Quality-Aware Memory Network for Interactive Volumetric Image Segmentation

Quality-Aware Memory Network for Interactive Volumetric Image Segmentation

Tianfei Zhou[1], Liulei Li[2], Gustav Bredell[1], Jianwu Li[2(✉)], and Ender Konukoglu[1]

[1] Computer Vision Laboratory, ETH Zurich, Zurich, Switzerland
{tianfei.zhou,gustav.bredell,ender.konukoglu}@vision.ee.ethz.ch
[2] School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China
{liliulei,ljw}@bit.edu.cn
https://github.com/0liliulei/Mem3D

Abstract. Despite recent progress of automatic medical image segmentation techniques, fully automatic results usually fail to meet the clinical use and typically require further refinement. In this work, we propose a *quality-aware memory network* for interactive segmentation of 3D medical images. Provided by user guidance on an arbitrary slice, an interaction network is firstly employed to obtain an initial 2D segmentation. The quality-aware memory network subsequently propagates the initial segmentation estimation bidirectionally over the entire volume. Subsequent refinement based on additional user guidance on other slices can be incorporated in the same manner. To further facilitate interactive segmentation, a quality assessment module is introduced to suggest the next slice to segment based on the current segmentation quality of each slice. The proposed network has two appealing characteristics: 1) The memory-augmented network offers the ability to quickly encode past segmentation information, which will be retrieved for the segmentation of other slices; 2) The quality assessment module enables the model to directly estimate the qualities of segmentation predictions, which allows an active learning paradigm where users preferentially label the lowest-quality slice for multi-round refinement. The proposed network leads to a robust interactive segmentation engine, which can generalize well to various types of user annotations (*e.g.,* scribbles, boxes). Experimental results on various medical datasets demonstrate the superiority of our approach in comparison with existing techniques.

Keywords: Interactive segmentation · Memory-augmented network

T. Zhou and L. Li—Contribute equally to this work.

M. de Bruijne et al. (Eds.): MICCAI 2021, LNCS 12902, pp. 560–570, 2021.
https://doi.org/10.1007/978-3-030-87196-3_52

1 Introduction

Accurate segmentation of organs/lesions from medical imaging data holds the promise of significant improvement of clinical treatment, by allowing the extraction of accurate models for visualization, quantification or simulation. Although recent deep learning based automatic segmentation engines [16,21,29,34] have achieved impressive performance, they still struggle to achieve sufficiently accurate and robust results for clinical practice, especially in the presence of poor image quality (*e.g.,* noise, low contrast) or highly variable shapes (*e.g.,* anatomical structures). Consequently, *interactive segmentation* [2,18,27,28,32,33] garners research interests of the medical image analysis community, and recently became the choice in many real-life medical applications.

In interactive segmentation, the user is factored in to play a crucial role in guiding the segmentation process and in correcting errors as they occur (often in an iteratively-refined manner). Classical approaches employ Graph Cuts [1], GeoS [4] or Random Walker [5,6] to incorporate scribbles for segmentation. Yet, these methods require a large amount of input from users to segment targets with low contrast and ambiguous boundaries. With the advent of deep learning, there has been a dramatically increasing interest in learning from user interactions. Recent methods demonstrate higher segmentation accuracy with fewer user interactions than classical approaches. Despite this, many approaches [11,22,26] only focus on 2D medical images, which are infeasible to process ubiquitous 3D data. Moreover, 2D segmentation does not allow the integration of prior knowledge regarding the 3D structure, and slice-by-slice interactive segmentation will impose extremely high annotation cost to users. To address this, many works [3,13,20,27,28] carefully design 3D networks to segment voxels at a time. While these methods enjoy superior ability of learning high-order, volumetric features, they require significantly more parameters and computations in comparison with the 2D counterparts. This necessitates compromises in the 3D network design to fit into a given memory or computation budget.

To address these issues, we take a novel perspective to explore memory-augmented neural networks [12,14,23,25] for 3D medical image segmentation. Memory networks augment neural networks with an external memory component, which allows the network to explicitly access the past experiences. They have been shown effective in few-shot learning [23], contrastive learning [7,29], and also been explored to solve reasoning problems in visual dialog [12,25]. The basic idea is to retrieve the relevant information from the external memory to answer a question at hand by using trainable memory modules. We take inspiration from these efforts to cast volumetric segmentation as a memory-based reasoning problem. Fundamental to our model is an external memory, which enables the model to online store segmented slices in the memory and later mine useful representations from the memory for segmenting other slices. In this way, our model makes full use of context within 3D data, and at the same time, avoids computationally expensive 3D operations. During segmentation, we dynamically update the memory to maintain shape or appearance variations of the target.

