



# Interactive Deep Editing Framework for Medical Image Segmentation

Bowei Zhou<sup>1,2</sup>, Li Chen<sup>1,2</sup>(✉), and Zhao Wang<sup>1,2</sup>

<sup>1</sup> School of Software, Tsinghua University, Beijing 100084, China  
chenlee@tsinghua.edu.cn

<sup>2</sup> Beijing National Research Center for Information Science and Technology  
(BNRist), Beijing, China

**Abstract.** Deep neural networks exhibit superior performance in dealing with segmentation of 3D medical images. However, the accuracy of segmentation results produced by fully automatic algorithms is insufficiently high due to the complexity of medical images; as such, further manual editing is required. To solve this problem, this paper proposes an interactive editing method combined with 3D end-to-end segmentation network. In the training stage, we simulate the user interactions, which are used as training data, by comparing the segmentation automatically generated by convolutional neural network with the ground truth. User interactions are fed into the network along with the images, allowing the network to adjust the segmentation results based on user edits. Our system provides three editing tools for smartly fixing segmentation errors, which cover most commonly used editing styles in medical image segmentation. With the high-level semantic information in the network, our method can efficiently and accurately edit the 3D segmentation. The interactive editing experiments on the BraTS dataset show that our method can significantly improve the segmentation accuracy with a small number of interactions only. The proposed method presents potential for clinical applications.

**Keywords:** Interactive image segmentation · Medical image segmentation · Deep learning

## 1 Introduction

Deep neural networks have greatly promoted the development of research on medical image processing in recent years. However, fully automatic methods do not always generate correct segmentation. As such, these methods cannot be applied directly to clinics, and the segmentation needs to be further edited manually by using other methods.

Editing the segmentation of a medical image is often difficult and time consuming. When using traditional 2D image segmentation techniques, such as graph cuts [1], the foreground and background regions should be marked slice

by slice. The problem of tumor segmentation is also complicated because the boundary between the tumor and surrounding tissues is often not obvious and requires expert experience to decide. In this regard, traditional methods like JF-Cut [7] that rely on low-level image features are ineffective, and users can only use tools, such as a paintbrush, to refine the boundaries of the tumor on each slice. Smarter editing methods should be developed to reduce the complexity of the interaction.

Deep neural network can extract high-level semantic information of an image, which can be a solution. However, training deep learning network to learn manual editing is difficult. One of the challenges is the lack of interactive editing data for training. We do not know what segmentation error the user will face and what feedback is made when actually using it. Active learning [11] is one of the ways to solve this problem, which can reduce the cost of labeling; however, doctors still need to participate in the training stage. Several studies have attempted to train interactive segmentation networks directly. Refs. [10] and [5] implement interactive segmentation of 2D natural images based on FCN, in which users can select objects in images by clicking with the mouse. These works focus on segmentation rather than editing the existing segmentation. In [8], the user can fine-tune the segmentation result by scribbling. Another work [9] proposes an interactive editing method to improve the segmentation results by combining geodesic distance transform with CNN. In this design, the prediction result of each round is fed back into the network, which may cause the network to over-fit this channel, making it difficult for the interaction to affect the existing segmentation.

In view of the lack of training data and easy over-fitting for segmentation editing, this paper proposes an interactive segmentation editing framework based on end-to-end CNN. We compare the difference between the ground truth and the segmentation generated by the fully automated method. We then generate simulated user interactions, which are used to train the segmentation editing network. We have designed three editing tools for different editing purposes that medical image segmentation may need. When using the system, the user can refine the segmentation result through the provided tools. Editing is repeated until the segmentation accuracy meets the requirements.

We used the BraTS 2018 Challenge dataset [6] to segment the whole tumor regions from the multi-modality MRI images. Experiments on the three editing tools show that our method can significantly improve the segmentation accuracy by using a small number of interactions. The developed method exhibits potential in clinical applications.

