Regional Interactive Image Segmentation Networks

Jun Hao Liew[1] Yunchao Wei[2] Wei Xiong[3] Sim-Heng Ong[1,2] Jiashi Feng[2]
[1] NUS Graduate School for Integrative Sciences and Engineering, National University of Singapore
[2] Department of Electrical and Computer Engineering, National University of Singapore [3] Institute for Infocomm Research
liewjunhao@u.nus.edu {eleweiyv, eleongsh, elefjia}@nus.edu.sg wxiong@i2r.a-star.edu.sg

Abstract

The interactive image segmentation model allows users to iteratively add new inputs for refinement until a satisfactory result is finally obtained. Therefore, an ideal interactive segmentation model should learn to capture the user's intention with minimal interaction. However, existing models fail to fully utilize the valuable user input information in the segmentation refinement process and thus offer an unsatisfactory user experience. In order to fully exploit the user-provided information, we propose a new deep framework, called Regional Interactive Segmentation Network (RIS-Net), to expand the field-of-view of the given inputs to capture the local regional information surrounding them for local refinement. Additionally, RIS-Net adopts multiscale global contextual information to augment each local region for improving feature representation. We also introduce click discount factors to develop a novel optimization strategy for more effective end-to-end training. Comprehensive evaluations on four challenging datasets well demonstrate the superiority of the proposed RIS-Net over other state-of-the-art approaches.

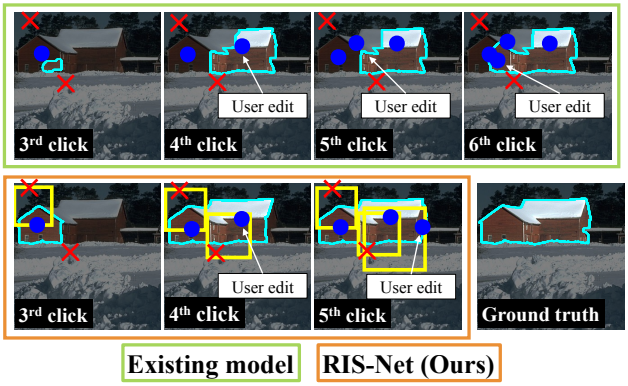


Figure 1: Demonstration of our motivation. Existing interactive segmentation algorithms do not fully utilize the user-given information during the refinement process, thus requiring substantial additional interactions from the user in order to obtain an accurate segmentation result (top row). In contrast, our proposed RIS-Net fully exploits the user-provided information by including the local regional context surrounding the clicks, leading to much faster refinement progress (bottom row). Best viewed in color.

1. Introduction

Interactive image segmentation is a popular research domain with many important applications, such as medical image analysis (*e.g.* interactive segmentation of brain tumor for treatment planning [2]), photo editing and image/video composition. Unlike semantic segmentation that partitions an image into multiple regions of pre-defined semantic categories, *interactive* image segmentation aims at extracting the object of interest based on user inputs.

The primary goal of interactive segmentation is to improve overall user experience by extracting the object of interest accurately with minimal user effort. The typical interactive image segmentation working flow is as follows: the user first provides positive and negative inputs to indicate the interested foreground and background; then the algorithm produces an initial output based on the input; and more user inputs are added for refinement until the segmentation result is satisfactory.

To achieve this, many algorithms have been proposed in the literature, such as graph cut [4, 22, 16], random walker [11], geodesic segmentation [1, 6], the combination of graph cut and geodesics [21], growcut [26], *etc*. However, all these conventional algorithms typically rely on low-level cues, such as colors, texture or other handcrafted features to predict foreground/background segmentation, leading to low accuracy in cases of similar foreground and background appearances, weak edges and cluttered background.

Recently, Xu *et al.* [29] proposed to use a deep fully convolutional network (FCN) [18] to solve the interactive segmentation problem, which we denote as iFCN to differentiate it from the FCN used in semantic segmenta-

2380-7504/17 $31.00 © 2017 IEEE
DOI 10.1109/ICCV.2017.297

tion [18, 5, 30, 27, 28]. Given positive and negative clicks that specify the foreground and background respectively, [29] transformed the user-provided clicks into Euclidean distance maps and concatenated them with the input image to train an end-to-end iFCN. Compared to traditional methods, the iFCN model has a higher level understanding of objectness and semantics, therefore leading to better segmentation quality.

Despite the outstanding performance of [29] over the conventional models, the iFCN often struggles to refine its prediction given additional inputs, thus requiring excessive user effort to produce desirable outputs (Figure 1). In contrast, an ideal interactive segmentation model should be capable of capturing the user's intention with minimal user interaction. In this regard, the user-provided inputs play an important role in guiding the segmentation process towards the desired output. However, existing models fail to fully utilize the valuable user input information in the refinement process and offer an unsatisfactory user experience.

