Interactive Image Segmentation via Backpropagating Refinement Scheme

Won-Dong Jang
Harvard University
Cambridge, MA
wdjang@g.harvard.edu

Chang-Su Kim
Korea University
Republic of Korea
changsukim@korea.ac.kr

Abstract

An interactive image segmentation algorithm, which accepts user-annotations about a target object and the background, is proposed in this work. We convert user-annotations into interaction maps by measuring distances of each pixel to the annotated locations. Then, we perform the forward pass in a convolutional neural network, which outputs an initial segmentation map. However, the user-annotated locations can be mislabeled in the initial result. Therefore, we develop the backpropagating refinement scheme (BRS), which corrects the mislabeled pixels. Experimental results demonstrate that the proposed algorithm outperforms the conventional algorithms on four challenging datasets. Furthermore, we demonstrate the generality and applicability of BRS in other computer vision tasks, by transforming existing convolutional neural networks into user-interactive ones.

1. Introduction

Interactive image segmentation is a task to separate a target object (or foreground) from the background. A target object is annotated by a user in the type of bounding box [51, 24, 42] or scribble [52, 11, 10, 25]. For the bounding box annotation, a box is supposed to surround a target. On the contrary, in the scribble-based interface, foreground and background scribbles are drawn on foreground and background regions, respectively. In general, scribble-based algorithms yield more detailed object masks than box-based ones do. In scribble-based algorithms, it is important to extract an accurate mask of a target using fewer scribbles.

Thanks to the release of large image datasets [23] and the use of convolution layers, deep-learning-based algorithms have been showing remarkable performances in segmentation problems: semantic segmentation [13, 30, 35, 6], saliency detection [29, 36], and object proposal [39, 38]. Most deep-learning-based segmentation algorithms exploit convolutional neural networks (CNNs). In [35, 30, 29], the encoder-decoder architecture [40] is used: deep features are extracted from the encoders, and they are used to predict pixel-level segmentation or saliency labels in the decoders. The encoder-decoder architecture can provide reliable performances, since it can adopt well-trained encoders, including AlexNet [23], VGGNet [44], GoogLeNet [48], ResNet [15], and DenseNet [17]. In segmentation tasks, it is important to achieve segments with accurate and detailed boundaries. However, deep features from an encoder lose most low-level details and have high-level (or semantic) information only [56]. To address this problem, [29, 39] adopt skip connections that exploit intermediate output responses of the encoders for improving segmentation qualities.

Backpropagation for activations[1] is a process that conveys data through network layers backwardly. In [43, 46, 56, 58], backpropagation schemes have been developed to visualize characteristics of neural networks. Also, texture synthesis [8] and image style transfer [9] are performed via backpropagation. They update activation responses backwardly, while freezing parameters, in the networks.

In this work, based on a backpropagation scheme, we propose a novel interactive image segmentation algorithm, which accepts user scribbles. To segment a target object, we train a fully convolutional neural network. In the test phase, we perform the forward pass in the proposed network using an input image and user-annotations. We also develop the backpropagating refinement scheme (BRS), which constrains user-specified locations to have correct labels and refines the segmentation result of the forward pass. To this end, we define two energy functions: corrective energy and inertial energy. We minimize a weighted sum of the two energies via backpropagation. Experimental results show that the proposed BRS algorithm outperforms the conventional algorithms [11, 10, 3, 52, 50, 2, 27, 26] on the GrabCut [42], Berkeley [34], DAVIS [37], and SBD [12] datasets. Also, we generalize BRS for various CNN-based vision techniques to make them interactive with user-annotations. To summarize, this work has three main contributions.

[1]This is different from the typical backpropagation for parameters, which is used for training neural networks.

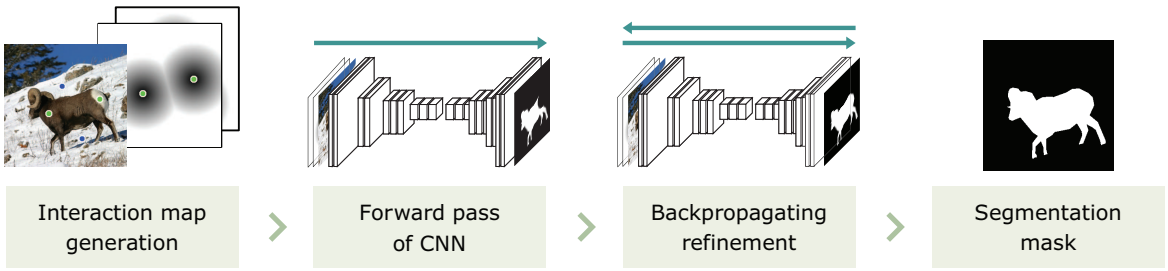


Figure 1. Overview of the proposed algorithm: we perform this segmentation process again when a user provides a new annotation.

- ▷ Development of a CNN for interactive image segmentation, which is fully convolutional.
- ▷ Introduction of the backpropagating refinement strategy, which corrects mislabeled locations.
- ▷ Generalization of BRS, which can make existing CNNs user-interactive without extra training.

2. Related Work

2.1. Interactive Image Segmentation

In interactive image segmentation, a target object is annotated roughly by a user and then is extracted as a binary mask. Interactive segmentation algorithms can be categorized into box-interfaced or scribble-interfaced ones. A box-interfaced one obtains the mask of a target object within a given bounding box. On the other hand, a scribble-interfaced one accepts foreground and background annotations from a user. While a box-interfaced algorithm attempts to obtain a one-shot segmentation result in general, a scribble-interfaced algorithm allows a user to provide scribbles several times until a satisfactory result is obtained.