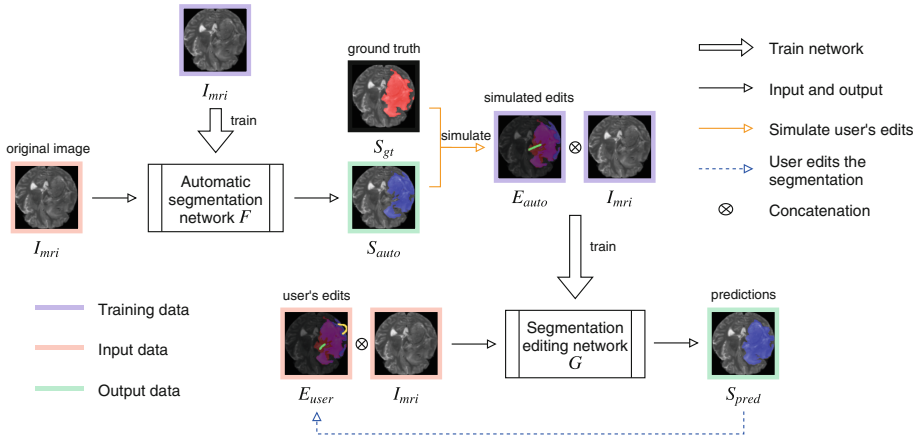
## 2 Methods

### 2.1 Overview

The overview of the system is shown in Fig. 1. In the training stage, we first use the automatic segmentation network  $F$  to predict the tumor segmentation of all images in the dataset. Given that it is a fully automatic algorithm, the

segmentation is not accurate enough, which has some differences compared with the ground truth. Users tend to edit these mis-segmented regions while using so we simulate user's edits in these regions according to the user's editing habits. The simulated user's edits are transformed into a volume data  $E_{auto}$ , which is used to train the segmentation editing network  $G$ . The network  $G$  takes the user's edits and the original image as input. Here, the volume data of edits are used as additional channels concatenated with the original image.

When using the system, the network  $G$  will first predict an initial segmentation  $S_{pred}$ . The user looks for the segmentation errors and adds edits  $E_{user}$ . The network  $G$  then generates more precise segmentation based on  $E_{user}$ . The user can repeat this step until the segmentation accuracy meets the requirements.



**Fig. 1.** Overview of our system. Automatically simulated edits  $E_{auto}$  are used to train the segmentation editing network  $G$ . For a given input image  $I_{mri}$ ,  $G$  will first generate an initial segmentation. And then, it will refine the segmentation according to user's edits.

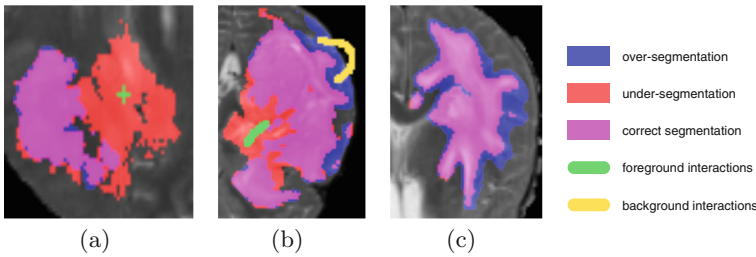
## 2.2 Interaction Design

In a well-designed segmentation editing system, the form of user's interaction should be as simple as possible, and the number of interactions required should be as few as possible. Based on these principles, combined with the practical application scenarios, we have designed three segmentation editing tools as follows.

1. **Selection tool.** Users can use this tool to click on a large block of mis-segmented region to fill/erase it. In an end-to-end segmentation network, if the deep layers fail to recognize the tumor region, then the segmentation of the tumor will not be restored in the decoder part, resulting in mis-segmentation (Fig. 2(a)).

2. **Brush tool.** This tool is used to refine medium size segmentation errors at the segmentation boundaries. Users need to use the brush to mark on these regions by scribbling. This error is common, and one possible cause is a change in the image intensity value within the tumor (Fig. 2(b)).
3. **Adjustment tool.** This tool is used to **intelligently expand or shrink the segmentation boundary**. When in use, the user inputs a scalar value, which indicates how much expansion/shrinkage is required. Since the boundaries of tumors are often unclear, the tumor is sometimes over-segmented or under-segmented slightly on boundaries, which is difficult to refine by the two other tools (Fig. 2(c)).

Users can use these tools together. All of these interactions are performed on 2D image slices but can be propagated to the 3D volume, which makes it unnecessary for users to edit on every slice.



**Fig. 2.** Usage scenarios of the three tools. (a) Selection tool for filling/erasing large block of mis-segmented regions. (b) Brush tool for refining the boundary. (c) Adjustment tool expanding or shrinking the boundary.