To address this issue, we devise a new deep neural network model, called Regional Interactive Segmentation Network (RIS-Net), to fully utilize the user-provided information. By expanding the field-of-view of each click pair to a larger local region covering the object boundaries, RIS-Net exploits the regional context within these regions to refine the whole-image segmentation output. The proposed model can learn global contextual information to facilitate segmentation over each local region. We also introduce the click discount factor to develop an effective training strategy that enforces the model to decrease the loss more rapidly in the early stage yet enables end-to-end training of both whole-image-segmentation and local-region-refinement tasks. The key contributions of our work are summarized as follows:

- We propose a new architecture that exploits the local regional context around the user-provided inputs while performing segmentation, thus offering a stronger ability to refine local segmentations.
- We propose to use multiscale global contextual information to augment each local region and demonstrate experimentally that this strategy significantly improves the performance.
- We develop an effective end-to-end training pipeline based on the click discount factor.
- We achieve state-of-the-art results on Grabcut, Berkeley, Pascal VOC and MSCOCO datasets. On average, our proposed RIS-Net reduces the number of clicks required by the iFCN and the best performing conventional method on each dataset by 1.83 and 5.75 clicks respectively, which significantly reduces the amount of user interaction required for accurate segmentation.

2. Related Work

Early interactive image segmentation methods, such as the parametric active contour model [14] and intelligent scissors [20] mainly consider boundary properties when performing segmentation, thus performing poorly at weak edges. More recent interactive image segmentation algorithms are formulated based on graphical models. For instance, Boykov and Jolly [4] formulated interactive segmentation as a graph cut optimization problem and solved it using the min-cut/max-flow algorithm [3]. Bai and Sapiro [1] classified each pixel into foreground and background based on weighted geodesic distances. Grady [11] estimated the probability that a random walker at each unlabeled pixel will first reach one of the labeled pixels by formulating it as a combinatorial Dirichlet problem. To further improve the performance, shape priors such as [9, 25, 7, 12] have been considered in the literature. However, all these conventional methods typically rely on low-level cues, such as colors, texture or other hand-crafted features to predict foreground/background segmentation. Therefore, these models often give unsatisfactory results in cases of significantly overlapping foreground and background appearances, complex background or varying lighting conditions.

Recently, Xu *et al.* [29] proposed a deep-learning based algorithm to address the aforementioned problems by learning a deep representation. Despite its excellent performance over the conventional solutions, the approach typically struggles to correct its prediction by producing similar outputs regardless of additional clicks added. In contrast, our model attempts to fully utilize the user-provided information by attending to the local regions surrounding the clicks to refine the segmentation output, reducing the user efforts required to segment an object.

3. Proposed Method

Our proposed RIS-Net consists of the following two complementary branches: 1) a global branch producing coarse segmentation on the full image, and 2) a local branch performing refinement at the fine-grained local regions around the user clicks. After the global branch that outputs a coarse prediction over the whole image, the local branch processes each ROI generated based on user inputs and produces refined local segmentation. However, the restricted view of each ROI may pose a challenge to the refinement task. To address this problem, the proposed RIS-Net reuses global contextual information from the global branch to augment each ROI for better local segmentation. Finally, we fuse both the global and local predictions and combine the fused output with graph cut optimization [4] to produce the final segmentation result. The overall architecture is illustrated in Figure 2.

