

DeepEdit: Deep Editable Learning for Interactive Segmentation of 3D Medical Images

Andres Diaz-Pinto^{1,6}, Pritesh Mehta¹, Sachidanand Alle⁶, Muhammad Asad¹, Richard Brown¹, Vishwesh Nath⁶, Alvin Ihsani⁶, Michela Antonelli¹, Daniel Palkovics², Csaba Pinter³, Ron Alkalay⁴, Steve Pieper⁵, Holger R. Roth⁶, Daguang Xu⁶, Prerna Dogra⁶, Tom Vercauteren¹, Andrew Feng⁶, Abood Quraini⁶, Sebastien Ourselin¹, and M. Jorge Cardoso¹

¹ School of Biomedical Engineering & Imaging Sciences, King’s College London. London, UK. {andres.diaz-pinto,m.jorge.cardoso}@kcl.ac.uk

² Department of Periodontology, Semmelweis University, Hungary

³ EBATINCA, S.L. Canary Islands, Spain.

⁴ Beth Israel Deaconess Medical Center, MA, USA.

⁵ Isomics, Inc., MA, USA.

⁶ NVIDIA Santa Clara, CA, USA.

Abstract. Automatic segmentation of medical images is a key step for diagnostic and interventional tasks. However, achieving this requires large amounts of annotated volumes, which can be tedious and time-consuming task for expert annotators. In this paper, we introduce DeepEdit, a deep learning-based method for volumetric medical image annotation, that allows automatic and semi-automatic segmentation, and click-based refinement. DeepEdit combines the power of two methods: a non-interactive (i.e. automatic segmentation using nnU-Net, UNET or UNETR) and an interactive segmentation method (i.e. DeepGrow), into a single deep learning model. It allows easy integration of uncertainty-based ranking strategies (i.e. aleatoric and epistemic uncertainty computation) and active learning. We propose and implement a method for training DeepEdit by using standard training combined with user interaction simulation. Once trained, DeepEdit allows clinicians to quickly segment their datasets by using the algorithm in auto segmentation mode or by providing clicks via a user interface (i.e. 3D Slicer, OHIF). We show the value of DeepEdit through evaluation on the PROSTATEx dataset for prostate/prostatic lesions and the Multi-Atlas Labeling Beyond the Cranial Vault (BTCV) dataset for abdominal CT segmentation, using state-of-the-art network architectures as baseline for comparison. DeepEdit could reduce the time and effort annotating 3D medical images compared to DeepGrow alone. Source code is available at <https://github.com/Project-MONAI/MONAILabel>

Keywords: Interactive Segmentation · Deep Learning · CNNs

1 Introduction

Inspired by the landmark contributions of 2D U-Net[1], 3D U-Net[2], and V-Net[3], Convolutional Neural Networks (CNN) have become high-performing methods for automatic segmentation of medical images [4,5,6]. Medical image segmentation challenges, such as the Medical Segmentation Decathlon (*MSD*) [7], has helped steer methodological innovations and performance improvements for CNN-based methods. At the time of writing, one of the first positions on the live leaderboard ⁷ for *MSD* is held by the nnU-Net [4], a segmentation pipeline based on U-Net that automatically configures to any new medical image segmentation task. More recently, transformer-based [8] networks introduced by Hatamizadeh et. al. (Swin UNETR [9] and UNETR [6]), have further improved on nnUNET's performance, achieving state-of-the-art performance on the *MSD* segmentation tasks.

Despite their outstanding performance, automatic segmentation algorithms have not yet reached the desired level of performance needed for certain clinical applications [10]. In particular, automatic segmentation accuracy can be impacted by patient variation, acquisition differences, image artifacts [11] and limited amount of training data. In an attempt to address these challenges, interactive segmentation methods that accept user guidance to improve segmentation have been proposed [12,13,14,15]. Normalized cuts [12], random walks [13], graph-cuts [14], and geodesics [15] have been proposed for interactive segmentation using bounding-box or scribbles-based user interactions. However, a major limitation of these classical methods is that they only succeed in addressing simpler segmentation problems where objects have clear structural boundaries, and require extensive user interaction for more complex segmentation cases containing ambiguity in object boundaries [10].