Box-interfaced algorithms: Rother *et al*. [42] construct Gaussian mixture models for foreground and background, respectively, and then use the models in graph-cut optimization to obtain a foreground mask. These processes are performed iteratively until the convergence. To avoid these iterations, Tang *et al*. [49] define a cost function that can be minimized in a single pass of graph-cut optimization. Assuming that user-provided bounding boxes are not too loose, Lempitsky *et al*. [24] use the notion of box tightness to prevent excessive shrinking of a target segment. Wu *et al*. [51] over-segment an image into superpixels and generate the foreground and background bags for multiple instance learning. The foreground bag consists of the superpixels inside a bounding box, and the background bag contains the other superpixels.

Scribble-interfaced algorithms: Li *et al*. [25] compute the distances from each pixel to foreground and background seeds in terms of RGB colors and employ a graph-cut algorithm to separate a target object from the background. Grady [10] lets a random walker start at each pixel and finds the first foreground or background seeds that the walker reaches. Kim *et al*. [21] perform the random walk with restart simulation to compute affinities between pixels. Gulshan *et al*. [11] propose a shape constraint for interactive image segmentation and use geodesic distances from user scribbles to pixels for energy minimization. Kim *et al*. [22] generate various segmentation maps for an image, by employing different parameters, and then encourage pixels within a segment to have the same label in the final result. To alleviate user efforts, [47, 1] develop error-tolerant interactive image segmentation algorithms. Recently, Xu *et al*. [52] propose a deep-learning-based interactive segmentation algorithm. They generate foreground and background maps from user-annotations and concatenate them with an input image to feed it into a CNN. The probability that each pixel belongs to foreground is predicted by the network. Liew *et al*. [27] refine a global prediction by combining local predictions on patches that include pairs of foreground and background clicks. Li *et al*. [26] produce multiple hypothesis segmentations and select one using the selection network. Maninis *et al*. [31] introduce an interactive segmentation algorithm that requires human annotations on tight object boundaries. Song *et al*. [45] locates foreground and background seeds to multiply annotations automatically.

2.2. Backpropagation for Activations

In this section, we discuss backpropagation schemes that update activation responses only while fixing parameters in neural networks. Zeiler and Fergus [56] visualize characteristics of each convolutional filter using DeconvNet [57], which performs inverse processes of convolution, rectified linear function, and max pooling. They discovered that, while low-level features are extracted in shallow layers, high-level ones are produced in deep layers. Springenberg *et al*. [46] propose the guided backpropagation strategy, which produces sharper reconstructed images than [56] does. Simonyan *et al*. [43] generate the appearance model of each object class in an image classification task. They find a regularized image to maximize a classification score, by updating activation responses in the image classification

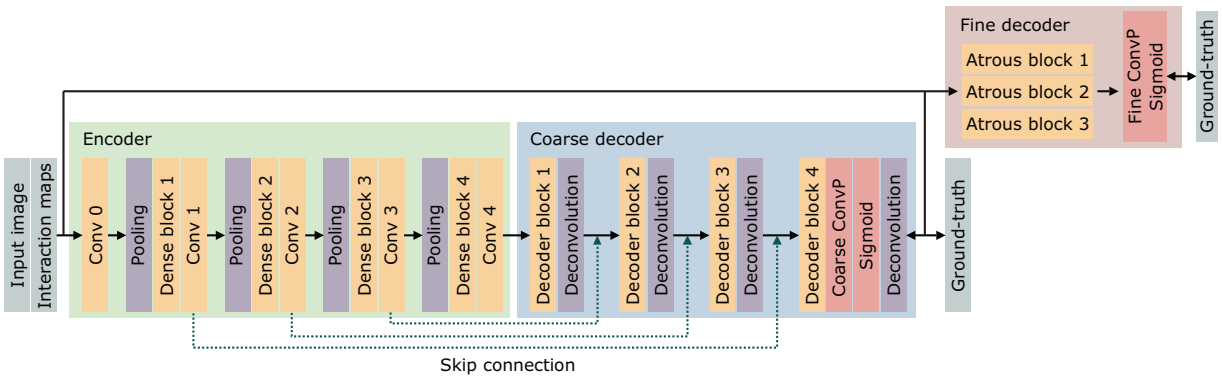


Figure 2. Architecture of the proposed network for interactive image segmentation.

network. Yosinski *et al*. [55] develop visualization tools for both convolutional filter reconstruction and class appearance model generation. Also, Zhang *et al*. [58] estimate an attention map by performing the probabilistic winner-take-all backpropagation strategy in CNNs for image classification. Given a class, they discover rough locations and shapes of corresponding objects in an image. Gatys *et al*. [8] synthesize textures via backpropagation, by encouraging a newly synthesized texture to have the same Gram matrix as an original texture. In [9], they also use backpropagation for image style transfer.

3. Proposed Algorithm

The proposed interactive image segmentation algorithm outputs a binary mask of a user-annotated object. It is a scribble-interfaced method, requiring foreground and background clicks as annotations, which indicate expected labels at the corresponding pixels.

Figure 1 is an overview of the proposed algorithm. Given user-annotations, we first generate foreground and background interaction maps. Then, we feed the input image and the interaction maps into a CNN, which yields a probability map of a user-specified object. Even though the interaction maps clearly represent the annotated labels in the clicked locations, the probability map may convey wrong information at those clicked locations. Therefore, we force the clicked locations to have the user-specified labels by employing the proposed BRS. Finally, we obtain the segmentation mask of the target object by performing the forward pass again.