### 2.3 Simulate the Interactions

The automatically generated segmentation  $S_{auto}$  is used to compare with the ground truth  $S_{gt}$  to simulate the user's edit  $E_{auto}$ . To improve the editing effect, this simulation should be as close to the user's actual operating behavior as possible. The corresponding simulation generation methods of the three editing tools we designed are as follows.

**Selection Tool.** When users use the selection tool, they **tend to click on the center of a large block of mis-segmented region**. Basing on this assumption, we try to find the center of each mis-segmented region as a simulation for editing. We subtract  $S_{auto}$  from  $S_{gt}$ . **The mis-segmented region contains a number of 3D connected components**. For **large connected components** among them, we apply 3D erode operations repeatedly until the block is reduced to a core. We then choose one core voxel for each large connected component as a simulated point that the user is likely to click.



**Brush Tool.** Since the user’s strokes are performed on the axial slices, the simulation is also for this direction. We find the slice that has the largest 2D mis-segmented region and find the region on the slice. For this region, or connected component, we find the two points that are furthest apart and find the shortest path between them. This shortest path acts as a stroke that simulates the user’s scribble.

**Adjustment Tool.** This tool is designed to intelligently expand or shrink the current segmentation result. The user should be able to give different levels of adjustment (i.e., different input values), depending on the current degree of under-segmentation or over-segmentation. Conversely, to simulate the value that the user adjusts, we need to quantify the degree of mis-segmentation. We simply use the following equation for quantification:  $K = (V_{over} - V_{under})/V_{gt}$ . Here,  $V_{over}$ ,  $V_{under}$ , and  $V_{gt}$  represent the volumes of the over-segmented, under-segmented, and ground truth regions, respectively.

## 2.4 Implementation Details

The segmentation network used in this paper was proposed by Isensee et al. [3], which is an end-to-end 3D segmentation network. Our method does not assume the specific structure of the network and is theoretically applicable to other U-Net-like end-to-end segmentation networks, as confirmed by our experiments on 3D-UNet [2]. However, our method is unsuitable for patch-based segmentation networks, such as deepmedic [4], because of the strong dependence on global context information.

Overfitting is a problem to consider when training a segmentation editing network. Since user interaction is a strong cue in the segmentation problem, using too many interactions to train the network may cause the network to rely too much on them. This practice leads to the degradation of the basic segmentation ability of the network. The editing system would become less intelligent, requiring users to add a large number of edits. Therefore, only a small amount of necessary interactions are used to train the network. For the selection tool, we remove small 3D connected components. For the brush tool, we ignore short strokes and generate at most one stroke in each image. For the adjustment tool, we truncate the  $K$  values to  $\pm 0.5$  to avoid the influence of excessive  $K$  values on the network. We do not limit the adjustment range when users use this tool.

The edit information is commonly transformed into images and feed directly into the network as additional channels [5, 9, 10]. For the selection tool and the brush tool, we apply a two-voxel dilation operation for the voxels passed by the selected points and strokes, as the editing channels. Compared with the Euclidean distance [10] and the geodesic distance transform [9], the proposed method enables the network to receive more specific “editing signals”, which makes the editing more accurate. In these two channels, the foreground (under-segmentation) interaction and the background (over-segmentation) interaction have values 1 and  $-1$ , respectively, and the other regions are set to 0. For the adjustment tool, we simply set the value of all voxels in the channel to  $K$ .



In the testing stage, the real user interaction is fed into the network. Unlike the training stage, we do not limit the number of user interactions. Users can draw more than one stroke and can also set a K value outside the range of  $\pm 0.5$ .

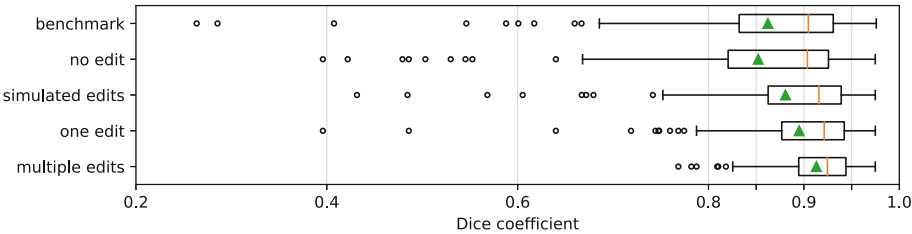
### 3 Experiments

The experiments are performed on the BraTS 2018 Challenge [6] dataset, which has 285 subjects. Each subject contains MRI image data of four modalities of T1, T1Gd, T2 and FLAIR. All the subjects are annotated by experts for the tumor regions. In our experiments, 50% of the data was used for training and the rest was used for testing.