This facilitates easy model updating without extensive parameter optimization. Based on the memory network, we propose a novel interactive segmentation engine with three basic processes: 1) *Initialization:* an interaction network is employed to respond to user guidance on an arbitrary slice to obtain an initial 2D segmentation of a target. 2) *Propagation:* the memory network propagates the initial mask to the entire volume. 3) *Refinement:* the physician could provide extra guidance on low-quality slices for iterative refinement if the segmentation results are unsatisfactory.

Our contributions are three-fold: **First**, we propose a memory-augmented network for volumetric interactive segmentation. It is able to incorporate rich 3D contextual information, while avoiding expensive 3D operations. **Second**, we equip the memory network with a quality assessment module to assess the quality of each segmentation. This facilitates automatic selection of appropriate slices for iterative correction via human-in-the-loop. **Third**, our approach outperforms previous methods by a significant margin on two public datasets, while being able to handle various forms of interactions (*e.g.,* scribbles, bounding boxes).

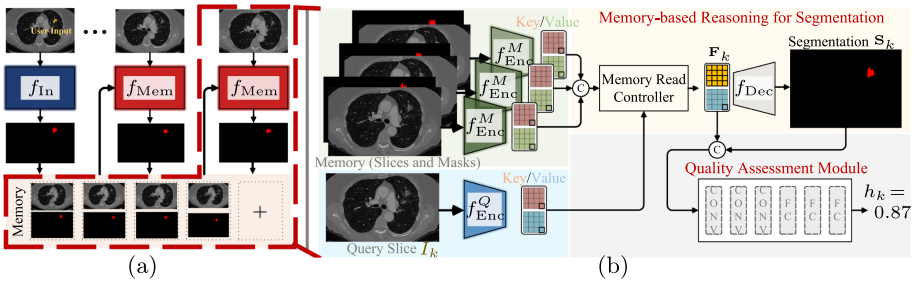

Fig. 1. Illustration of the proposed 3D interactive segmentation engine. (a) Simplified schematization of our engine that solves the task with an interaction network (f_{In}) and a quality-aware memory network (f_{Mem}). (b) Detailed network architecture of f_{Mem}. ⓒ denotes the concatenation operation. Zoom in for details.

2 Methodology

Let $V \in \mathbb{R}^{h \times w \times c}$ be a volumetric image to be segmented, which has a spatial size of $h \times w$ and c slices. Our approach aims to obtain a 3D binary mask $\mathbf{S} \in \{0,1\}^{h \times w \times c}$ for a specified target by utilizing user guidance. As shown in Fig. 1(a), the physician is asked to provide an initial input on an arbitrary slice $I_i \in \mathbb{R}^{h \times w}$, where I_i denotes the i-th slice of V. Then, an interaction network (f_{In}, Sect. 2.1) is employed to obtain a coarse 2D segmentation $\mathbf{S}_i \in [0,1]^{h \times w}$ for I_i. Subsequently, $\mathbf{S}_i$ is propagated to all other slices with a quality-aware memory network (f_{Mem}, Sect. 2.2) to obtain $\mathbf{S}$. Our approach also takes into account iterative refinement so that segmentation performance can be progressively improved with multi-round inference. To aid the refinement, the memory

network has a module that estimates the segmentation performance on each slice and suggests the user to place guidance on the slice with the worst segmentation quality.

2.1 Interaction Network

The interaction network takes the user annotation at an interactive slice I_i to segment the specified target (or refine the previous result). At the t^{th} round, its input consists of three images: the original gray-scale image I_i, the segmentation mask from the previous round $\mathbf{S}_i^{t-1}$, and a binary image $\mathbf{M}_i \in \{0,1\}^{h \times w}$ that encodes user guidance. Note that in the first round (*i.e.,* $t = 0$), the segmentation mask $\mathbf{S}_i^{-1}$ is initialized as a neutral mask with 0.5 for all pixels. These inputs are concatenated along the channel dimension to form an input tensor $\mathbf{X}_i^t \in \mathbb{R}^{h \times w \times 3}$. The interaction network f_{IN} conducts the segmentation for I_i as follows:

$$\mathbf{S}_i^t = f_{\text{In}}(\mathbf{X}_i^t) \in \mathbb{R}^{h \times w}. \tag{1}$$

Region-of-Interest (ROI). To further enhance performance and avoid mistakes in case of small targets or low-contrast tissues, we propose to crop the image according to the rough bounding-box estimation of user input, and apply f_{IN} only to the ROI. We extend the bounding box by 10% along sides to preserve more context. Each ROI region is resized into a fixed size for network input. After segmentation, the mask made within the ROI is inversely warped and pasted back to the original location.