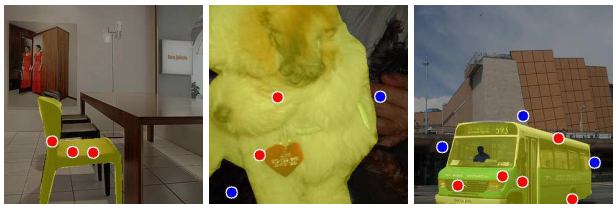
We initiate this process when a user provides the first click on a target object. Then, by taking into account the segmentation result, the user may click a new location either on the object or the background. Then, the proposed algorithm is executed again to achieve more accurate segmentation. Note that these two steps are conducted recursively until the user stops clicking.

3.1. CNN for Interactive Image Segmentation

We perform interactive image segmentation using a CNN, which accepts user-annotations. The user-annotations are converted into interaction maps, as done in [52]. Specifically, the foreground and background interaction maps are obtained, respectively, by computing the distance of each pixel to the closest user-annotated foreground and background pixels. We limit the maximum distances to 255. Figure 1 includes examples of interaction maps.

Network architecture: The proposed CNN has the encoder-decoder architecture [40] in Figure 2. As input, the proposed network takes an image and two interaction maps for foreground and background. We adopt DenseNet [17] as the encoder to extract high-level features, as well as low-level features. We use the extracted features by employing the skip connections, which have been used in many image-to-image transition tasks [39, 41, 19]. Also, we add a squeeze and excitation module [16] at the end of each dense block.

We have a coarse decoder and a fine decoder. The two decoders produce probability maps, whose elements have high probabilities on target object regions. While we predict a rough segment of a target object in the coarse decoder, the fine decoder improves its detail using low-level features. The coarse decoder consists of four decoding blocks. Each decoding block includes three convolution layers. After obtaining a coarse segment, we concatenate it with the input of the network, and feed them into the fine decoder. In the fine decoder, we use atrous convolutions [4] to expand receptive fields at high resolution tensors. Each convolution layer is followed by a parametric rectified linear unit [14] and batch normalization [18], except for the prediction layers 'Coarse ConvP' and 'Fine ConvP.' We employ the deconvolution layers to restore the spatial resolutions of down-sampled features to the original input image size. The output of the proposed network is normalized to $[0, 1]$ using the sigmoid layer. We use 3×3 and 1×1 kernels in convolution layers. Since the proposed network is fully convolutional, it does



(a) 3 FG / 0 BG (b) 2 FG / 2 BG (c) 5 FG / 3 BG

Figure 3. Examples of generated user-annotations for training. The foreground and background annotations are depicted in red and blue circles, respectively. Also, the ground-truth object masks are highlighted in yellow.

not need to modify the spatial resolution or aspect ratio of an input image for its segmentation.

Training phase: We use the SBD dataset [12] to train the proposed CNN. It includes 8,498 training images. Around each object instance, we randomly crop a 360×360 patch to yield pairs of an image patch and its object mask. We declare that the center pixel of a cropped patch belongs to foreground in the object mask. We further augment the data with horizontal flips.

Since user-annotations are not available in the SBD dataset, we imitate them through a simple clustering strategy. First, the numbers of foreground and background clicks are determined randomly within [1, 10] and [0, 10], respectively. Then, we set pixels in a ground-truth object mask as foreground candidates. On the other hand, we set background candidates to be at least 5 pixels and at most 40 pixels away from the boundaries of the ground-truth object. By applying the k-medoids algorithm [20] on each set of candidates, we find foreground and background medoids and use them as foreground and background annotations, respectively. Figure 3 exemplifies generated user-annotations.

We employ the cross-entropy losses between ground-truth masks and inferred probability maps. Whereas the initial parameters of the encoder are from [17], we initialize parameters in the decoders with random values. We train the network via the stochastic gradient descent. While we set the learning rate to 10^{-9} in the encoder, we set it to 10^{-7} for the decoders. A minibatch is composed of four training data. We first train the proposed network for 20 epochs without the fine decoder. Then, we perform learning for another 15 epochs with the fine decoder.

Inference phase: The proposed network accepts an image and foreground and background interaction maps as the input. Given user clicks, we first update the foreground and background interaction maps by computing the distance of each pixel to the nearest clicks. Then, we feed them into the proposed network to yield a probability map of the target object. We determine the locations, whose probabilities are higher than 0.5, as the foreground.

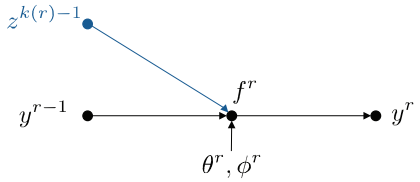


Figure 4. Notations for the proposed network. The concatenated $z^{k(r)-1}$ and y^{r-1} are fed into a convolution layer f^r.

3.2. Backpropagating Refinement Scheme

The forward pass of the proposed algorithm yields a decent segmentation quality. However, it has a shortcoming of being incapable of guaranteeing that clicked pixels have user-annotated labels. In other words, even clicked pixels may have incorrect labels in the segmentation result. Therefore, we enforce them to be labeled correctly to achieve more accurate segmentation. The proposed BRS performs backpropagation iteratively until all clicked pixels have correct labels.