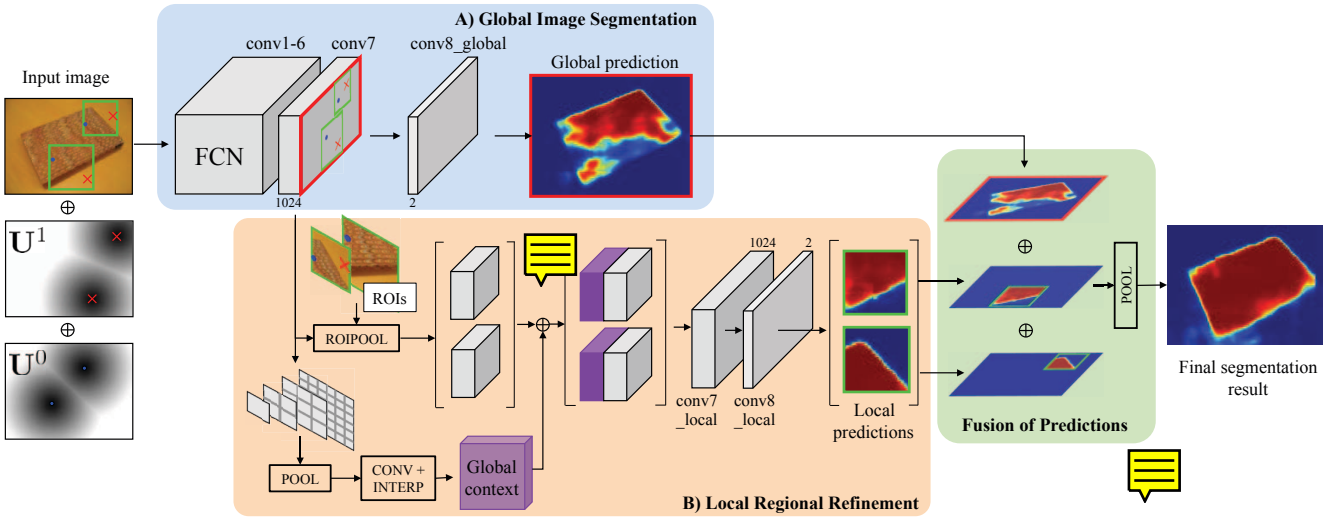


Figure 2: Overall architecture of Regional Interactive Segmentation Network (RIS-Net). It can be decomposed into two parts: a global branch for whole-image segmentation and a local branch for local regional refinement. In the global branch, we feedforward the input (image concatenated with the transformed clicks) to an FCN to obtain a coarse global prediction. In the local branch, we generate ROI proposals based on each click pair and extract a local descriptor for each ROI using the ROI pooling layer. Multiscale global context is appended to each local descriptor before entering convolution layers to produce local predictions. Finally, both global and local predictions are fused and combined with graph cut optimization (not shown here) to output the final segmentation result.

3.1. Global Image Segmentation

We first transform the positive and negative clicks into two Euclidean distance maps and concatenate them with the input image, forming a (image, user interaction) pair that serves as a 5-channel input. Formally, the positive click set $\mathcal{S}^1$ and the negative click set $\mathcal{S}^0$ are transformed into two distance maps, $\mathbf{U}^1$ and $\mathbf{U}^0$ using a Euclidean distance transformation:

$$u^k(p) = \min_{\forall q \in \mathcal{S}^k} \text{dist}(p, q), \quad \forall k \in \{0, 1\}, \tag{1}$$

where $u^k(p)$ refers to the value of $\mathbf{U}^k$ at pixel location p, while $\text{dist}(p, q)$ refers to the Euclidean distance between pixel locations p and q.

As for the FCN in the global branch, we employ the state-of-the-art DeepLab-LargeFOV architecture [5], which we refer to as DeepLab in the rest of this paper. DeepLab is a fully convolutional variant of the VGG-16 network [24] with filter dilation and reduced dimensionality at the fully connected layers (to be precise, these layers are convolution layers instead). We denote the conv8 layer as conv8_{global} to differentiate it from the conv8_{local} used in the local region refinement branch. Note that our proposed interactive segmentation model is a universal one and any other FCN architecture may be employed.

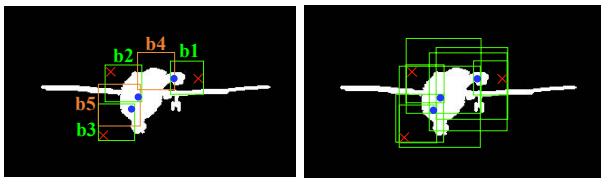
3.2. Local Regional Refinement

In the local branch, our RIS-Net first generates N_{ROI} regions of interest based on the user clicks. For each ROI, besides local representation, we also append global contextual information to each ROI before passing it to convolution layers for segmentation.

3.2.1 Sampling of ROI Proposals

Following the sampling strategy suggested in [29], we first randomly sample $n_{pos} \in [1, N_{pos}]$ positive and $n_{neg} \in [0, N_{neg}]$ negative clicks for each image to simulate user interactions for training. To generate ROI proposals, we find the nearest negative click for each positive click and construct a bounding box whose size is equal to the distance between the clicks pair. Since the ROI pooling layer takes an input of arbitrary size and produces a fixed-size output (e.g. 41×41), a rectangular ROI may result in unwanted deformed content which may harm the performance. To avoid this, we explicitly constrain the ROI to be square by resizing the shorter side to be the same as the longer side.

Since the number of ROIs is essentially limited by the number of positive clicks available in each image, there is often not enough ROIs for training. To address this, we propose a sliding-based sampling scheme to sample additional ROIs given the previously sampled ROIs. For instance, as shown in Figure 3 (a), we first sample 3 ROIs based on the nearest clicks pairs (indicated by the green boxes). Then, we sample extra ROIs between the neighboring ROIs (indicated by the orange boxes) such that the total number of ROIs can be increased to N_{ROI} (we use $N_{ROI} = 5$ in this work). A general rule is to sample additional ROIs along



(a) (b)

Figure 3: (a) Sliding-based sampling scheme. A bounding box "b1" slides along the object boundaries towards its neighboring ROI "b2" to generate additional ROI "b3" for training. (b) Sampling based on all combinations of click pairs often yields large ROIs that share a similar receptive field as the whole image.

the *shorter* path between the neighboring ROIs such that the newly sampled ROIs do not deviate too much from the sampled clicks.