2.2 Quality-Aware Memory Network

Given the initial segmentation $\mathbf{S}_i^t$, our memory network learns from the interactive slice I_i and segments the specified target in other slices. It stores previously segmented slices in an external memory, and takes advantage of the stored 3D image and corresponding segmentation to improve the segmentation of each 2D query image. The network architecture is shown in Fig. 1(b). In the following paragraphs, the superscript 't' is omitted for conciseness unless necessary.

Key-Value Embedding. Given a query slice I_k, the network mines useful information from memory $\mathcal{M}$ for segmentation. Here, each memory cell $\mathcal{M}_j \in \mathcal{M}$ consists of a slice I_{m_j} and its segmentation mask $\mathbf{S}_{m_j}$, where m_j indicates the index of the slice in the original volume. As shown in Fig. 1(b), we first encode the query I_k as well as each memory cell $\mathcal{M}_j = \{I_{m_j},\ \mathbf{S}_{m_j}\}$ into pairs of *key* and *value* using dedicated encoders (*i.e.,* query f_{Enc}^Q and memory encoder f_{Enc}^M):

$$\mathbf{K}_k^Q, \mathbf{V}_k^Q = f_{\text{Enc}}^Q(I_k), \tag{2}$$

$$\mathbf{K}_{m_j}^M, \mathbf{V}_{m_j}^M = f_{\text{Enc}}^M(I_{m_j}, \mathbf{S}_{m_j}). \tag{3}$$

Here, $\mathbf{K}_k^Q \in \mathbb{R}^{H\times W\times C/8}$ and $\mathbf{V}_k^Q \in \mathbb{R}^{H\times W\times C/2}$ indicate key and value embedding of the query I_k, respectively, whereas $\mathbf{K}_{v_j}^M$ and $\mathbf{V}_{v_j}^M$ correspond to the key and value of the memory cell $\mathcal{M}_j$. H, W and C denote the height, width and channel dimension of the feature map from the backbone network, respectively. Note that for each memory cell, we apply Eq. (3) to obtain pairs of key and value embedding. Subsequently, all memory embedding are stacked together to build a pair of 4D key and value features (*i.e.*,$\mathbf{K}^M \in \mathbb{R}^{N\times H\times W\times C/8}$ and $\mathbf{V}^M \in \mathbb{R}^{N\times H\times W\times C/2}$), where $N = |\mathcal{M}|$ denotes memory size.

Memory Reading. The memory read controller retrieves relevant information from the memory based on the current query. Following the key-value retrieval mechanism in [12,25], we first compute the similarity between every 3D location $p \in \mathbb{R}^3$ in $\mathbf{K}^M$ with each spatial location $q \in \mathbb{R}^2$ in $\mathbf{K}_k^Q$ with dot product:

$$s_k(p,q) = \frac{\mathbf{K}^M(p)\cdot\mathbf{K}_k^Q(q)}{\|\mathbf{K}^M(p)\|\|\mathbf{K}_k^Q(q)\|} \in [-1,1], \tag{4}$$

where $\mathbf{K}^M(p) \in \mathbb{R}^{C/8}$ and $\mathbf{K}_k^Q(q) \in \mathbb{R}^{C/8}$ denote the features at the p^{th} and q^{th} position of $\mathbf{K}^M$ and $\mathbf{K}_k^Q$, respectively. Next, we compute the read weight w_k by softmax normalization:

$$w_k(p,q) = \exp(s_k(p,q))/\sum\nolimits_o \exp(s_k(o,q)) \in [0,1]. \tag{5}$$

Here, $w_k(p,q)$ measures the matching probability between p and q. The memory summarization is then obtained using the weight to combine the memory value:

$$\mathbf{H}_k(q) = \sum\nolimits_p w_k(p,q)\mathbf{V}^M(p) \in \mathbb{R}^{C/2}. \tag{6}$$