A number of deep learning-based interactive segmentation methods based have been proposed for improving the robustness of interactive image segmentation [16,17]. In [16], user foreground and background clicks were converted into euclidean distance maps, and subsequently learned from as additional input channels to a CNN. Inspired by the aforementioned studies and other incremental works, interactive methods for medical image segmentation based on deep learning have been recently proposed [18,10,19]. In [18], a bounding-box and scribble-based CNN segmentation pipeline was proposed, whereby a user-provided bounding box is first used to assist the CNN in foreground segmentation. This was followed by image-specific fine-tuning using user-provided scribbles. Due to the inclusion of user interaction within a CNN, this method provided greater robustness and accuracy than state-of-the-art for segmenting previously unseen objects, while also using fewer user interactions than existing interactive segmentation methods. In contrast, Sakinis et al. [10] proposed a click-based method, motivated in part by the work of [16]. In their work, Gaussian-smoothed foreground and background clicks were added as input chan-

⁷ <https://decathlon-10.grand-challenge.org/evaluation/challenge/leaderboard/>

nels to an encoder-decoder CNN. Experiments on multiple-organ segmentation in CT volumes showed that their method delivers 2D segmentations in a fast and reliable manner, generalizes well to unseen structures, and accurately segments organs with few clicks. An alternate method that first performs an automatic CNN segmentation step, followed by an optional refinement through user clicks or scribbles, was proposed by [19]. Their method, named DeepIGeoS, achieved substantially improved performance compared to automatic CNN on 2D placenta and 3D brain tumour segmentation, and higher accuracy with fewer interactions than traditional interactive segmentation methods.

Automatic and semi-automatic segmentation methods are available as part of open-source software packages for medical imaging analysis: ITK-SNAP [20] which offers semi-automatic active contour segmentation [21]; 3D Slicer [22] and MITK [23] offer automatic, boundary-points-based [24]; DeepGrow [10] segmentation through the NVIDIA Clara AI-Assisted Annotation Extension; as well as other classic semi-automatic segmentation methods such as region growing [25] and level sets [26].

We propose DeepEdit, a method that combines an automatic and a semi-automatic approach for 3D medical images into a single deep learning-based model. DeepEdit has three working modes: first, it can be used in click-free inference mode (similar to a regular segmentation network), providing fully-automatic segmentation predictions which can be used as a form of initialisation; second, it allows users to provide clicks to initialise and guide a semi-automatic segmentation model; lastly, given an initial segmentation, DeepEdit can be used to refine and improve the initial prediction by providing editing clicks. DeepEdit training process is similar to the algorithm proposed by Sakinis et al. [10] (DeepGrow) - Gaussian-smoothed clicks for all labels and background are generated and added as input to the backbone CNN, but removes the minimum-click limitation of DeepGrow. Contrary to DeepGrow, our proposed DeepEdit model allows the prediction of an automatic segmentation-based initialisation without user-provided clicks, which can then be further edited by providing clicks. Lastly, the proposed model can also be used for multi-label segmentation problems, allowing the user to generate/segment all labels simultaneously instead of one label at a time.

The flexibility offered by embedding these three functionalities (auto segmentation, semi-automatic segmentation and label refinement) allows DeepEdit to be integrated into an active learning pipeline. For instance, it could be used in automatic segmentation mode for aleatoric and/or epistemic uncertainty computation to rank unlabeled images (See Fig. 1(b)).

In order to show the performance of DeepEdit, we present applications for single and multiple label segmentation for annotating the datasets: prostate, prostatic lesion, and abdominal organ segmentation.

2 Proposed Method

The DeepEdit architecture is based on a backbone that can be any segmentation network (i.e. UNET, nnU-Net [4], UNETR, SwinUNETR[9]). The main difference resides in how this backbone is trained and the number of channels in the input tensor. For training, the input tensor could be either the image with zeroed tensors (automatic segmentation mode) or the image with tensors representing label and background clicks provided by the user (interactive mode). In Fig. 1, DeepEdit is presented in its training and inference mode.

