DeepIGeoS: A Deep Interactive Geodesic Framework for Medical Image Segmentation

Guotai Wang, Maria A. Zuluaga, Wenqi Li, Rosalind Pratt, Premal A. Patel, Michael Aertsen, Tom Doel, Anna L. David, Jan Deprest, Sébastien Ourselin, and Tom Vercauteren

Abstract—Accurate medical image segmentation is essential for diagnosis, surgical planning and many other applications. Convolutional Neural Networks (CNNs) have become the state-of-the-art automatic segmentation methods. However, fully automatic results may still need to be refined to become accurate and robust enough for clinical use. We propose a deep learning-based interactive segmentation method to improve the results obtained by an automatic CNN and to reduce user interactions during refinement for higher accuracy. We use one CNN to obtain an initial automatic segmentation, on which user interactions are added to indicate mis-segmentations. Another CNN takes as input the user interactions with the initial segmentation and gives a refined result. We propose to combine user interactions with CNNs through geodesic distance transforms, and propose a resolution-preserving network that gives a better dense prediction. In addition, we integrate user interactions as hard constraints into a back-propagatable Conditional Random Field. We validated the proposed framework in the context of 2D placenta segmentation from fetal MRI and 3D brain tumor segmentation from FLAIR images. Experimental results show our method achieves a large improvement from automatic CNNs, and obtains comparable and even higher accuracy with fewer user interventions and less time compared with traditional interactive methods.

Index Terms—Interactive image segmentation, convolutional neural network, geodesic distance, conditional random fields

1 Introduction

Segmentation of anatomical structures is an essential task for a range of medical image processing applications such as image-based diagnosis, anatomical structure modeling, surgical planning and guidance. During the past decades, researchers have developed many automatic segmentation approaches [1]. However, fully automatic segmentation methods rarely achieve sufficiently accurate and robust results to be clinically useful [2]. This is typically due to poor image quality (with noise, partial volume effect, artifacts and low contrast), large variations among patients, inhomogeneous appearances brought by pathology, and variability of protocols among clinicians leading to different definitions of a given structure boundary. To address the limitations of automatic segmentation approaches, interactive segmentation methods are desirable as they allow higher accuracy and robustness in many applications [3], such as planning of radiotherapy treatment of brain tumors [4]. As providing manual annotations for segmentation is time-consuming and labor-intensive, an efficient interactive segmentation tool is of great importance for practical use.

A good interactive segmentation method should obtain accurate results efficiently with as few user interactions as possible, leading to interaction efficiency. For such a method, there are mainly two factors that have a critical impact on its performance and usefulness. The first is the type of user interactions used as input to the method, and the second is the algorithm's underpinning model. Despite the large number of existing interactive segmentation methods [3], most of them are confronted by requiring a large amount of user interactions and long user time, or limited learning ability with their underpinning models.

For example, the widely used ITK-SNAP [5] takes user-provided seed pixels or blobs as a starting point and employs an active contour model for segmentation. It requires most of the user interactions to be given at the beginning and the underpinning model can hardly be refined with additional user interactions once an initial segmentation is obtained. SlicSeg [6] accepts user-provided scribbles in a single start slice to train an Online Random Forest for 3D segmentation, but lacks in flexibility to allow further user-editing. Random Walks [7] and Graph Cuts [8] learn from scribbles and allow the user to provide additional scribbles for refinement. They use Random Walker

- *R. Pratt is with the Translational Imaging Group, Wellcome EPSRC Centre for Interventional and Surgical Sciences (WEISS), University College London, London WC1E 6BT, United Kingdom, and also with the Institute for Women's Health, University College London, London WC1E 6BT, United Kingdom. E-mail: rosalind.pratt@ucl.ac.uk.*
- *G. Wang, M.A. Zuluaga, W. Li, P.A. Patel, T. Doel, S. Ourselin, and T. Vercauteren are with Translational Imaging Group, Wellcome EPSRC Centre for Interventional and Surgical Sciences (WEISS), University College London, London WC1E 6BT, United Kingdom. E-mail: {guotai.wang.14, wenqi.li, premal.patel, t.doel, s.ourselin, t.vercauteren}@ucl.ac.uk, maria.zuluaga@gmail.com.*
- *A.L. David is with the Institute for Women's Health, University College London, London WC1E 6BT, United Kingdom. E-mail: a.david@ucl.ac.uk.*
- *M. Aertsen is with the Department of Radiology, University Hospitals KU Leuven, Leuven 3000, Belgium. E-mail: michaelaertsen@gmail.com.*
- *J. Deprest is with the Department of Obstetrics, University Hospitals KU Leuven, Leuven 3000, Belgium. E-mail: Jan.Deprest@uzleuven.be.*