Here, $\mathbf{V}^M(p) \in \mathbb{R}^{C/2}$ denotes the feature of the p^{th} 3D position in $\mathbf{V}^M$. $\mathbf{H}_k(q)$ indicates the summarized representation of location q. For all $H \times W$ locations in $\mathbf{K}_k^Q$, we independently apply Eq. (6) and obtain the feature map $\mathbf{H}_k \in \mathbb{R}^{H\times W\times C/2}$. To achieve a more comprehensive representation, the feature map is concatenated with query value $\mathbf{V}_k^Q$ to compute a final representation $\mathbf{F}_k = \mathtt{cat}(\mathbf{H}_k, \mathbf{V}_k^Q) \in \mathbb{R}^{H\times W\times C}$.

Final Segmentation Readout. $\mathbf{F}_k$ is leveraged by a decoder network f_{Dec} to predict the final segmentation probability map for the query slice I_k:

$$\mathbf{S}_k = f_{\text{Dec}}(\mathbf{F}_k) \in [0,1]^{h\times w}. \tag{7}$$

Quality Assessment Module. While the memory network provides a compelling way to produce 3D segmentation, it does not support human-in-the-loop scenarios. To this end, we equip the memory network with a lightweight quality assessment head, which computes a quality score for each segmentation mask.

In particular, we consider *mean intersection-over-union (mIoU)* as the basic index for quality measurement. For each query I_k, we take the feature $\mathbf{F}_k$ and the corresponding segmentation $\mathbf{S}_k$ together to regress a mIoU score h_k:

$$h_k = f_{\mathrm{QA}}(\mathbf{F}_k, \mathbf{S}_k) \in [0, 1], \tag{8}$$

where $\mathbf{S}_k$ is firstly resized to a size of $H \times W$ and then concatenated with $\mathbf{F}_k$ for decoding. The slice with the lowest score is curated for next-round interaction.

2.3 Detailed Network Architecture

We follow [31] to implement the interaction network $f_{\mathrm{In}}(\cdot)$ as a coarse-to-fine segmentation network, however, other network architectures (*e.g.,* U-Net [21]) can also be used here instead. The network is trained using the cross-entropy loss. For the quality-aware memory network, we utilize ResNet-50 [8] as the backbone network for both f^{Q}_{Enc} (Eq. (2)) and f^{M}_{Enc} (Eq. (3)). The `res4` feature map of ResNet-50 is taken for computing the key and value embedding. For $f_{\mathrm{Dec}}(\cdot)$, we first apply Atrous Spatial Pyramid Pooling module after the memory read operation to enlarge the receptive field. We use three parallel dilated convolution layers with dilation rates 2, 4 and 8. Then, the learned feature is decoded with a residual refinement module proposed in [19]. The quality-aware module, $f_{\mathrm{QA}}(\cdot)$, consists of three 3×3 convolutional layers and three fully connected layers.

3 Experiment

Experimental Setup. Our experiments are conducted on two public datasets: ***MSD*** [24] includes ten subsets with different anatomy of interests, with a total of 2,633 3D volumes. In our experiments, we study the most challenging lung (64/32 for `train`/`val`) and colon (126/64 for `train`/`val`) subsets. ***KiTS***$_{19}$ [9] contains 300 arterial phase abdominal CT scans with annotations of kidney and tumor. We use the released 210 scans (168/42 for `train`/`val`) for experiments.

For comparison, we build a baseline model, named Interactive 3D nnU-Net, by adapting nnU-Net [10] into an interactive version. Specifically, we use the interaction network (Sect. 2.1) to obtain an initial segmentation, and this segment is then concatenated with the volume as the input of 3D nnU-Net. The quality-aware iterative refinement is also applied. In addition, we compare with a state-of-the-art method DeepIGeoS [28]. Several non-interactive methods are also included.