#### 3.1 Results

We performed an interactive segmentation editing test on 143 subjects of test data. Since our method is based on the work of Isensee et al. [3], this network is also used as a benchmark method for comparison. For the segmentation editing network, we did four experiments: (1) do not edit, (2) add simulated edits (introduced in Sect. 2.3), (3) add no more than one edit, (4) add mutiple edits.

Figure 3 is the box plot of the Dice coefficient scores of these experiments. The initial prediction of the segmentation editing network is slightly less accurate than the benchmark method, which reflects the network's dependence on user's edits. After manually adding one edit, the average Dice coefficient has increased from 0.8622 to 0.8949 compared to the baseline. And after adding multiple edits, it is 0.9129, which is higher than DeepIGeoS's 0.8993 [9]. The box plot shows that multiple edits have a significant improvement over the segmentation which are originally inaccurate. It is reflected in the great increase of the lower quartile position and the reduction in outliers. In the experiment, the initial segmentation accuracy of about 1/3 of the test data is high enough, and for these data, we have not added edits. The average number of interactions for all test data is 2.68, and it takes about 1–2 min to complete a segmentation edit of each data.



**Fig. 3.** Box plot of the Dice coefficient scores on different experiments. (a) The benchmark network, (b) segmentation editing network  $G$  without edits, (c) add simulated edits, (d) add no more than one edit, (e) add mutiple edits (average 2.68 edits per image).

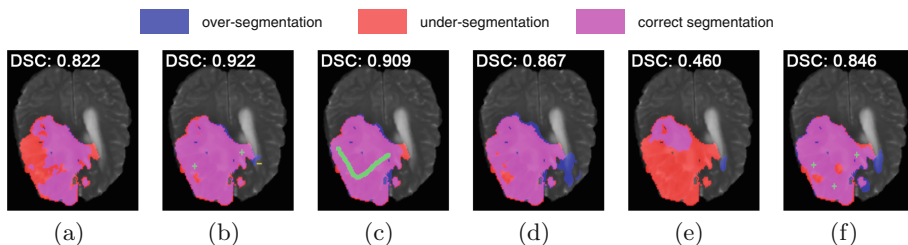
### 3.2 Examples

Figure 4 shows the results of refining a complex tumor image using the three tools separately. We compared the results of using 3D-UNet as the base network as well, where the selection tool also works well (Fig. 4(e), (f)).

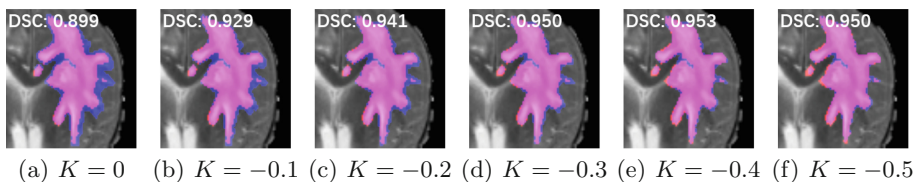
Figure 5 shows an example of using the adjustment tool to set a series of different values. The over-segmented region in this example is difficult to edit using the other two tools. It can be seen that with the decrease of the input value, the segmentation region also shrinks correspondingly.

### 3.3 Robustness Testing

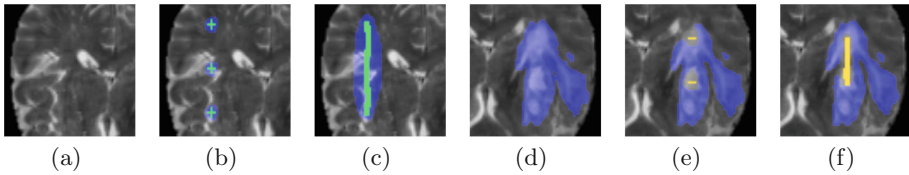
To verify the robustness of the editing tools, we tested the effects of the selection tool and the brush tool on abnormal inputs. We use these two tools to add foreground edits in non-tumor regions, i.e., to tell the network that here is a tumor. As shown in Fig. 6(b), (c), the range of the edits propagated is very small, indicating that the network believes it is probably not a tumor. In turn, we use these two tools to add background edits in the tumor region. As shown in Fig. 6(e), (f), the range of edit propagation is also very small, which indicates that the network thinks it is very likely to be a tumor here.