There are essentially two benefits of using the sliding-based sampling scheme here. Firstly, this solves the problem of insufficient ROIs for training. Secondly, as compared to the ROIs sampled using all the combinations of click pairs (Figure 3 (b)), the sampled ROIs using the sliding-based approach typically cover the local fine-grained details much better.

We use all N_{ROI} ROIs for training the local branch. In the case where there is only one positive click (implying only one ROI is available), we simply set all ROIs for this image to be the same. During testing, in the early stages where there is no negative click, we pass the whole image as the only ROI to our network. In the latter stages, when there are more than N_{ROI} positive clicks, we either process all the ROIs or randomly sample the maximum allowable number of ROIs depending on the available GPU memory.

3.2.2 Global Context-Aware Regional Segmentation

Global Contextualization: Since we restrict the receptive field of each region proposal to a narrow and incomplete region, this could be challenging for local refinement without knowing what the object of interest is. To address this problem, we propose to add global contextual information to each local representation to enrich the feature representations before passing them to convolution layers for segmentation.

Since an object may occur at different scales in images, the ideal global context should be capable of summarizing the image content at multiple scales to handle scale variability in segmentation. Inspired by [30], we use the pyramid pooling to extract hierarchical global cues from the conv7 feature map in the global branch. In particular, we use a four-level pyramid pooling with feature map size of 1×1, 2×2, 3×3 and 6×6 respectively. At each level, an average pooling kernel is applied to extract a smaller feature map with the corresponding bin size, followed by a 1×1 convolution layer (for dimension reduction), an upsampling layer and a concatenation layer to produce the global contextual prior for each local ROI.

Local Regional Segmentation: The RIS-Net first uses an ROI pooling layer to extract a local regional representation of each ROI from the conv7 feature map in the global branch. Then, each local ROI is concatenated with the multiscale global context before being passed to other two convolution layers (conv7$_{local}$ and conv8$_{local}$) to generate a local segmentation. Similarly, the ROI pooling layer is used to extract the corresponding cropped ground truth for training, resulting in multiple local losses per image.

3.3. Deep Supervision for Training RIS-Net

Shared Computation: Our proposed RIS-Net optimizes both global image segmentation and local regional refinement objectives jointly in an end-to-end manner. To reduce the computational complexity, the conv1_1 to conv7 layers are shared across both branches to improve the inference speed. Within the local branch, the conv7$_{local}$ and conv8$_{local}$ layers are shared across all ROIs with one local loss per ROI. Due to the shared computation between both tasks and the end-to-end pipeline, the gradients from the local branch can be backpropagated to conv7 and the preceding layers, resulting in an improved representation for the whole-image segmentation problem.

Click Discounting Factor for Training: Inspired by the reinforcement learning algorithms, we devise a new training scheme that incorporates a click discount factor to each ROI in the local branch such that the latter ROI receives less reward, enforcing the model to use minimal amount of user interaction for refinement. The details of the training process are described below: 1) a global coarse prediction is first obtained from the global branch and the difference of global prediction and ground truth is computed; 2) within the local branch, the ROI with the largest segmentation error is first chosen to output the corresponding local prediction; 3) the global and local predictions are fused (details will be given in the next subsection) and the difference of the fused output and ground truth is computed; 4) the next ROI with the second largest segmentation error is selected to output the corresponding prediction. Steps 3) and 4) are repeated until all the ROIs are selected. For each image, the total loss is defined as

$$\mathcal{L}_{total} = \mathcal{L}_{global} + \sum_{t=1}^{N_{ROI}} \gamma^{t-1} \exp\left(-\Delta\mathcal{L}_t\right) \qquad (2)$$

where $0 \leq \gamma \leq 1$ is the discount factor. Here, the Δ term denotes how much the local loss is decreased by adding a new ROI (*i.e.* the reward). The latter added ROI will get less

reward on its contribution to the decrease in loss. Therefore, the training process enforces the model to decrease the loss more rapidly at the first few iterations. Note that $\gamma = 1$ denotes the case where all the ROIs from the same image are used simultaneously for training whereas $\gamma = 0$ reduces to the case where there is no local branch.

3.4. Fusion of Global and Local Segmentation

Once we obtain the global and local prediction maps in a single forward pass, we devise a new approach to fuse the two to produce the final prediction for both training and testing. Given N ROIs, let f_i be the local prediction map of the ith ROI and f_{N+1} be the global prediction map. We first insert each f_i back to its corresponding image space with zero padding outside f_i and we denote this as F_i. Then, we aggregate all the outputs using a max-pooling operation:

$$P(x, y) = \max_i (F_i(x, y)), \quad \forall i. \tag{3}$$

It should be noted that the max pooling operation can also be replaced by average pooling. We choose the former in our work since we empirically find that it performs slightly better.