Let us first define notations for the proposed network. In Figure 4, tensors y^{r-1} and z^{r-1} are concatenated, and parameters θ^r and ϕ^r are used to obtain y^r, which denotes the responses of the rth layer in the network. Hence, y^0, y^R, and z^0 become an input image, the output of the network, interaction maps, respectively, where R is the index of the last layer in the fine decoder. Thus, y^r can be formulated as

$$y^r = f^r(y^{r-1}, z^{r-1}, \theta^r, \phi^r). \tag{1}$$

Note that this formulation can represent all convolution layers in the proposed network including the first layer and the layers with skip connections.

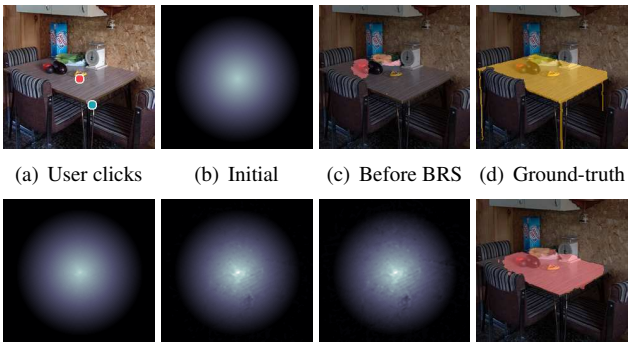
Initial interaction maps, which are converted from the user-annotations, may be imperfect for making the network yield correct labels in user-annotated locations. The correction can be done by modifying initial interaction maps or fine-tuning the network. However, the re-trained network may lose the knowledge learned in the training phase. Therefore, we choose to modify interaction maps, instead of fine-tuning network. The goal of BRS is to assign correct labels to user-annotated locations by optimizing interaction maps z^0. By combining a corrective energy $\mathcal{E}_\text{C}$ and an inertial energy $\mathcal{E}_\text{I}$, the energy function $\mathcal{E}(z^0)$ of the interaction maps z^0 is defined as

$$\mathcal{E}(z^0) = \mathcal{E}_\text{C}(z^0) + \lambda \mathcal{E}_\text{I}(z^0) \tag{2}$$

where λ matches scale differences between the two energies, which is fixed to 10^{-3}. Then, we find an optimal z^0 by minimizing $\mathcal{E}(z^0)$,

$$\hat{z}^0 = \arg\min_{z^0} \mathcal{E}(z^0). \tag{3}$$

The minimization of the corrective energy compels the proposed network to yield correct labels in user-annotated



(a) User clicks (b) Initial (c) Before BRS (d) Ground-truth

(e) 5 iterations (f) 10 iterations (g) Convergence (h) After BRS

Figure 5. Foreground and background user-annotations are presented in red and blue dots in (a), respectively. An initial FG interaction map in (b) is updated in (e), (f), and (g). Segmentation results before and after BRS are in (c) and (h). The BG interaction map is not shown due to limited space.

locations. We define the corrective energy as

$$\mathcal{E}_{\mathrm{C}}(z^0) = \sum_{\mathbf{u}\in\mathcal{U}} \big(l(\mathbf{u}) - y^R(\mathbf{u})\big)^2 \tag{4}$$

where $\mathcal{U}$ is the set of annotated pixels. Also, $l(\mathbf{u})$ denotes a user-annotated label, which is 1 for foreground and 0 for background, and $y^R(\mathbf{u})$ is the output of the proposed network. The derivative of the corrective energy can be computed through a backpropagation technique. By employing these backward recursive equations, we obtain the partial derivative, $\frac{\partial \mathcal{E}_{\mathrm{C}}}{\partial z^0}$, of the corrective energy with respect to the interaction maps.

The inertial energy prevents excessive perturbations of the interaction maps, which is defined as

$$\mathcal{E}_{\mathrm{I}}(z^0) = \sum_{\mathbf{x}\in\mathcal{N}} \big(z^0(\mathbf{x}) - z_{\mathrm{i}}^0(\mathbf{x})\big)^2 \tag{5}$$

where $\mathcal{N}$ is the set of coordinates in the interaction maps, z_{i}^0 denotes the initial interaction maps used in the forward pass. The inertial energy yields a high cost when the interaction maps are different from their initial values. We compute the partial derivative of the inertial energy with respect to the interaction maps by

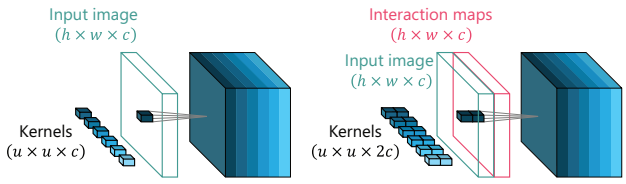
$$\frac{\partial \mathcal{E}_{\mathrm{I}}}{\partial z^0} = 2 \times \sum_{\mathbf{x}\in\mathcal{N}} \big(z^0(\mathbf{x}) - z_{\mathrm{i}}^0(\mathbf{x})\big), \tag{6}$$

which is easily obtainable at the input layer of the network.

We blend the derivatives of the corrective energy and the inertial energy using the parameter λ in (2) as

$$\frac{\partial \mathcal{E}}{\partial z^0} = \frac{\partial \mathcal{E}_{\mathrm{C}}}{\partial z^0} + \lambda \frac{\partial \mathcal{E}_{\mathrm{I}}}{\partial z^0}. \tag{7}$$

Finally, we minimize the energy function, by employing L-BFGS algorithm [28], and obtain the optimal interaction maps. Note that the forward pass and the backpropagation are performed alternately. Figure 5 shows how BRS updates a foreground interaction map to correct mislabeled pixels. Note that BRS considers the background user click when modifying the foreground interaction map.