Manuscript received 9 Feb. 2017; revised 17 Apr. 2018; accepted 22 May 2018. Date of publication 31 May 2018; date of current version 12 June 2019. (Corresponding author: Guotai Wang.)
Recommended for acceptance by P. Kohli.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2018.2840695

and Gaussian Mixture Model (GMM) as the underpinning model. However, they need a large number of scribbles to get satisfactory segmentation. GrabCut [9] works with a user-provided bounding box to start the segmentation and requires fewer interactions compared with Graph Cuts [8], but the performance is still limited by the representativity of the underpinning GMM. Therefore, a more efficient way for user interactions and a better underpinning model are highly demanded for interactive medical image segmentation.

Recently, deep learning with convolutional neural networks (CNNs) has achieved the state-of-the-art performance in many image analysis applications [10]. With the high-quality automatic segmentation results achieved by Fully Convolutional Network (FCN) [11], U-Net [12], V-Net [13], HighRes3DNet [14] and DeepMedic [15], etc., CNNs have been shown to be powerful learning models for segmentation tasks. However, they have not yet been adapted to interactive medical image segmentation.

In this paper, we propose a novel interactive method for 2D and 3D medical image segmentation that leverages deep learning. We propose a two-stage pipeline, where a first CNN automatically obtains an initial segmentation and a second CNN refines the initial segmentation by taking advantages of a small number of user interactions that we encode as geodesic distance maps. We refer to the proposed interactive segmentation method as Deep Interactive Geodesic Framework (DeepIGeoS).

Compared with existing interactive segmentation methods, DeepIGeoS has several appealing properties. First, it uses a more powerful underpinning learning model, i.e., CNN with automatic feature learning to take advantages of knowledge from a large training set. Second, it requires far fewer user interactions, as the method starts with a high-quality automatic segmentation and only needs user-provided clicks or short scribbles in the refinement stage. Third, it is efficient and can respond to user interactions in real time, which leads to very short user time.

The contributions of this work are four-fold: 1) We propose a deep CNN-based interactive framework for 2D and 3D medical image segmentation; 2) to make CNNs suitable for interactive segmentation with high efficiency and accuracy, we propose two new networks for 2D and 3D images respectively; 3) we propose to integrate user interactions with CNNs by converting them into geodesic distance maps as part of the input for CNNs, and use them as constraints for a trainable Conditional Random Field (CRF); 4) we demonstrate that CNNs lead to state-of-the-art performance for interactive medical image segmentation, with far less user efforts and user time than existing methods.

2 Related Works

2.1 Image Segmentation Based on CNNs

Typical CNNs [16], [17], [18] were originally designed for image classification tasks. Some early works adapted such networks for pixel labeling with patch or region-based methods [19], [20]. Such methods achieved higher accuracy than traditional methods that relied on hand-crafted features. However, they suffered from inefficiency for testing. FCNs [11] take an entire image as input and give a dense segmentation. In order to overcome the problem of loss of spatial resolution due to multi-stage max-pooling and downsampling, it uses a stack of deconvolution (a.k.a. upsampling) layers and activation functions to upsample the feature maps. Inspired by the convolution and deconvolution framework of FCNs, a U-shape network (U-Net) [12] and its 3D version [21] were proposed for biomedical image segmentation. A similar network (V-Net) [13] was proposed to segment the prostate from 3D MRI volumes.

To overcome the drawbacks of successive max-pooling and downsampling that lead to a loss of feature map resolution, dilated convolution [22], [23] was proposed to preserve the resolution of feature maps and enlarge the receptive field to incorporate larger contextual information. In [24], a stack of dilated convolutions was used for object tracking and semantic segmentation. Dilated convolution has also been used for instance-sensitive segmentation [25] and action detection from video frames [26].

Multi-scale features extracted from CNNs have been shown to be effective for improving segmentation accuracy [11], [22], [23]. One way of obtaining multi-scale features is to pass several scaled versions of the input image through the same network. The features from all the scales can be fused for pixel classification [27]. In [15], [19], the features of each pixel were extracted from two concentric patches with different sizes. In [28], multi-scale images at different stages were fed into a recurrent convolutional neural network. Another widely used way to obtain multi-scale features is exploiting the feature maps from different levels of a CNN. For example, in [29], features from intermediate layers are concatenated for segmentation and localization. In [11], [22], predictions from the final layer are combined with those from previous layers.