4. Experiments

4.1. Datasets and Settings

Datasets: We evaluate the performance of the RIS-Net and compare it with the state-of-the-arts on four public datasets.

GrabCut dataset [22]: The GrabCut dataset consists of 50 natural images with ground truth. This dataset has been used as a common benchmark for most popular interactive segmentation algorithms.

Berkeley dataset [19]: This dataset contains 100 single-object images chosen from the Berkeley dataset to represent various challenges encountered in interactive segmentation, including similar foreground/background appearances, texture, existence of multiple similar objects, *etc*.

Pascal VOC dataset [8]: We use 1,449 validation images in the Pascal dataset that are not used in training. Note that all the categories in this dataset have been included in our training set.

MSCOCO dataset [17]: MSCOCO contains 80 different object categories, where 20 of them are the same as the Pascal dataset (these 20 are called "seen" categories, and the rest are "unseen" categories). For fair comparison with [29], we also split the dataset into 20 seen categories and 60 unseen categories, and randomly sample 10 images per category for evaluation.

Evaluation Metrics: We follow [15] by running an active robot user that simulates the user behavior when evaluating an interactive segmentation model. This is also widely used in other works [12, 29] for performance evaluation.

The evaluation begins with a single positive click placed at the center of the object of interest. The model then outputs an initial prediction based on this input. Subsequent clicks are iteratively added to the middle of the largest mislabeled regions and this step is repeated until the maximum number of clicks (20) is achieved. To evaluate, we record the IU accuracy of the model at each click. As in [29], we also record the number of clicks required to achieve a certain IU accuracy on a given dataset. If the IU accuracy cannot be achieved within 20 clicks, it will be thresholded to 20. Each metric reported is averaged over all images in a dataset. Note that we follow [29] by using different IU thresholds for different datasets (*e.g.* 90% for Grabcut and 85% for Pascal) for fair comparison.

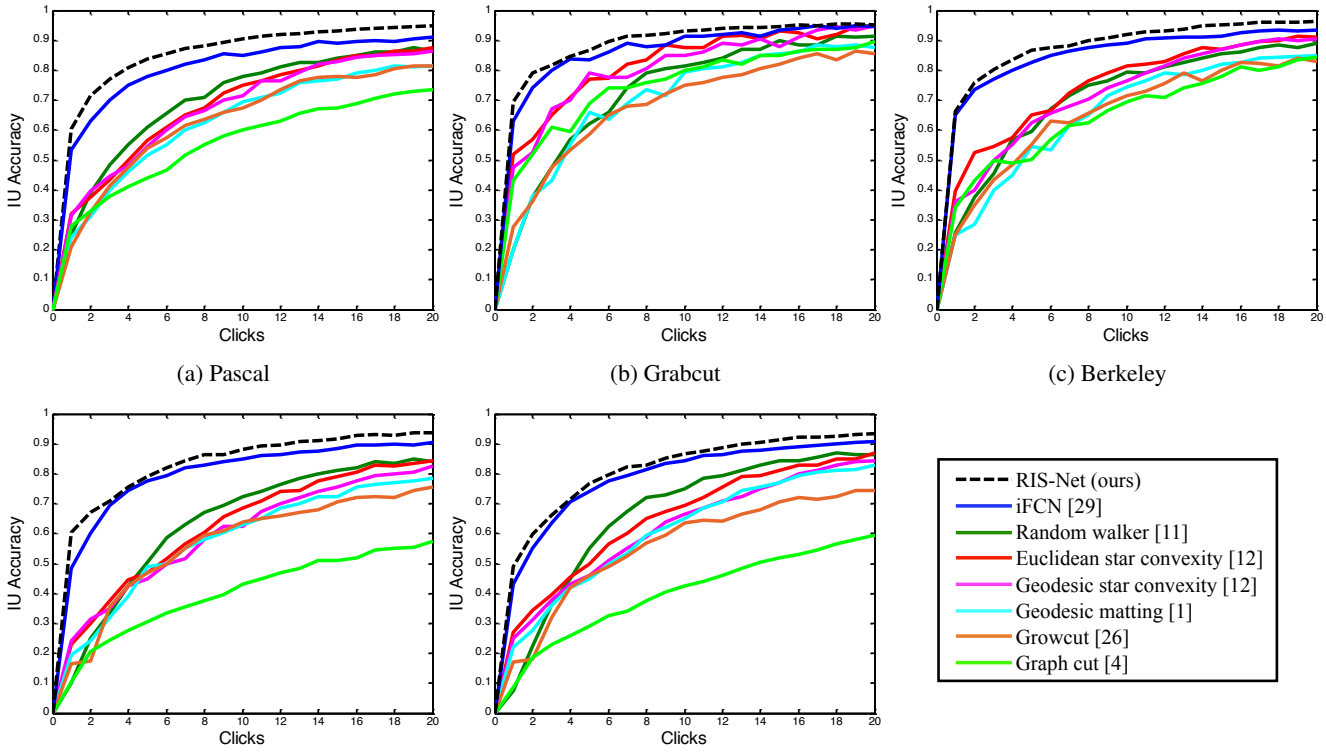
Training Details: We use the same sampling strategy as in [29] to sample clicks for training. All the 1,464 training images from the PASCAL VOC segmentation dataset [8] are used to sample 15 (image, user interaction) pairs per image, generating about 80k training samples (including the flipped version).