(a) Baseline architecture (b) Interactive architecture

Figure 6. Reconfiguration of a network architecture in the first convolution layer. The baseline architecture in (a) is transformed to the interactive one in (b) by the training-free conversion scheme.

3.3. Generalization

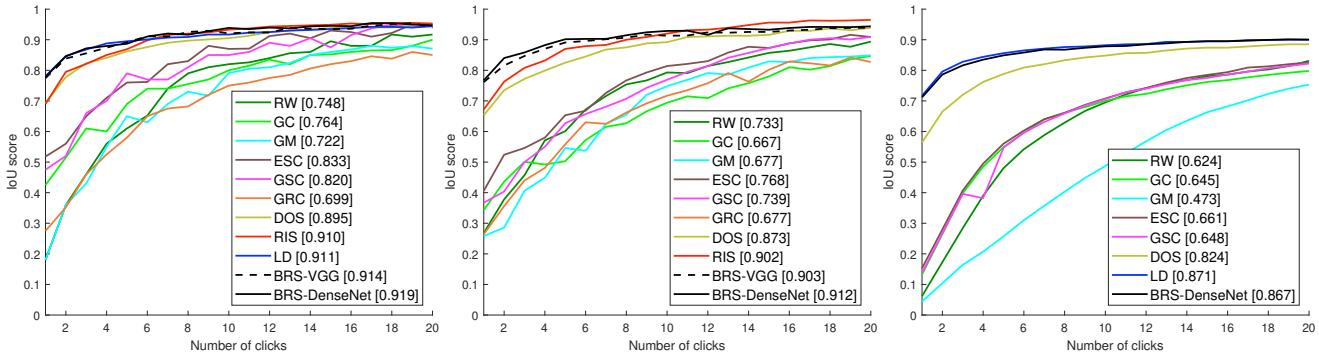
We apply the proposed BRS to the well trained network with the interaction maps. However, we can employ BRS for general networks that are not trained with interaction maps. Note that the recursive backpropagation computations in (4) are still applicable, even when the architecture of a network (*e.g.* the number of convolution layers and skip connections between the encoder and the decoder) is different from that of the proposed network. Based on this generality, we show that BRS can transform existing CNNs into user-interactive ones without extra training.

The development of interactive algorithms requires time and expertise for training, in terms of composition of training data, network architectures, and hyperparameters. Also, even though interactive algorithms are trained successfully, they often yield inferior results compared to non-interactive algorithms when user interactions are not given.

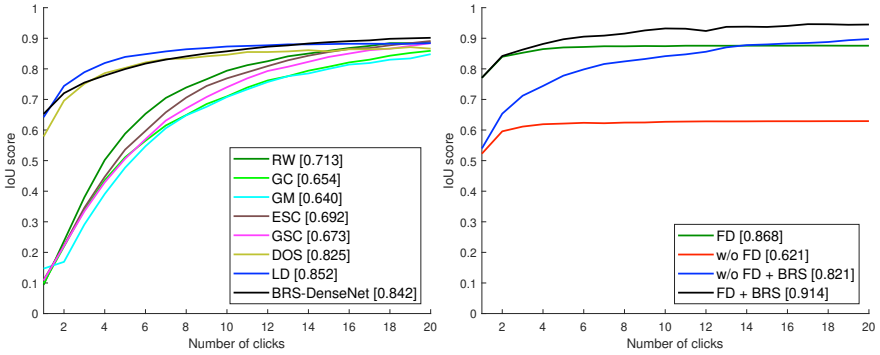
We develop a training-free conversion scheme to overcome these issues. Given a baseline network, we reconfigure its architecture at the first convolution layer, as shown in Figure 6. In addition to an input image, we also use interaction maps. As input, we concatenate the image and the maps, which share the same weight parameters in the first convolution layer. Then, we can perform BRS in the reconfigured network to achieve interaction. Notice that the network needs no additional training. Moreover, it yields the same output as the original algorithm, when the interaction maps are filled with zeros. Applications of the training-free conversion will be shown in Section 4.

4. Experimental Results

We evaluate the performance of the proposed interactive image segmentation algorithm on four datasets: GrabCut [42], Berkeley [34], DAVIS [37], and SBD [12]. The GrabCut dataset [42] has 50 images for assessing interactive



(a) GrabCut (b) Berkeley (c) DAVIS



(d) SBD (e) Ablation study

Figure 7. Comparison of the average IoU scores according to the number of clicks on the GrabCut [42], Berkeley [34], DAVIS [37], and SBD [12] datasets. The legend contains the AuC score for each algorithm. An ablation study of the proposed algorithm is also in (e).

image segmentation algorithms. It provides a single object mask for each image. The Berkeley dataset [32] consists of 200 training images and 100 test images. We use 100 object masks on 96 test images, provided by [34]. Thus, some images have more than one object masks. The DAVIS dataset [37] is for benchmarking video object segmentation algorithms. Even though they are composed with video sequences, we can use their individual frames to evaluate interactive image segmentation methods. The dataset have 50 videos with high quality segmentation masks. We randomly sample 10% of the annotated frames as done in [26]. In total, 345 images are used in the evaluation. The SBD dataset [7], for evaluating object segmentation techniques, is divided into a training set of 8,498 images and a validation set of 2,820 images. Note that we use the training set to train the network in Section 3.1. Therefore, we use the validation set, which includes 6,671 instance-level objet masks, for the performance evaluation.

