J. Malik, A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, in: Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001, volume 2, IEEE Comput. Soc, 2001, pp. 416–423. doi:10.1109/iccv.2001.937655.

[47] K. McGuinness, N. E. O'Connor, A comparative evaluation of interactive segmentation algorithms, Pattern Recognition 43 (2010) 434–444.

[48] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. V. Gool, M. Gross, A. Sorkine-Hornung, A benchmark dataset and evaluation methodology for video object segmentation, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2016, pp. 724–732. doi:10.1109/cvpr.2016.85.

[49] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, Pytorch: An imperative style, high-performance deep learning library, in: H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, 2019, pp. 8024–8035.

[50] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, A. Torralba, Scene parsing through ADE20k dataset, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2017, pp. 633–641. doi:10.1109/cvpr.2017.544.

[51] M. A. Rahman, Y. Wang, Optimizing intersection-over-union in deep neural networks for image segmentation, in: Advances in Visual Computing, Springer International Publishing, 2016, pp. 234–244. doi:10.1007/978-3-319-50835-1_22.