For our proposed RIS-Net, we fix the size of the pooling output after the ROI pooling layer to be 41×41. We set the discount factor, γ to 0.8. The region proposals for training are pre-computed. All networks are initialized from VGG-16 weights pre-trained on ILSVRC 2012 [23]. The weight matrices of the newly added layers are randomly initialized from a Gaussian distribution with standard deviation of 0.01. We train our model using stochastic gradient descent with a batch size of 2 images and 5 ROIs per image. We fix the momentum to 0.9 and weight decay to 0.0005 throughout the training process. We train all models for roughly 20 epochs with a fixed base learning rate of 10^{-3}, reducing the rate by $10\times$ after every 5 epochs. We implement our model based on the Fast R-CNN [10] framework. All our experiments are conducted on the Caffe framework [13]. All networks are trained on a single NVIDIA Pascal Titan X GPU with 12GB memory.

Our proposed RIS-Net typically takes about 0.4 seconds for a 640×480 color image with 6 user inputs (3 positive and 3 negative clicks) on a Pascal Titan X GPU while the graph cut optimization takes about 0.25 seconds on modern CPUs. Therefore, our proposed method is suitable for real-time applications.

4.2. Comparison with State-of-the-arts

Figure 4 and Table 1 show the comparison results with several state-of-the-art methods over four datasets. Firstly, from Figure 4, we can see that the deep learning-based models (RIS-Net and iFCN [29]) outperform the traditional methods by a large margin, demonstrating the effectiveness of deep representation over the low-level cues for segmentation. Secondly, we see that our proposed RIS-Net con-



(a) Pascal (b) Grabcut (c) Berkeley

(d) MS COCO seen categories (e) MS COCO unseen categories (f) Legend

Figure 4: The plot of the mean IU accuracy against the number of clicks on different datasets. The legend of these plots is shown in (f).

Segmentation models	Pascal (85% IU)	Grabcut (90% IU)	Berkeley (90% IU)	MSCOCO seen categories (85% IU)	MSCOCO unseen categories (85% IU)
Graph cut [4]	15.06	11.10	14.33	18.67	17.80
Geodesic matting [1]	14.75	12.44	15.96	17.32	14.86
Random walker [11]	11.37	12.30	14.02	13.91	11.53
Euclidean star convexity [12]	11.79	8.52	12.11	13.90	11.63
Geodesic star convexity [12]	11.73	8.38	12.57	14.37	12.45
Growcut [26]	14.56	16.74	18.25	17.40	17.34
iFCN [29]	6.88	6.04	8.65	8.31	7.82
RIS-Net (ours)	**5.12**	**5.00**	**6.03**	**5.98**	**6.44**

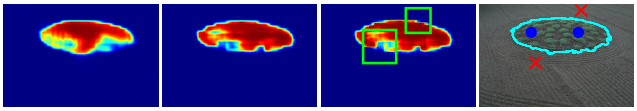
Table 1: The mean number of clicks required to achieve the specific IU accuracy on different datasets. The best results are highlighted in **bold**.

sistently outperforms the iFCN on all four datasets. For example, on the Grabcut dataset, our proposed RIS-Net requires fewer than 4 clicks to reach the performance of iFCN with 6 clicks. Similarly, on the Berkeley dataset, our proposed RIS-Net requires 2.62 fewer clicks than the iFCN to reach 90% of IU accuracy. Furthermore, we can also see that our algorithm achieves higher IU accuracy at every step and increases the IU accuracy much faster than other algorithms during refinement. This is because at every step, our model learns to fully utilize the local regional information surrounding the new clicks, therefore accelerating the refinement process.

Single-click Performance: Although the proposed RIS-Net is formulated based on at least two clicks (a positive and

(a) (b) (c)

Figure 5: Segmentation results using a single click. The images selected represent 3 different segmentation challenges: (a) textured foreground, (b) changing illumination, (c) overlapping foreground and background appearances.