We use two performance measures, as in [29]. First, we compute the mean intersection over union (IoU) score according to the number of clicks and its area under curve (AuC). When computing AuC, we normalize the area to be within $[0, 1]$. Second, we adopt the NoC metric, which is the mean number of clicks required to achieve a certain IoU. We set the target IoU score as 90%.

To compare interactive segmentation algorithms fairly, we use the same clicking strategy as done in [26, 52]. In general, a user first decides the type of an annotation (*i.e.* foreground or background) by finding the dominant type of prediction errors. Thus, the clicking strategy counts the numbers of false foregrounds and false backgrounds, respectively. It chooses a background annotation if there are more false foregrounds, and a foreground annotation otherwise. Also, a user tends to click a location around the center of false predictions. Hence, the clicking strategy determines a pixel to click, which is far from the boundaries of false predictions. The maximum number of clicks is limited to 20 in all experiments.

Figure 7(a)∼(d) compares the proposed algorithm with eight conventional algorithms: graph-cut (GC) [3], geodesic matting (GM) [2], random walk (RW) [10], Euclidean star convexity (ESC) [11], geodesic star convexity (GSC) [11], Growcut (GRC) [50], deep object selection (DOS) [52], regional image segmentation (RIS) [27], and segmentation with latent diversity (LD) [26]. Note that the scores are from [26, 27]. We report two versions of the proposed algorithm using different backbone networks: BRS-VGG and BRS-DenseNet. The proposed BRS outperforms all conventional algorithms on all four datasets, with a single exception of LD [26] on the GrabCut dataset.

Table 1. Comparison of NoC 85% and 90% indices on the GrabCut [42], Berkeley [34], DAVIS [37], and SBD [12] datasets. The best and the second best results are boldfaced and underlined, respectively.

	GrabCut		Berkeley	DAVIS		SBD	
Algorithm	85%	90%	90%	85%	90%	85%	90%
GC [3]	7.98	10.00	14.33	15.13	17.41	13.60	15.96
GM [2]	13.32	14.57	15.96	18.59	19.50	15.36	17.60
RW [10]	11.36	13.77	14.02	16.71	18.31	12.22	15.04
ESC [11]	7.24	9.20	12.11	15.41	17.70	12.21	14.86
GSC [11]	7.10	9.12	12.57	15.35	17.52	12.69	15.31
GRC [50]	-	16.74	18.25	-	-	-	-
DOS [52]	5.08	6.08	8.65	9.03	12.58	9.22	12.80
RIS [27]	-	5.00	6.03	-	-	-	-
LD [26]	3.20	4.79	-	5.95	9.57	7.41	10.78
BRS-VGG	2.90	3.84	5.74	-	-	-	-
BRS-DenseNet	**2.60**	**3.60**	**5.08**	**5.58**	**8.24**	**6.59**	**9.78**

Figure 8. Segmentation results of the proposed algorithm. The segmented object masks are highlighted in yellow masks. Foreground and background user-annotations are depicted in red and blue dots, respectively.

Table 1 reports the NoC 85% and 90% indices, the mean numbers of clicks required to achieve the 85% and 90% IoU scores, respectively. The proposed algorithm requires much fewer clicks than the conventional algorithms, which indicates that the proposed algorithm yields accurate object masks with less user efforts. While the proposed algorithm is comparable to LD [26] in terms of AuC, BRS outperforms LD in both NoC 85% and NoC 90% measures significantly. This means that even though LD outputs precise segmentations, it has more failure cases than BRS does.

Figure 8 shows segmentation results of the proposed algorithm. It is observable that the proposed algorithm delineates target objects precisely and robustly. It segments out even small objects well. Also, it yields object masks with accurate boundaries, even when the colors of a target object and its background are similar. We provide more segmentation results in the supplementary materials.

Ablation study: We analyze the efficacy of each component in the proposed algorithm, by performing three ablation studies on the GrabCut and Berkeley datasets. First, we measure the performance of the proposed algorithm when only the forward pass is executed. Second, we do not employ the fine decoder. Third, we apply BRS without the fine decoder. Let us refer to the first, second, and third settings as 'FD,' 'w/o FD,' and 'w/o FD + BRS.' Table 2 lists the NoC 85% and 90% indices. In all results, the performances are degraded severely, which indicate that the proposed BRS and the fine decoder are essential for accurate interactive image segmentation. Figure 7(e) also shows that the performance of the proposed BRS is much better than the other ablated settings.

Table 2. NoC 85% and 90% indices of the proposed algorithm in various settings.

	GrabCut		Berkeley	
Setting	NoC 85%	NoC 90%	NoC 85%	NoC 90%
FD	4.12	6.12	5.33	7.65
w/o FD	14.34	17.4	17.80	19.63
w/o FD + BRS	6.60	10.28	10.09	15.30
FD+BRS	2.60	3.60	3.16	5.08

Moreover, we report the accuracy for each ablation setting by calculating the average ratio of correctly labeled pixels over user-annotated locations on the images in the GrabCut and Berkeley datasets. Figure 9 plots the accuracy in terms of the number of clicks. It is observable that BRS makes the network yield correct labels at user-annotated locations. Moreover, there is a significant improvement in the 'w/o FD + BRS' setting compared to the accuracy of 'w/o FD.' It means that the proposed BRS can correct labels at user-annotated locations regardless of the performance of networks.