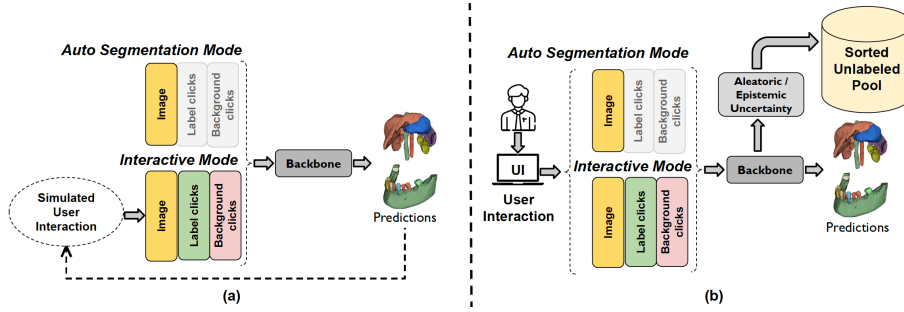


Fig. 1. General schema of the DeepEdit: (a) Training and (b) Inference Mode. DeepEdit training process consists of two modes: the automatic segmentation mode and the interactive mode. Simulated clicks for all labels plus background are added to a backbone network as input channels. Input tensor could be either the image with zero-tensors (automatic segmentation mode) or the image with tensors representing label clicks and background clicks provided by the user (interactive mode).

As shown in Fig. 1, DeepEdit can integrate an active learning strategy in which the trained model is used to rank the unlabelled volumes from the most uncertain to the least uncertain. Every time the expert annotator fetches an image, DeepEdit presents the one with more uncertainty, allowing the model to learn from the most challenging cases first.

2.1 User interaction and simulated clicks

Our proposed method embeds three approaches: automatic segmentation, semi-automatic segmentation and interactive segmentation. This means, for some iterations, DeepEdit is trained click-free, and for others is trained as DeepGrow (clicks are simulated and included in the input tensor as extra channels). As DeepGrow relies on clicks provided by a user or agent, we simulated those following the similar approach presented in Sakinis’ work - voxels where clicks are located are set to one and smoothed with a Gaussian filter. This is done for the positive label, if single label task or, in general, for all labels and background.

2.2 Training DeepEdit

As previously mentioned, the training process of the DeepEdit algorithm involves click-free iterations and iterations with simulated clicks. As shown in Fig. 1(a), the input of the network is a concatenation of multiple tensors: the image, a tensor containing clicks simulated for each label and a tensor containing clicks simulated for the background. Our proposed algorithm mixes two types of training loops: a) click-free training iterations - meaning that for some iterations, the tensors representing the labels and background clicks are zeros (training for the automatic segmentation); b) simulated-click based training iterations - where labels and background clicks are simulated and placed in the tensors. We sample from a uniform distribution with probability p to determine which iteration is click-free and which one uses simulated clicks. An additional hyper-parameter, the number of simulated clicks per iteration, is set by the user as a function of the task complexity. These two training loops allow DeepEdit to be used fully automatically, semi-automatic, and as a segmentation refinement approach. We developed all these new transforms for click simulation and mixed training in MONAI [10,27].

3 Experimental Results

In order to demonstrate the flexibility and value of DeepEdit, and the impact of the number of simulated clicks, a set of experiments were performed on the PROSTATEx and Multi Atlas Labeling Beyond The Cranial Vault (BTCV) datasets. We present the impact of the number of clicks in the prostate, prostatic lesion, and abdominal organs (BTCV dataset) segmentation. For both single and multilabel segmentation experiments, we used a learning rate of $1e-4$, batch size equal to 1 and Adam optimizer. The following MONAI transforms were used to train and validate DeepEdit: intensity normalization, random flipping (vertically and horizontally), random shift intensity and random rotation.

All our experiments have been implemented using the MONAI Core library [27] (version 0.8.1) and MONAI Label platform (version 0.3.1). All source code for DeepEdit algorithm and Active Learning strategies have been made publicly available and documented at <https://github.com/Project-MONAI/MONAILabel> as part of the MONAI Label repository.

3.1 Prostate Segmentation Tasks

DeepEdit applications were built for whole prostate segmentation and prostatic lesion segmentation. Experiments were run using the PROSTATEx Challenge training dataset, [28], hereby referred to as the PROSTATEx dataset. For both single- and multi-label tasks, experiments were conducted to compare the segmentation performance of DeepEdit as the hyperparameter controlling the number of training iterations with zero simulated clicks is varied. We compared DeepEdit on different click-free training iterations: DeepEdit-0 (equivalent to

DeepGrow), DeepEdit-0.25 and DeepEdit-0.50, meaning that 0, 25 and 50 percent of the training iterations were click-free. Ten-fold cross-validation was performed for both tasks. Segmentation quality was assessed using the Dice coefficient. As in [10], segmentation performance at inference time was assessed using simulated clicks instead of user mouse-clicks to objectively assess how segmentation quality improves as clicks are added; segmentation performance was assessed at 0, 1, 5, and 10 simulated inference clicks. The presented results are an average of three repetitions to account for variability in simulated inference click placement.