(a) (b)(i) (b)(ii) (c)

Figure 6: Network prediction by a model with (a) global branch only, (b) both global and local branches (RIS-Net) and (c) the corresponding ground truth. The (b)(i) and (b)(ii) show the prediction before and after the fusion of global and local segmentations.

a negative click) provided for local refinement, we demonstrate that our algorithm produces very good predictions even with just a single click (Figure 5). On average, the RIS-Net achieves about 61% IU accuracy using a single positive click, which is 6.5% and 26.2% higher than the iFCN and the best performing traditional model on each dataset, based on the IU accuracies of other methods reported in [29].

Amount of User Interaction Required: In Table 1, we also observe that RIS-Net requires the fewest clicks to achieve a particular IU accuracy on all the datasets. On average, our proposed RIS-Net reduces the number of clicks required by the iFCN and the other best performing traditional model on each dataset by 1.83 and 5.75 clicks respectively, significantly reducing the amount of user effort.

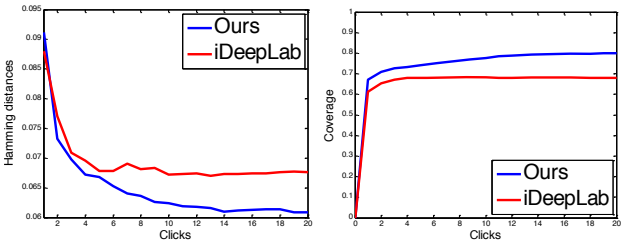
4.3. Ablation Experiments

In Table 2, we quantitatively analyze the effect of each component in our proposed network to justify our design choices. All the numbers reported in this section denote the number of clicks required to achieve 90% IU accuracy on the Berkeley dataset.

We train a new baseline model with the DeepLab-LargeFOV model and denote it as iDeepLab. We observe that this results in a slight improvement over [29]. Adding global contextual further enhances the performance by reducing the number of clicks needed by 0.63 clicks. This indicates the importance of global cues for the segmentation task. Introducing the local branch to the model further reduces the clicks number needed by another 0.75 click. Here,

Global context?	Local branch?	Discount factor?	# Clicks (90%IU)
			7.60
✓			6.97
✓	✓		6.22
✓	✓	✓	**6.03**

Table 2: Effects of local branch, global contexts and click discount factor. The check mark is used to indicate if a particular component is used. All the numbers reported denote the number of clicks required to achieve 90% IU on the Berkeley dataset.



(a) Hamming distance (b) Coverage score

Figure 7: Hamming distance and coverage score against clicks number on the Berkeley dataset. The smaller Hamming distance implies a closer match between the prediction and the ground truth whereas the higher coverage score suggests a higher recall rate of the model.

we also provide a visual comparison to illustrate the effects of the local branch. As shown in Figure 6 (b)(ii), the local branch "diffuses" its prediction within each ROI during refinement, increasing its confidence of prediction along the boundaries. By comparing Figure 6 (a) and (b), we observe that the local branch also helps optimize the whole-image segmentation task concurrently since the shared computation allows the gradient from the local branch to be back-propagated to the same conv7 layer trained for global image segmentation. Finally, we observe that the click discount factor allows effective training of the RIS-Net, leading to a further reduction in the number of clicks required to 6.03.

To ensure a fair comparison, all the evaluation metrics reported above are based on the segmentation output combined with graph cut optimization following [29]. Here we conduct more experiments to analyze more detailedly the performance of RIS-Net by isolating the effect of the graph cut. Let x_i and y_i denote the prediction map and ground truth at pixel location i respectively. We use the following metrics to further evaluate the RIS-Net: (i) Hamming distance: $(\sum_{i=1}^{N} x_i \cdot (1-y_i) + (1-x_i) \cdot y_i)/N$ and (ii) coverage score: $C = \sum_{i=1}^{N}(x_i \cdot y_i)/\sum_{i=1}^{N} y_i$. Due to space limitation, we provide justification on choosing these three metrics in the supplementary materials instead.

Figure 8: Qualitative comparison between the baseline and our model given the same set of user interactions. The positive and negative clicks are denoted by blue dots and red crosses respectively. Object boundaries are highlighted in cyan.

Figure 9: Some failure cases by RIS-Net.

The results are summarized in Figure 7. In terms of Hamming distance, our algorithm has an overall smaller Hamming distance compared to the iDeepLab, implying that the prediction generated by our algorithm matches the ground truth more closely. Moreover, our algorithm has a consistently higher coverage score compared to the iDeepLab. We also notice that the iDeepLab has a nearly constant coverage score with respect to the number of clicks used, suggesting that the improved performance given more clicks is due to the use of graph cut optimization for better boundaries localization. On the other hand, the increasing coverage score of our algorithm indicates a form of information propagation towards the local regions covering the clicks for refinement.