Running time analysis: We measure the average computational time of the proposed algorithm in seconds per click (SPC). We test it on the DAVIS dataset [37] using a PC with an Intel i7-5820K 3.30 GHz CPU and a Titan X GPU. The proposed algorithm runs in 0.81 SPC, which is fast enough for practical usage. A realtime demo of the proposed algorithm is available in the supplementary video. Figure 10 plots how the computation time varies as the number of clicks increases. We see that the complexity is acceptable

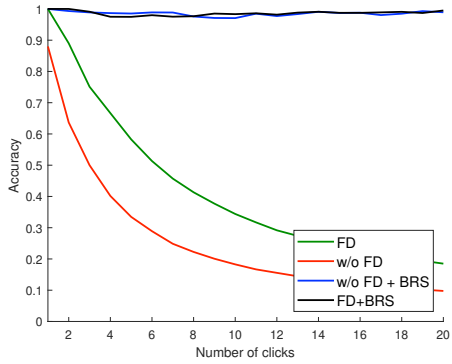


Figure 9. Comparison of accuracy curves. An accuracy is defined as the average ratio of correctly labeled pixels over user-annotated locations.

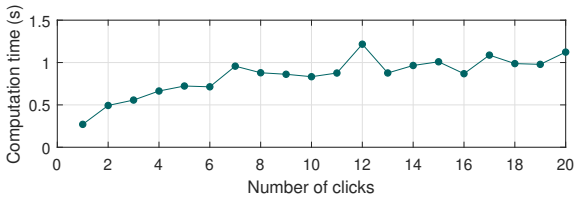


Figure 10. Computation time according to the click number.

Table 3. The average accuracy of the interactive FCN according to the number of clicks.

# of clicks	Baseline	1	2	3	4	5
Avg. acc. (%)	65.4	70.9	72.5	73.5	74.0	74.4

even when a large number of clicks are given.

Applications of the training-free conversion: To demonstrate the generality and the versatile applicability of BRS, we apply the training-free conversion scheme in Section 3.3 to three vision tasks: semantic segmentation, saliency detection, and medical image segmentation.

First, we use FCN [30] as a baseline semantic segmentation algorithm. A user annotates a label on a single pixel, which indicates its class, such as aeroplane, bicycle, and bird. We evaluate this interactive FCN on the validation set in the PASCAL VOC 2012 dataset [7]. Table 3 lists average accuracies according to the number of clicks. The performance is significantly improved even with a small number of user-annotations.

Second, for saliency detection, DHSNet [29] is used as a baseline network. As an annotation, a binary label of being salient or non-salient is used to correct a mislabeled location. We use three datasets: ECSSD [53], DUT-OMRON [54], and MSRA10K [5]. Figure 11 shows the precision-recall curves of the interactive DHSNet in terms of the number of clicked locations on the ECSSD dataset. It is observable that, with BRS, DHSNet provides better saliency detection performance by accepting user-annotations. Due to the page limitation, we report the performance of the interactive DHSNet on the other two

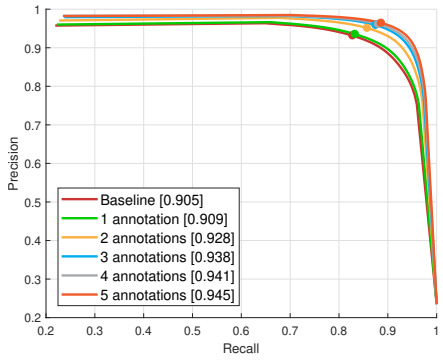


Figure 11. Comparison of the precision-recall curves of the interactive DHSNet, according to the numbers of annotations, on the ECSSD [53] dataset. A legend includes the maximum F-score for each algorithm.

Table 4. Average IoU scores and gains according to the number of clicks. An IoU gain is measured on annotated cells only.

# of clicks	Baseline	1	2	3	4	5
Avg. IoU (%)	88.2	88.9	89.1	89.4	89.5	89.6
Avg. gain (%)	-	3.6	1.8	0.8	1.3	0.3

datasets in the supplementary document.

Third, U-Net [41] is one of the most well-known medical image segmentation algorithms. It segments out cells from the background. We assess the performance of the interactive U-Net on the two test sequences in the PhC-U373 dataset [33]. Since ground-truth segmentation maps are not available, we extract them manually. Table 4 reports the average IoU scores according to the numbers of annotations. For a focused analysis, we also measure the average IoU gains on only the cells that include annotated locations. The interactive U-Net yields better segmentation qualities when more clicks are given. To summarize, the training-free conversion, based on the proposed BRS, can convert various CNN-based vision algorithms into interactive ones effectively and easily.

5. Conclusions

In this work, we proposed a novel interactive image segmentation algorithm. First, a user-annotation is transformed into the interaction maps. Then, the proposed network yields a probability map, which is an initial segmentation result. We perform BRS to enforce user-specified locations to have correct labels. Experimental results demonstrated that the proposed algorithm outperforms the conventional algorithms [11, 10, 3, 52, 50, 2, 27, 26] on the GrabCut [42], Berkeley [34], DAVIS [37], and SBD [12] datasets. Moreover, we generalized BRS to make CNN-based techniques interactive with user-annotations. Specifically, we showed that the training-free conversion scheme can be successfully applied to semantic segmentation, saliency detection, and medical image segmentation.

References

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

[2] Xue Bai and Guillermo Sapiro. Geodesic matting: A framework for fast interactive image and video segmentation and matting. *Int. J. Comput. Vis.*, 82(2):113–132, 2009.