Interaction Simulation. Our approach can support various types of user interactions, which facilitates use in clinical routine. We study three common interactions: ***Scribbles*** provide sparse labels to describe the targets and rough outreach, ***Bounding Boxes*** outline the sizes and locations of targets, whereas ***Extreme Points*** [15] outline a more compact area of a target by labeling its *leftmost*, *rightmost*, *top*, *bottom* pixels. To simulate scribbles, we manually label

the data in KiTS_{19} and MSD, resulting in 3,585 slices. Bounding boxes and extreme points can be easily simulated from ground-truths with relaxations. We train an independent f_{In} for each of these interaction types. In the first interaction round, we compute a rough ROI according to user input. Then we treat all the pixels out of the enlarged ROI as the background, and the pixels specified by scribbles, or in regions of bounding boxes and extreme points as foreground. We encode user guidance as a binary image $\mathbf{M}$ (Sect. 2.1) for input.

Table 1. Quantitative results (DSC %) on (left) MSD [24] and (right) KiTS_{19} [9] `val`.

method	lung cancer	colon cancer	method	kidney (organ)	kidney (tumor)
non-interactive methods:			*non-interactive methods*		
C2FNAS [30]	70.4	58.9	Mu et al. [17]	**97.4**	78.9
3D nnU-Net [10]	66.9	56.0	3D nnU-Net [10]	96.9	85.7
interactive methods:			*interactive methods*		
Interactive 3D nnU-Net [10]			Interactive 3D nnU-Net [10]		
scribbles	73.9	68.1	scribbles	94.5	86.3
bounding boxes	74.7	68.5	bounding boxes	95.3	86.8
extreme points	75.1	69.8	extreme points	95.6	87.6
DeepIGeoS [28]			DeepIGeoS [28]		
scribbles	76.6	72.3	scribbles	95.7	87.6
bounding boxes	77.2	73.0	bounding boxes	96.4	88.5
extreme points	77.5	73.2	extreme points	96.7	88.9
Ours			**Ours**		
scribbles	80.9	79.7	scribbles	96.9	88.2
bounding boxes	81.5	79.3	bounding boxes	97.0	88.4
extreme points	**82.0**	**80.4**	extreme points	97.0	**89.1**

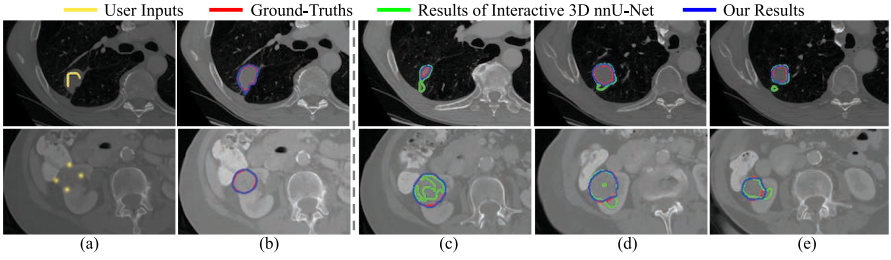

Fig. 2. Qualitative results of our approach *v.s.* Interactive 3D nnU-Net on two samples in MSD-Lung (row #1) and KITS_{19} (row #2), using scribbles and extreme points as supervision, respectively. (a) Interactive slices; (b) Results of interactive slices using the interaction network; (c)–(e) Results of other slices. Zoom in for details.

Training and Testing Details. Our engine is implemented in PyTorch. We use the same settings as [31] to train f_{In} (Sect. 2.1). The quality-aware memory network f_{Mem} (Sect. 2.2) is trained using Adam with learning rate 1e–5 and batch size 8 for 120 epochs. To make a training sample, we randomly sample 5 temporally ordered slices from a 3D image. During training, the memory is dynamically updated by adding the slice and mask at the previous step to the memory for the next slice.

During inference, simulated user hints are provided to f_{IN} for an initial segmentation of the interactive slice. Then, for each query slice, we put this interactive slice and the previous slice with corresponding segmentation mask into the memory as the most important reference information. In addition, we save a new memory item every N slices, where N is empirically set to 5. We do not add all slices and corresponding masks into memory to avoid large storage and computational costs. In this way, our memory network achieves the effect of online learning and adaption without additional training.

Quantitative and Qualitative Results. Table 1 (left) reports segmentation results of various methods on MSD `val`. For interactive methods, we report results at the 6^{th} round which well balances accuracy and efficiency. It can be seen that our method leads to consistent performance gains over the baselines. Specifically, our approach significantly outperforms Interactive 3D nnU-Net by more than **7%** for lung cancer and **10%** for colon cancer, and outperforms DeepIGeoS [28] by more than **4%** and **7%**, respectively. Moreover, for different types of interaction, our method produces very similar performance, revealing its high robustness to user input. Table 1 (right) presents performance comparisons on KiTS_{19} `val`. The results demonstrate that, for kidney tumor segmentation, our engine generally outperforms the baseline models. The improvements are lower than seen for the MSD dataset due to the fact that the initial segmentation is already of high quality resulting in smaller dice score gains for adjustments.