Whole Prostate Segmentation The whole prostate segmentation task concerns the segmentation of the prostate on T2-weighted MRI (T2WI). Eleven patients from the PROSTATEx dataset were excluded due to inconsistencies between T2WI and the ground-truth segmentations, leaving a total of 193 patients for use in experiments.

T2WI were pre-processed by resampling to a common resolution of $0.5\text{mm} \times 0.5\text{mm} \times 3.0\text{mm}$, normalization using per-image whitening, and cropping/padding to a common size of $320 \times 320 \times 32$.

A comparison of DeepEdit-0 (equivalent to DeepGrow), DeepEdit-0.25, and DeepEdit-0.5 is shown in Table 1. Furthermore, the distributions of Dice scores are shown in Fig. 2. DeepEdit-0.5 was found to have the highest click-free mean Dice score (0.908), while DeepEdit-0 gave the highest mean Dice scores at 1 to 10 simulated inference clicks.

Scheme	Model	Number of simulated clicks			
		0	1	5	10
Fully automatic	nnU-Net [4]	0.910±0.033	-	-	-
Automatic	DeepEdit-0.25	0.908±0.054	0.912±0.049	0.921±0.041	0.929±0.026
+ interactive editing	DeepEdit-0.5	0.908±0.046	0.911±0.044	0.919±0.035	0.926±0.028
Fully interactive	DeepEdit-0 (DeepGrow)	0.907±0.041	0.915±0.035	0.926±0.024	0.932±0.020

Table 1. Whole prostate segmentation mean Dice scores $\pm$ one standard deviation for the 193 PROSTATEx dataset patients used in the ten-fold cross-validation, for nnU-Net, DeepEdit-0 (equivalent to DeepGrow), DeepEdit-0.25, and DeepEdit-0.5. The highest mean Dice in each column is shown in bold.

Prostatic Lesion Segmentation The prostatic lesion segmentation task concerns the segmentation of lesions within the prostate using T2WI, apparent diffusion coefficient (ADC) map, and computed high b-value diffusion-weighted MRI (DWI). Since our experiments were conducted using the PROSTATEx dataset, we used the PROSTATEx definition of a lesion, i.e., a prostatic lesion is defined as any area of suspicion attributed to a Prostate Imaging-Reporting and Data System (PI-RADS) score by the expert radiologist (anonymous clinician) who read and reported PROSTATEx dataset cases; all lesions in the PROSTATEx dataset were scored PI-RADS ≥ 2 . Four patients from the PROSTATEx dataset were excluded due to not containing contoured lesions in the ground truth (the

assessment metrics would have been undefined), leaving a total of 200 patients with a total of 299 lesions for use in experiments.

A b-value, $b = 2000$, was selected for computing high b-value DWI; computed b2000 (Cb2000) DWI were generated using DWI acquired at lower b-values, extrapolated by assuming a monoexponential model for the per-voxel observed signal. ADC map and Cb2000 DWI were registered to T2WI to account for voluntary/involuntary patient movement between acquisitions and differences in resolution. T2WI and Cb2000 DWI were normalised by dividing voxel intensities by the interquartile mean of central gland (CG) voxel intensities [29]; ADC maps were not normalised as they contain a quantitative measurement. T2WI, ADC map, and Cb2000 DWI were resampled to a common resolution of $0.5 \text{ mm} \times 0.5 \text{ mm} \times 3 \text{ mm}$. Then, whole prostate masks were used to crop the prostate region on all MR modalities; a margin was applied in each direction to reduce the likelihood of prostate tissue being discarded. Next, a cropping/padding transformation was used to ensure a common spatial size of $256 \times 256 \times 32$.

A comparison of DeepEdit-0 (equivalent to DeepGrow), DeepEdit-0.25, and DeepEdit-0.5 is shown in Table 2. Furthermore, the distributions of Dice scores are shown in Fig 3. As in the whole prostate segmentation task, DeepEdit-0.5 gave the highest click-free mean Dice score (0.272), while DeepEdit-0 (equivalent to DeepGrow) gave the highest mean Dice scores at 1 to 10 simulated inference clicks.