**Fig. 4.** Comparison of Dice similarity coefficient (DSC) of different experiments. (a) Isensee’s original network. (b) Isensee + selection tool. (c) Isensee + brush tool. (d) Isensee + adjustment tool. (e) 3D-UNet (f) 3D-UNet + selection tool.



**Fig. 5.** Use different parameters for the adjustment tool.



**Fig. 6.** Robustness testing. (a–c) Try to mark a non-tumor region. (d–f) Try to erase a tumor. The effect of these abnormal operations is insignificant.

## 4 Conclusion

This paper proposes an interactive editing method combined with 3D end-to-end segmentation network. We designed three editing tools based on the actual application scenario and different editing purposes. Automatically simulated edits are used to train the segmentation editing network. In this process, we have considered the problem of over-fitting that may be caused by feeding user edits to the network. Our approach is also applicable to other end-to-end segmentation networks. Experiments show that the three editing tools can refine the segmentation as expected, which can significantly improve the accuracy with a small number of interactions only.

**Acknowledgements.** This research is partially supported by the National Key R&D Program of China (Grant No. 2017YFB1304301) and National Natural Science Foundation of China (Grant Nos. 61572274, 61972221).

## References

1. Boykov, Y.Y., Jolly, M.P.: Interactive graph cuts for optimal boundary & region segmentation of objects in ND images. In: ICCV, vol. 1, pp. 105–112. IEEE (2001)
2. Çiçek, Ö., Abdulkadir, A., Lienkamp, S.S., Brox, T., Ronneberger, O.: 3D U-Net: learning dense volumetric segmentation from sparse annotation. In: Ourselin, S., Joskowicz, L., Sabuncu, M.R., Unal, G., Wells, W. (eds.) MICCAI 2016. LNCS, vol. 9901, pp. 424–432. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-46723-8\\_49](https://doi.org/10.1007/978-3-319-46723-8_49)
3. Isensee, F., Kickingeder, P., Wick, W., Bendszus, M., Maier-Hein, K.H.: Brain Tumor segmentation and radiomics survival prediction: contribution to the BRATS 2017 challenge. In: Crimi, A., Bakas, S., Kuijf, H., Menze, B., Reyes, M. (eds.) BrainLes 2017. LNCS, vol. 10670, pp. 287–297. Springer, Cham (2018). [https://doi.org/10.1007/978-3-319-75238-9\\_25](https://doi.org/10.1007/978-3-319-75238-9_25)
4. Kamnitsas, K., Ledig, C., et al.: Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. *MIA* **36**, 61–78 (2017)
5. Liew, J., Wei, Y., Xiong, W., Ong, S.H., Feng, J.: Regional interactive image segmentation networks. In: ICCV, pp. 2746–2754. IEEE (2017)
6. Menze, B.H., Jakab, A., Bauer, S., et al.: The multimodal brain tumor image segmentation benchmark (BRATS). *TMI* **34**(10), 1993–2024 (2015)
7. Peng, Y., Chen, L., Ou-Yang, F.X., Chen, W., Yong, J.H.: Jf-cut: A parallel graph cut approach for large-scale image and video. *TIP* **24**(2), 655–666 (2014)



8. Wang, G., Li, W., Zuluaga, M.A., Pratt, R., et al.: Interactive medical image segmentation using deep learning with image-specific fine tuning. *TMI* **37**(7), 1562–1573 (2018)
9. Wang, G., Zuluaga, M.A., Li, W., Pratt, R., et al.: DeepIGeoS: a deep interactive geodesic framework for medical image segmentation. *TPAMI* **41**(7), 1559–1572 (2018)
10. Xu, N., Price, B., Cohen, S., Yang, J., Huang, T.S.: Deep interactive object selection. In: *CVPR*, pp. 373–381 (2016)
11. Yang, L., Zhang, Y., Chen, J., Zhang, S., Chen, D.Z.: Suggestive annotation: a deep active learning framework for biomedical image segmentation. In: Descoteaux, M., Maier-Hein, L., Franz, A., Jannin, P., Collins, D.L., Duchesne, S. (eds.) *MICCAI 2017*. LNCS, vol. 10435, pp. 399–407. Springer, Cham (2017). [https://doi.org/10.1007/978-3-319-66179-7\\_46](https://doi.org/10.1007/978-3-319-66179-7_46)