2.2 Interactive Image Segmentation

Interactive image segmentation has been widely used in various applications [30], [31], [32]. There are many kinds of user interactions, such as click-based [33], contour-based [34] and bounding box-based methods [9]. Drawing scribbles is user-friendly and particularly popular, e.g., in Graph Cuts [8], GeoS [35], [36], and Random Walks [7]. However, most of these methods rely on low-level features and require a relatively large amount of user interactions to deal with images with low contrast and ambiguous boundaries. Machine learning methods [6], [37], [38] have been proposed to learn from user interactions. They can achieve higher segmentation accuracy with fewer user interactions. However, they are limited by hand-crafted features that depend on the user's experience.

Recently, using deep CNNs to improve interactive segmentation has attracted increasing attention due to CNNs' automatic feature learning and high performance. For instance, 3D U-Net [21] learns from sparsely annotated images and can be used for semi-automatic segmentation. ScribbleSup [39] also trains CNNs for semantic segmentation supervised by scribbles. DeepCut [32] employs user-provided bounding boxes as annotations to train CNNs for the segmentation of fetal MRI. However, these methods are not fully interactive for testing since they do not accept further interactions for refinement. In [40], a deep interactive object selection method was proposed where user-provided clicks are transformed into euclidean distance maps and

then concatenated with the input of FCNs. However, the euclidean distance does not take advantage of image context information. In contrast, the geodesic distance transform [35], [36], [41] encodes spatial regularization and contrast-sensitivity but it has not been used for CNNs.

2.3 CRFs for Spatial Regularization

Graphical models such as CRFs [22], [42], [43] have been widely used to enhance segmentation accuracy by introducing spatial consistency. In [42], spatial regularization was obtained by minimizing the Potts energy with a min-cut/max-flow algorithm. In [43], the discrete max-flow problem was mapped to its continuous optimization formulation. Such methods encourage segmentation consistency between adjacent pixel pairs with high similarity. In order to better model long-range connections within the image, a fully connected CRF was used in [44] to establish pairwise potentials on all pairs of pixels in the image. To make the inference of this CRF efficient, the pairwise edge potentials were defined by a linear combination of Gaussian kernels in [45]. The parameters of CRFs in these works were manually tuned or inefficiently learned by grid search. In [46], a maximum margin learning method was proposed to learn CRFs using Graph Cuts. Other methods including structured output Support Vector Machines [47], approximate marginal inference [48] and gradient-based optimization [49] were also proposed to learn parameters in CRFs. They treat the learning of CRFs as an independent step after the training of classifiers.

The CRF-RNN network [50] formulated dense CRFs as RNNs so that the CNNs and CRFs can be jointly trained in an end-to-end system for segmentation. However, the pairwise potentials in [50] are limited to weighted Gaussians and not all the parameters are trainable due to the Permutohedral lattice implementation [51]. In [52], a Gaussian Mean Field (GMF) network was proposed and combined with CNNs where all the parameters are trainable. More freeform pairwise potentials for a pair of super-pixels or image patches were proposed in [27], [53], but such CRFs have a low resolution. In [54], a generic CNN-CRF model was proposed to handle arbitrary potentials for labeling body parts in depth images. However, it has not yet been validated with other segmentation applications.

3 Method

The proposed DeepIGeoS for deep interactive segmentation is depicted in Fig. 1. To minimize the number of user interactions, we propose a two-stage framework: In Stage 1, which is an automatic segmentation problem and requires fast inference, one CNN (P-Net) automatically proposes an initial segmentation. In Stage 2, the user checks the initial segmentation and gives some interactions (clicks and short scribbles) to indicate mis-segmented regions, and a second CNN (R-Net) refines the segmentation by taking as input the original image, the initial segmentation and the user interactions. The user is allowed to give clicks/scribbles to refine the result more than one time through R-Net. P-Net and R-Net use a resolution-preserving structure that captures high-level features from a large receptive field without loss of resolution. They share the same structure except the difference in the input dimensions. Differently from