3.2 Abdominal Organ Segmentation

A second set of experiments using the UNETR [6] as backbone were performed on the BTCV dataset. For this, we used 23 images for training, 6 for validation, and an image size of $128 \times 128 \times 128$. As the previous analysis, we compared DeepEdit

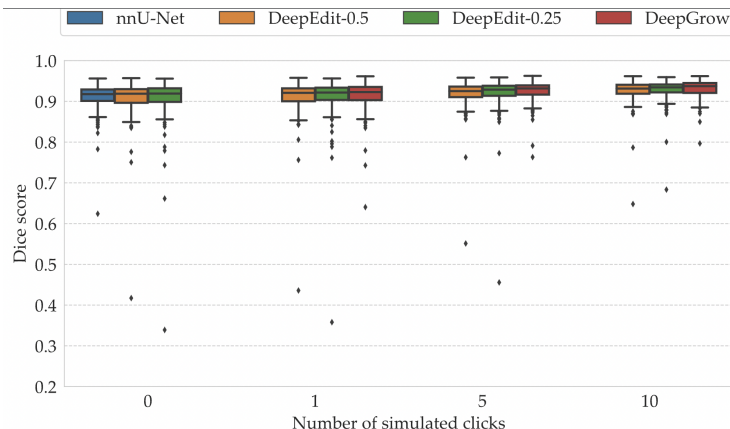


Fig. 2. Whole prostate segmentation: Dice score box plots for the 193 PROSTATEx dataset patients used in the ten-fold cross-validation, for DeepEdit-0 (equivalent to DeepGrow), DeepEdit-0.25, and DeepEdit-0.5.

Scheme	Model	Number of simulated clicks			
		0	1	5	10
Fully automatic	nnU-Net [4]	0.332±0.254	-	-	-
Automatic	DeepEdit-0.25	0.268±0.271	0.498±0.174	0.632±0.130	0.697±0.114
+ interactive editing	DeepEdit-0.5	0.272±0.266	0.453±0.197	0.592±0.163	0.663±0.145
Fully interactive	DeepEdit-0 (DeepGrow)	0.166±0.254	0.527±0.166	0.670±0.111	0.723±0.095

Table 2. Prostatic lesion segmentation mean Dice scores $\pm$ one standard deviation for the 200 PROSTATEx dataset patients used in the ten-fold cross-validation, for nnU-Net, DeepEdit-0 (equivalent to DeepGrow), DeepEdit-0.25, and DeepEdit-0.5. The highest mean Dice in each column is shown in bold.

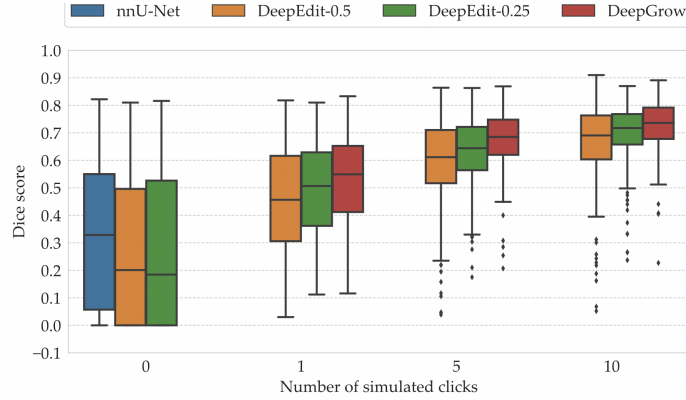


Fig. 3. Prostatic lesion segmentation: Dice score box plots for the 200 PROSTATEx dataset patients used in the ten-fold cross-validation, for DeepEdit-0 (equivalent to DeepGrow), DeepEdit-0.25, and DeepEdit-0.5.

trained with 0% click-free training iterations (equivalent to DeepGrow), 25% click-free training iterations, and 50% click-free training iterations.