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

[4] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(4):834–848, 2018.

[5] Ming-Ming Cheng, Niloy J Mitra, Xiaolei Huang, Philip HS Torr, and Shi-Min Hu. Global contrast based salient region detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(3):569–582, 2015.

[6] Jifeng Dai, Kaiming He, and Jian Sun. Instance-aware semantic segmentation via multi-task network cascades. In *CVPR*, pages 3150–3158, 2016.

[7] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The PASCAL visual object classes (VOC) challenge. *Int. J. Comput. Vis.*, 88(2):303–338, 2010.

[8] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. In *NIPS*, pages 262–270, 2015.

[9] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *CVPR*, pages 2414–2423, 2016.

[10] L. Grady. Random walks for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(11):1768–1783, 2006.

[11] V. Gulshan, C. Rother, A. Criminisi, A. Blake, and A. Zisserman. Geodesic star convexity for interactive image segmentation. In *CVPR*, pages 3129–3136, 2010.

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

[13] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Simultaneous detection and segmentation. In *ECCV*, pages 297–312, 2014.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, pages 1026–1034, 2015.

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

[16] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2018.

[17] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, 2017.

[18] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456, 2015.

[19] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, pages 1125–1134, 2017.

[20] L Kaufman and PJ Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley-Interscience, 2005.

[21] Tae Hoon Kim, Kyoung Mu Lee, and Sang Uk Lee. Generative image segmentation using random walks with restart. In *ECCV*, pages 264–275, 2008.

[22] Tae Hoon Kim, Kyoung Mu Lee, and Sang Uk Lee. Nonparametric higher-order learning for interactive segmentation. In *CVPR*, pages 3201–3208, 2010.

[23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

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

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

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

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

[28] Dong C Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1):503–528, 1989.

[29] Nian Liu and Junwei Han. DHSNet: Deep hierarchical saliency network for salient object detection. In *CVPR*, pages 678–686, 2016.

[30] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015.

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

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

[33] Martin Maška, Vladimír Ulman, David Svoboda, Pavel Matula, Petr Matula, Cristina Ederra, Ainhoa Urbiola, Tomás España, Subramanian Venkatesan, Deepak MW Balak, et al. A benchmark for comparison of cell tracking algorithms. *Bioinformatics*, 30(11):1609–1617, 2014.

[34] Kevin McGuinness and Noel E O'connor. A comparative evaluation of interactive segmentation algorithms. *Pattern Recog.*, 43(2):434–444, 2010.

[35] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *ICCV*, pages 1520–1528, 2015.

[36] Junting Pan, Elisa Sayrol, Xavier Giro-i Nieto, Kevin McGuinness, and Noel E O'Connor. Shallow and deep convolutional networks for saliency prediction. In *CVPR*, pages 598–606, 2016.

[37] F Perazzi, J Pont-Tuset1 B McWilliams, L Van Gool, M Gross, and A Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, pages 724–732, 2016.

[38] Pedro O Pinheiro, Ronan Collobert, and Piotr Dollár. Learning to segment object candidates. In *NIPS*, pages 1990–1998, 2015.

[39] Pedro O Pinheiro, Tsung-Yi Lin, Ronan Collobert, and Piotr Dollár. Learning to refine object segments. In *ECCV*, pages 75–91, 2016.

[40] Marc'Aurelio Ranzato, Fu Jie Huang, Y-Lan Boureau, and Yann LeCun. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *CVPR*, 2007.

[41] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241, 2015.

[42] C. Rother, V. Kolmogorov, and A. Blake. GrabCut: Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graphics*, 23(3):309–314, 2004.

[43] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *ICLRW*, 2014.

[44] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[45] Gwangmo Song, Heesoo Myeong, and Kyoung Mu Lee. SeedNet: Automatic seed generation with deep reinforcement learning for robust interactive segmentation. In *CVPR*, pages 1760–1768, 2018.

[46] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. In *ICLRW*, 2015.

[47] Kartic Subr, Sylvain Paris, Cyril Soler, and Jan Kautz. Accurate binary image selection from inaccurate user input. In *Computer Graphics Forum*, pages 41–50, 2013.

[48] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

[49] Meng Tang, Lena Gorelick, Olga Veksler, and Yuri Boykov. GrabCut in one cut. In *ICCV*, pages 1769–1776, 2013.

[50] Vladimir Vezhnevets and Vadim Konouchine. Growcut: Interactive multi-label nd image segmentation by cellular automata. In *Proc. of GraphiCon*, volume 1, pages 150–156, 2005.

[51] Jiajun Wu, Yibiao Zhao, Jun-Yan Zhu, Siwei Luo, and Zhuowen Tu. MILCut: A sweeping line multiple instance learning paradigm for interactive image segmentation. In *CVPR*, pages 256–263, 2014.

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

[53] Qiong Yan, Li Xu, Jianping Shi, and Jiaya Jia. Hierarchical saliency detection. In *CVPR*, pages 1155–1162, 2013.

[54] Chuan Yang, Lihe Zhang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Saliency detection via graph-based manifold ranking. In *CVPR*, pages 3166–3173, 2013.

[55] Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. Understanding neural networks through deep visualization. In *ICMLW*, 2015.

[56] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*, pages 818–833, 2014.

[57] Matthew D Zeiler, Graham W Taylor, and Rob Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In *ICCV*, pages 2018–2025, 2011.

[58] Jianming Zhang, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. Top-down neural attention by excitation backprop. In *ECCV*, pages 543–559, 2016.