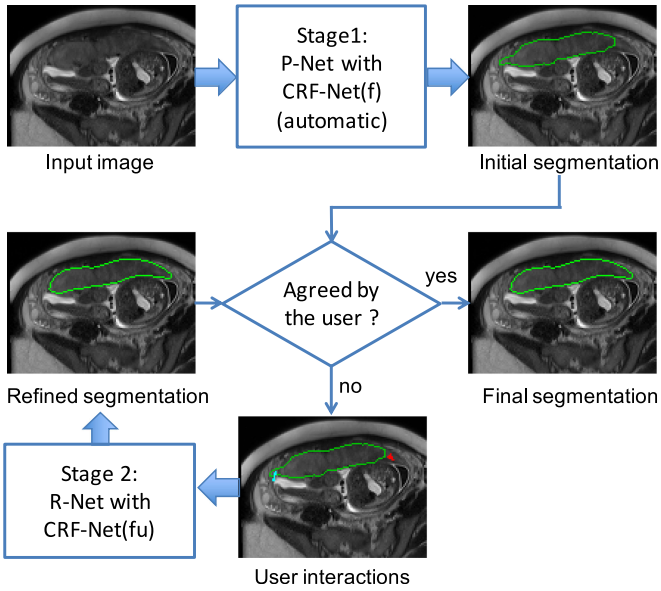


Fig. 1. Overview of the proposed interactive segmentation framework with two stages. Stage 1: P-Net automatically proposes an initial segmentation. Stage 2: R-Net refines the segmentation with user interactions indicating mis-segmentations. CRF-Net(f) is our proposed back-propagatable CRF that uses freeform pairwise potentials. It is extended to be CRF-Net(fu) that employs user interactions as hard constraints.

previous works [55] that re-train the learning model each time when new user interactions are given, the proposed R-Net is only trained with user interactions once since it takes a considerable time to re-train a CNN model with a large training set.

To make the segmentation result more spatially consistent and to use scribbles as hard constraints, both P-Net and R-Net are connected with a CRF, which is modeled as an RNN (CRF-Net) so that it can be trained jointly with P-Net/R-Net by back-propagation. We use freeform pairwise potentials in the CRF-Net. The way user interactions are used is presented in Section 3.1. The structures of 2D/3D P-Net and R-Net are detailed in Section 3.2. In Section 3.3, we describe the implementation of our CRF-Net. Training details are described in Section 3.4.

3.1 User Interaction-Based Geodesic Distance Maps

In Stage 2 of our method, scribbles are provided by the user to refine the initial automatic segmentation obtained by P-Net in Stage 1. A scribble labels a set of pixels as the foreground or background. Interactions with the same label are converted into a distance map. In [40], the euclidean distance was used due to its simplicity. However, the euclidean distance treats each direction equally and does not take the image context into account. In contrast, the geodesic distance helps to better differentiate neighboring pixels with different appearances, and improves label consistency in homogeneous regions [36]. GeoF [41] uses the geodesic distance to encode variable dependencies in the feature space and it is combined with Random Forests for semantic segmentation. However, it is not designed to deal with user interactions. We propose to encode user interactions via geodesic distance transforms for CNN-based segmentation.

Suppose $\mathcal{S}_f$ and $\mathcal{S}_b$ represent the set of pixels belonging to foreground scribbles and background scribbles, respectively. Let i be a pixel in an image $\mathbf{I}$, then the

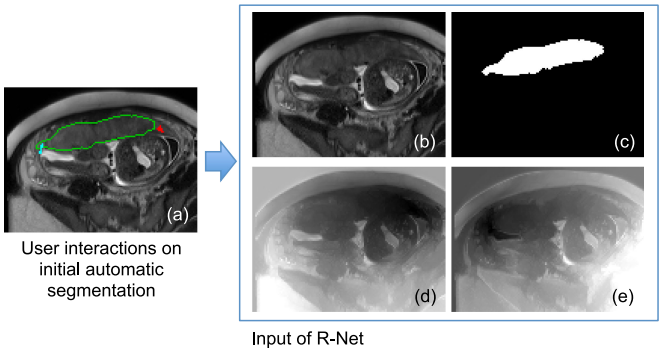


Fig. 2. Input of R-Net in Stage 2. (a) The user provides clicks/scribbles to correct foreground (red) and background (cyan) on the initial automatic segmentation. (d) and (e) are geodesic distance maps based on foreground and background interactions, respectively. The original image (b) is combined with the initial automatic segmentation (c) and the geodesic distance maps (d), (e) by channel-concatenation and used as the input of R-Net.