Scheme	Model	Single label				Multilabel			
		Number of simulated clicks				Number of simulated clicks			
		0	1	5	10	0	1	5	10
Fully automatic	UNETR	0.919	-	-	-	0.911	-	-	-
Automatic	DeepEdit-0.25	0.874	0.902	0.895	0.895	0.901	0.887	0.900	0.906
+ interactive editing	DeepEdit-0.5	0.835	0.850	0.864	0.876	0.875	0.895	0.899	0.905
Fully interactive	DeepEdit-0 (DeepGrow)	0.897	0.899	0.913	0.931	0.892	0.892	0.914	0.926

Table 3. Dice scores for single and multilabel segmentation on the validation set using the BTCV dataset. For single label we used the spleen organ and multilabel we used spleen, liver, and left and right kidneys. We show the results obtained for 0, 1, 5, and 10 clicks simulated during validation. Highest Dice scores in each column are shown in bold.

In Table 3, we show the obtained results on the validation set for 0, 1, 5, and 10 simulated clicks. As a fair comparison, we trained and validated a UNETR and the DeepEdit using the same images, same transforms and for the same number of epochs (200).

As it is shown in Table 3, any DeepEdit configuration performs slightly better than the UNETR on the validation set when simulated clicks are provided.

Additional qualitative results obtained from DeepEdit are presented in the supplementary material. We show how DeepEdit could also be applied on two additional clinical problems: segmentation of metastatic spines and teeth segmentation for treatment planning in reconstructive periodontal surgery.

4 Conclusion

In this study, we introduce DeepEdit, a method that enables an uncertainty-driven active learning workflow for labelling medical images using a framework that combines deep learning-based automatic segmentation and interactive edits. Compared to previous interactive approaches, DeepEdit can be easily integrated into any 3D medical segmentation pipeline that includes active learning strategies. Using DeepEdit, biologists/clinicians can 1) obtain an automatic segmentation that can later be modified or refined by providing clicks through a user interface (e.g., 3D Slicer, OHIF), or 2) provide clicks to get a segmentation (semi-automatic segmentation). This could significantly reduce the time clinicians/biologists spend on annotating more datasets, which translates in less cost and effort spent on this process.

References

1. O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241, Springer, 2015.
2. O. Çiçek, A. Abdulkadir, S. Lienkamp, T. Brox, and O. Ronneberger, “3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation,” in *Medical Image Computing and Computer-Assisted Intervention*, pp. 424–432, 2016.
3. F. Milletari, N. Navab, and S.-a. Ahmadi, “V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation,” in *3DV*, 2016.
4. F. Isensee, P. F. Jaeger, S. A. Kohl, J. Petersen, and K. H. Maier-Hein, “nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation,” *Nature Methods*, vol. 18, pp. 203–211, 2020.
5. Y. He, D. Yang, H. Roth, C. Zhao, and D. Xu, “DiNTS: Differentiable Neural Network Topology Search for 3D Medical Image Segmentation,” in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5837–5846, 2021.
6. A. Hatamizadeh, Y. Tang, V. Nath, D. Yang, A. Myronenko, B. Landman, H. R. Roth, and D. Xu, “UNETR: Transformers for 3D Medical Image Segmentation,” in *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1748–1758, 2022.
7. M. Antonelli, A. Reinke, S. Bakas, K. Farahani, A. Kopp-Schneider, B. A. Landman, G. Litjens, B. Menze, O. Ronneberger, R. M. Summers, *et al.*, “The medical segmentation decathlon,” *Nature Communications*, vol. 13, no. 1, pp. 1–13, 2022.

8. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," in *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems*, vol. 2017, pp. 6000–6010, 2017.
9. A. Hatamizadeh, V. Nath, Y. Tang, D. Yang, H. R. Roth, and D. Xu, "Swin UNETR: Swin Transformers for Semantic Segmentation of Brain Tumors in MRI Images," in *International MICCAI Brainlesion Workshop*, pp. 272–284, Springer, 2022.
10. T. Sakinis, F. Milletari, H. Roth, P. Korfiatis, P. M. Kostandy, K. Philbrick, Z. Akkus, Z. Xu, D. Xu, and B. J. Erickson, "Interactive segmentation of medical images through fully convolutional neural networks," *arXiv preprint arXiv:1903.08205*, vol. abs/1903.08205, 2019.
11. F. Zhao and X. Xie, "An Overview of Interactive Medical Image Segmentation," *Annals of the British Machine Vision Association*, vol. 2013, no. 7, pp. 1–22, 2013.
12. J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.
13. L. Grady, T. Schiweitz, S. Aharon, and R. Westermann, "Random walks for interactive organ segmentation in two and three dimensions: Implementation and validation," in *Medical Image Computing and Computer-Assisted Intervention*, pp. 773–780, 2005.
14. Y. Boykov and G. Funka-Lea, "Graph cuts and efficient N-D image segmentation," *International Journal of Computer Vision*, vol. 70, no. 2, pp. 109–131, 2006.
15. Z. Akkus, J. Sedlar, L. Coufalova, P. Korfiatis, T. L. Kline, J. D. Warner, J. Agrawal, and B. J. Erickson, "Semi-automated segmentation of pre-operative low grade gliomas in magnetic resonance imaging," *Cancer Imaging*, vol. 15, no. 12, pp. 1–10, 2015.
16. N. Xu, B. Price, S. Cohen, J. Yang, and T. Huang, "Deep Interactive Object Selection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 373–381, 2016.
17. E. Agustsson, J. R. Uijlings, and V. Ferrari, "Interactive Full Image Segmentation by Considering All Regions Jointly," in *2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 11614–11623, 2019.
18. G. Wang, W. Li, M. A. Zuluaga, R. Pratt, P. A. Patel, M. Aertsen, T. Doel, A. L. David, J. Deprest, S. Ourselin, and T. Vercauteren, "Interactive Medical Image Segmentation Using Deep Learning with Image-Specific Fine Tuning," *IEEE Transactions on Medical Imaging*, vol. 37, no. 7, pp. 1562–1573, 2018.
19. G. Wang, M. A. Zuluaga, W. Li, R. Pratt, P. A. Patel, M. Aertsen, T. Doel, A. L. David, J. Deprest, S. Ourselin, and T. Vercauteren, "DeepIGeoS: A Deep Interactive Geodesic Framework for Medical Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 7, pp. 1559–1572, 2019.
20. P. A. Yushkevich, J. Piven, H. Cody Hazlett, R. Gimpel Smith, S. Ho, J. C. Gee, and G. Gerig, "User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability," *Neuroimage*, vol. 31, no. 3, pp. 1116–1128, 2006.
21. M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active Contour Models," *International Journal of Computer Vision*, pp. 321–331, 1988.
22. A. Fedorov, R. Beichel, J. Kalpathy-Cramer, J. Finet, J.-C. Fillion-Robin, S. Pujol, C. Bauer, D. Jennings, F. Fennessy, M. Sonka, J. Buatti, S. Aylward, J. V. Miller, S. Pieper, and R. Kikinis, "3D Slicer as an Image Computing Platform for the

- Quantitative Imaging Network,” *Magnetic resonance imaging*, vol. 30, pp. 1323–1341, Nov. 2012.
23. M. Nolden, S. Zelzer, A. Seitel, D. Wald, M. Müller, A. M. Franz, D. Maleike, M. Fangerau, M. Baumhauer, L. Maier-Hein, K. H. Maier-Hein, H. P. Meinzer, and I. Wolf, “The medical imaging interaction toolkit: Challenges and advances: 10 years of open-source development,” *International Journal of Computer Assisted Radiology and Surgery*, vol. 8, no. 4, pp. 607–620, 2013.
 24. K. K. Maninis, S. Caelles, J. Pont-Tuset, and L. Van Gool, “Deep Extreme Cut: From Extreme Points to Object Segmentation,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 616–625, 2018.
 25. R. Adams and L. Bischof, “Seeded Region Growing,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 6, pp. 641–647, 1994.
 26. S. Osher and J. A. Sethian, “Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations,” *Journal of Computational Physics*, vol. 79, no. 1, pp. 12–49, 1988.
 27. MONAI Consortium, “MONAI: Medical Open Network for AI,” 3 2020.
 28. G. Litjens, O. Debats, J. Barentsz, N. Karssemeijer, and H. Huisman, “ProstateX Challenge data,” 2017.
 29. P. Mehta, M. Antonelli, S. Singh, N. Grondecka, E. W. Johnston, H. U. Ahmed, M. Emberton, S. Punwani, and S. Ourselin, “AutoProstate: Towards Automated Reporting of Prostate MRI for Prostate Cancer Assessment Using Deep Learning,” *Cancers*, vol. 13, no. 23, 2021.