Figure 2 depicts qualitative comparisons of our approach against Interactive 3D nnU-Net on representative examples from MSD and KITS_{19}. As seen, our approach produces more accurate segmentation results than the competitor.

Table 2. Ablation study of the quality assessment module in terms of DSC (%).

Variant	MSD (lung)	MSD (colon)	KiTS_{19} (tumor)
Oracle	81.4	80.4	89.1
Random	80.1	77.5	86.8
Quality assess.	81.3	79.7	88.6

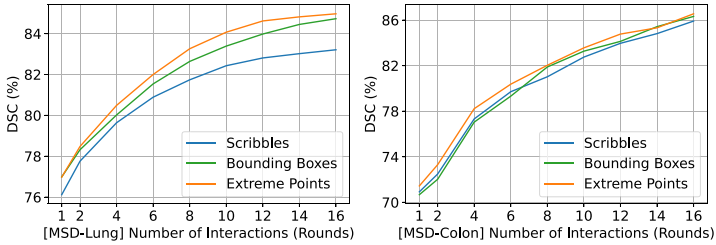

Fig. 3. The impact of number of interactions on MSD Lung (left) and Colon (right).

Efficacy of Quality Assessment Module. The quality assessment module empowers the engine to automatically select informative slices for iterative correction. To prove its efficacy, we design two baselines: 'oracle' selects the worst segmented slice by comparing the masks with corresponding ground-truths, while 'random' conducts random selection. As reported in Table 2, our method (*i.e.*, quality assessment module) significantly outperforms 'random' across three sets, and is comparable to 'oracle', proving its effectiveness.

Impact of Multi-round Refinement. Fig. 3 shows DSC scores with growing number of interactions on lung and colon subsets of MSD. We observe that multi-round refinement is crucial for achieving higher segmentation performance, and the performance becomes almost marginal at the 16^{th} round.

Runtime Analysis. For a 3D volume with size $512 \times 512 \times 100$, our method needs 5.13 s on average for one-round segmentation on a NVIDIA 2080Ti GPU, whereas Interactive 3D nnU-Net needs 200 s. Hence our engine enables a significant increase in inference speed.

4 Conclusion

This work presents a novel interactive segmentation engine for 3D medical volumes. The key component is a memory-augmented neural network, which employs an external memory for accurate and efficient 3D segmentation. Moreover, the quality-aware module empowers the engine to automatically select informative slices for user feedback, which we believe is an important added value of the memory network. Experiments on two public datasets show that our engine outperforms other alternatives while having a much faster inference speed.

Acknowledgment. This research was supported in part by the Varian Research Grant and Beijing Natural Science Foundation (No. L191004).