unsigned geodesic distance from i to the scribble set $\mathcal{S}(\mathcal{S} \in \{\mathcal{S}_f, \mathcal{S}_b\})$ is:

$$G(i, \mathcal{S}, \mathbf{I}) = \min_{j \in \mathcal{S}} D_{geo}(i, j, \mathbf{I}) \tag{1}$$

$$D_{geo}(i, j, \mathbf{I}) = \min_{p \in \mathcal{P}_{i,j}} \int_0^1 \|\nabla \mathbf{I}(p(s)) \cdot \mathbf{u}(s)\| ds, \tag{2}$$

where $\mathcal{P}_{i,j}$ is the set of all paths between pixel i and j. p is one feasible path and it is parameterized by $s \in [0, 1]$. $\mathbf{u}(s) = p'(s)/\|p'(s)\|$ is a unit vector that is tangent to the direction of the path. If no scribbles are drawn for either the foreground or background, the corresponding geodesic distance map is filled with random numbers.

Fig. 2 shows an example of geodesic distance transforms of user interactions. The geodesic distance maps of user interactions and the initial automatic segmentation have the same size as $\mathbf{I}$. They are concatenated with the raw channels of $\mathbf{I}$ so that a concatenated image with C_I+3 channels is obtained, which is used as the input of the refinement network R-Net.

3.2 Resolution-Preserving CNNs using Dilated Convolution

CNNs in our method are designed to capture high-level features from a large receptive field without the loss of resolution of the feature maps. They are adapted from VGG-16 [17] and made resolution-preserving. Fig. 3 shows the structure of 2D and 3D P-Net. In 2D P-Net, the first 13 convolution layers are grouped into five blocks. The first and second blocks have two convolution layers respectively, and each of the remaining blocks has three convolution layers. The size of the convolution kernel is fixed as 3×3 in all these convolution layers. 2D R-Net uses the same structure as 2D P-Net except that its number of input channels is C_I+3 and it employs user interactions in the CRF-Net. To obtain an exponential increase of the receptive field, VGG-16 uses a max-pooling and downsampling layer after each block. However, this implementation would decrease the resolution of feature maps exponentially. Therefore, to preserve resolution through the network, we remove the max-pooling and downsampling layers and use dilated convolution in each block.

Let $\mathbf{I}$ be a 2D image of size $W \times H$, and let K_{rq} be a square dilated convolution kernel with a size of $(2r + 1)\times(2r + 1)$ and a dilation parameter q, where $r \in \mathbb{Z}$ and $q \in \mathbb{Z}$. The dilated convolution of $\mathbf{I}$ with K_{rq} is defined as:

$$\mathbf{I}_c(x, y) = \sum_{i=-r}^{r} \sum_{j=-r}^{r} \mathbf{I}(x - qi, y - qj) K_{rq}(i + r, j + r) \tag{3}$$

For 2D P-Net/R-Net, we set r to 1 for block 1 to block 5, so the size of a convolution kernel becomes 3×3. The dilation parameter in block i is set to:

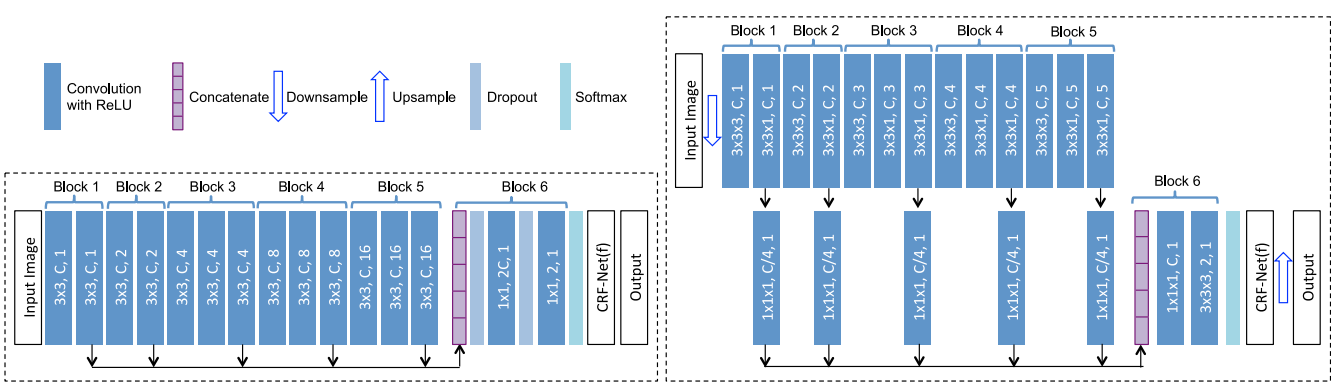
$$q_i = d \times 2^{i-1}, i = 1, 2, \ldots, 5, \tag{4}$$

where $d \in \mathbb{Z}$ is a system parameter controlling the base dilation parameter of the network. We set d=1 in experiments.