We also present some qualitative comparisons between our model and other state-of-the-arts given the same set of user interactions (Figure 8). In general, our proposed RIS-Net produces relatively good predictions by exploiting both local regional and global multiscale context to assist the local segmentation. We also show some failure cases in Figure 9 and find that our RIS-Net has difficulties in segmenting occluded or hairy objects.

5. Conclusion

In this work, we proposed a region-based interactive image segmentation model that fully exploits the local regional context highlighted by the user-provided clicks to accelerate the refinement progress. We appended global contextual information to each local ROI for improved feature representation. We also devised a new training pipeline based on click discounting factor for effective end-to-end training of our model. Comprehensive evaluations on four challenging datasets have well demonstrated the superiority of our algorithm against the state-of-the-art methods.

Acknowledgement

The work of Jiashi Feng was partially supported by NUS startup R-263-000-C08-133, MOE Tier-I R-263-000-C21-112 and IDS R-263-000-C67-646. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Pascal Titan X GPU used for this research.

References

[1] X. Bai and G. Sapiro. Geodesic matting: A framework for fast interactive image and video segmentation and matting. *International Journal on Computer Vision*, 82(2):113–132, 2009. 1, 2, 6

[2] N. Birkbeck, D. Cobzas, M. Jagersand, A. Murtha, and T. Kesztyues. An interactive graph cut method for brain tumor segmentation. In *Applications of Computer Vision (WACV), 2009 Workshop on*, pages 1–7, 2009. 1

[3] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 26(9):1124–1137, 2004. 2

[4] Y. Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In *IEEE International Conference on Computer Vision*, volume 1, pages 105–112, 2001. 1, 2, 6

[5] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and 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*, 2017. 2, 3

[6] A. Criminisi, T. Sharp, and A. Blake. Geos: Geodesic image segmentation. In *European Conference on Computer Vision*, pages 99–112, 2008. 1

[7] P. Das, O. Veksler, V. Zavadsky, and Y. Boykov. Semiautomatic segmentation with compact shape prior. *Image and Vision Computing*, 27(1):206–219, 2009. 2

[8] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal on Computer Vision*, 88(2):303–338, 2010. 5

[9] D. Freedman and T. Zhang. Interactive graph cut based segmentation with shape priors. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 755–762. IEEE, 2005. 2

[10] R. Girshick. Fast r-cnn. In *IEEE International Conference on Computer Vision*, pages 1440–1448, 2015. 5

[11] L. Grady. Random walks for image segmentation. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 28(11):1768–1783, 2006. 1, 2, 6

[12] V. Gulshan, C. Rother, A. Criminisi, A. Blake, and A. Zisserman. Geodesic star convexity for interactive image segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3129–3136, 2010. 2, 5, 6

[13] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 5

[14] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal on Computer Vision*, 1(4):321–331, 1988. 2

[15] P. Kohli, H. Nickisch, C. Rother, and C. Rhemann. User-centric learning and evaluation of interactive segmentation systems. *International Journal on Computer Vision*, 100(3):261–274, 2012. 5

[16] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum. Lazy snapping. In *ACM Transactions on Graphics*, volume 23, pages 303–308, 2004. 1

[17] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755, 2014. 5

[18] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. 1, 2

[19] K. McGuinness and N. E. Oconnor. A comparative evaluation of interactive segmentation algorithms. *Pattern Recognition*, 43(2):434–444, 2010. 5

[20] E. N. Mortensen and W. A. Barrett. Intelligent scissors for image composition. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 191–198, 1995. 2

[21] B. L. Price, B. Morse, and S. Cohen. Geodesic graph cut for interactive image segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3161–3168, 2010. 1

[22] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM Transactions on Graphics*, volume 23, pages 309–314, 2004. 1, 5

[23] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal on Computer Vision*, 115(3):211–252, 2015. 5

[24] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3

[25] O. Veksler. Star shape prior for graph-cut image segmentation. In *European Conference on Computer Vision*, pages 454–467, 2008. 2

[26] V. Vezhnevets and V. Konouchine. Growcut: Interactive multi-label n-d image segmentation by cellular automata. In *Proceedings of GraphiCon*, volume 1, pages 150–156, 2005. 1, 6

[27] Y. Wei, J. Feng, X. Liang, M.-M. Cheng, Y. Zhao, and S. Yan. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1568–1576, 2017. 2

[28] Y. Wei, X. Liang, Y. Chen, X. Shen, M.-M. Cheng, J. Feng, Y. Zhao, and S. Yan. Stc: A simple to complex framework for weakly-supervised semantic segmentation. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 2016. 2

[29] N. Xu, B. Price, S. Cohen, J. Yang, and T. S. Huang. Deep interactive object selection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 373–381, 2016. 1, 2, 3, 5, 6, 7

[30] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2881–2890, 2017. 2, 4