References

1. Boykov, Y.Y., Jolly, M.P.: Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In: ICCV, pp. 105–112 (2001)
2. Bredell, G., Tanner, C., Konukoglu, E.: Iterative interaction training for segmentation editing networks. In: International Workshop on Machine Learning in Medical Imaging, pp. 363–370 (2018)
3. 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
4. Criminisi, A., Sharp, T., Blake, A.: GeoS: geodesic image segmentation. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008. LNCS, vol. 5302, pp. 99–112. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-88682-2_9
5. Grady, L.: Random walks for image segmentation. IEEE TPAMI **28**(11), 1768–1783 (2006)
6. Grady, L., Schiwietz, T., Aharon, S., Westermann, R.: Random walks for interactive organ segmentation in two and three dimensions: implementation and validation. In: Duncan, J.S., Gerig, G. (eds.) MICCAI 2005. LNCS, vol. 3750, pp. 773–780. Springer, Heidelberg (2005). https://doi.org/10.1007/11566489_95
7. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: CVPR, pp. 9729–9738 (2020)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR, pp. 770–778 (2016)
9. Heller, N., et al.: The KiTS19 challenge data: 300 kidney tumor cases with clinical context, CT semantic segmentations, and surgical outcomes. arXiv preprint arXiv:1904.00445 (2019)
10. Isensee, F., et al.: nnU-Net: self-adapting framework for u-net-based medical image segmentation. arXiv preprint arXiv:1809.10486 (2018)
11. Kitrungrotsakul, T., Yutaro, I., Lin, L., Tong, R., Li, J., Chen, Y.W.: Interactive deep refinement network for medical image segmentation. arXiv preprint arXiv:2006.15320 (2020)
12. Kumar, A., et al.: Ask me anything: dynamic memory networks for natural language processing. In: ICML, pp. 1378–1387 (2016)
13. Liao, X., et al.: Iteratively-refined interactive 3D medical image segmentation with multi-agent reinforcement learning. In: CVPR, pp. 9394–9402 (2020)
14. Lu, X., Wang, W., Danelljan, M., Zhou, T., Shen, J., Van Gool, L.: Video object segmentation with episodic graph memory networks. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12348, pp. 661–679. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58580-8_39
15. Maninis, K.K., Caelles, S., Pont-Tuset, J., Van Gool, L.: Deep extreme cut: from extreme points to object segmentation. In: CVPR, pp. 616–625 (2018)
16. Milletari, F., Navab, N., Ahmadi, S.A.: V-Net: fully convolutional neural networks for volumetric medical image segmentation. In: 3DV, pp. 565–571 (2016)
17. Mu, G., Lin, Z., Han, M., Yao, G., Gao, Y.: Segmentation of kidney tumor by multi-resolution VB-nets (2019)
18. Olabarriaga, S.D., Smeulders, A.W.: Interaction in the segmentation of medical images: a survey. MedIA **5**(2), 127–142 (2001)

19. Qin, X., Zhang, Z., Huang, C., Gao, C., Dehghan, M., Jagersand, M.: BASNet: boundary-aware salient object detection. In: CVPR, pp. 7479–7489 (2019)
20. Rajchl, M., et al.: DeepCut: object segmentation from bounding box annotations using convolutional neural networks. IEEE TMI **36**(2), 674–683 (2016)
21. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
22. Sakinis, T., et al.: Interactive segmentation of medical images through fully convolutional neural networks. arXiv preprint arXiv:1903.08205 (2019)
23. Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., Lillicrap, T.: Meta-learning with memory-augmented neural networks. In: ICML, pp. 1842–1850 (2016)
24. Simpson, A.L., et al.: A large annotated medical image dataset for the development and evaluation of segmentation algorithms. arXiv preprint arXiv:1902.09063 (2019)
25. Sukhbaatar, S., Szlam, A., Weston, J., Fergus, R.: End-to-end memory networks. arXiv preprint arXiv:1503.08895 (2015)
26. Sun, J., et al.: Interactive medical image segmentation via point-based interaction and sequential patch learning. arXiv preprint arXiv:1804.10481 (2018)
27. Wang, G., et al.: Interactive medical image segmentation using deep learning with image-specific fine tuning. IEEE TMI **37**(7), 1562–1573 (2018)
28. Wang, G., et al.: DeepIGeoS: a deep interactive geodesic framework for medical image segmentation. IEEE TPAMI **41**(7), 1559–1572 (2018)
29. Wang, W., Zhou, T., Yu, F., Dai, J., Konukoglu, E., Van Gool, L.: Exploring cross-image pixel contrast for semantic segmentation. arXiv preprint arXiv:2101.11939 (2021)
30. Yu, Q., et al.: C2FNAS: coarse-to-fine neural architecture search for 3D medical image segmentation. In: CVPR (2020)
31. Zhang, S., Liew, J.H., Wei, Y., Wei, S., Zhao, Y.: Interactive object segmentation with inside-outside guidance. In: CVPR, pp. 12234–12244 (2020)
32. Zhao, F., Xie, X.: An overview of interactive medical image segmentation. Ann. BMVA **2013**(7), 1–22 (2013)
33. Zhou, Y., Xie, L., Shen, W., Wang, Y., Fishman, E.K., Yuille, A.L.: A fixed-point model for pancreas segmentation in abdominal CT scans. In: Descoteaux, M., Maier-Hein, L., Franz, A., Jannin, P., Collins, D.L., Duchesne, S. (eds.) MICCAI 2017. LNCS, vol. 10433, pp. 693–701. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66182-7_79
34. Zhou, Z., Siddiquee, M.M.R., Tajbakhsh, N., Liang, J.: UNet++: a nested u-net architecture for medical image segmentation. In: Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support, pp. 3–11 (2018)