The receptive field of a dilated convolution kernel K_{rq} is $(2rq+1)\times(2rq+1)$. Let $R_i \times R_i$ denote the receptive field of block i. R_i can be computed as:

$$R_i = 2\Big(\sum_{j=1}^{i} \tau_j \times (rq_j)\Big) + 1, i = 1, 2, \ldots, 5, \tag{5}$$

where τ_j is the number of convolution layers in block j, with a value of 2, 2, 3, 3, 3 for the five blocks respectively. When $r = 1$, the receptive field size of each block is $R_1 = 4d + 1$, $R_2 =$



(a) P-Net (2D) for segmentation of a 2D slice

(b) P-Net (3D) for segmentation of a 3D volume

Fig. 3. The CNN structure of 2D/3D P-Net with CRF-Net(f). The parameters of convolution layers are (kernel size, output channels, dilation) in dark blue rectangles. Block 1 to block 6 are resolution-preserving. 2D/3D R-Net uses the same structure as 2D/3D P-Net except its input has three additional channels shown in Fig. 2 and the CRF-Net(f) is replaced by the CRF-Net(fu) (Section 3.3).

$12d+1$, $R_3 = 36d+1$, $R_4 = 84d+1$, $R_5 = 180d+1$, respectively. Thus, these blocks capture features at different scales.

The stride of each convolution layer is set to 1. The number of output channels of convolution in each block is set to a fixed number C. In order to use multi-scale features, we concatenate the features from different blocks to get a composed feature of length $5C$. This feature is fed into a classifier that is implemented by two additional layers as shown in block 6 in Fig. 3a. These two layers use convolution kernels with size of 1×1 and dilation parameter of 0. Block 6 gives each pixel an initial score of belonging to the foreground or background class. In order to get a more spatially consistent segmentation and add hard constraints when scribbles are given, we apply a CRF on the basis of the output from block 6. The CRF is implemented by a recurrent neural network (CRF-Net, detailed in Section 3.3), which can be jointly trained with P-Net or R-Net. The CRF-Net gives a regularized prediction for each pixel, which is fed into a cross entropy loss function layer.

Similar network structures are used by 3D P-Net/R-Net for 3D segmentation, as shown in Fig. 3b. To reduce the memory consumption for 3D images, we use one downsampling layer before the resolution-preserving layers and compress the output features of block 1 to 5 by a factor four via $1 \times 1 \times 1$ convolutions before the concatenation layer.

3.3 Back-Propagatable CRF-Net with Freeform Pairwise Potentials and User Constraints

In [50], a CRF based on RNN was proposed and it can be trained by back-propagation. Rather than using Gaussian functions, we extend this CRF so that the pairwise potentials can be freeform functions and we refer to it as CRF-Net (f). In addition, we integrate user interactions in our CRF-Net(f) in the interactive refinement context, which is referred to as CRF-Net(fu). The CRF-Net(f) is connected to P-Net and the CRF-Net(fu) is connected to R-Net.

Let $\mathbf{X}$ be the label map assigned to an image $\mathbf{I}$ with a label set $\mathcal{L} = \{0, 1, \ldots, L-1\}$. The Gibbs distribution $P(\mathbf{X}=\mathbf{x}|\mathbf{I}) = \frac{1}{Z(\mathbf{I})}\exp(-E(\mathbf{x}|\mathbf{I}))$ models the probability of $\mathbf{X}$ given $\mathbf{I}$ in a CRF, where $Z(\mathbf{I})$ is the normalization factor known as the partition function, and $E(\mathbf{x})$ is the Gibbs energy:

$$E(\mathbf{x}) = \sum_i \psi_u(x_i) + \sum_{(i,j)\in\mathcal{N}} \psi_p(x_i, x_j), \tag{6}$$

where the unary potential $\psi_u(x_i)$ measures the cost of assigning label x_i to pixel i, and the pairwise potential $\psi_p(x_i, x_j)$ is the cost of assigning labels x_i, x_j to a pixel pair i, j. $\mathcal{N}$ is the set of all pixel pairs. In our method, the unary potential is obtained from P-Net or R-Net that gives classification scores for each pixel. The pairwise potential is:

$$\psi_p(x_i, x_j) = \mu(x_i, x_j) f(\tilde{\mathbf{f}}_{ij}, d_{ij}), \tag{7}$$

where d_{ij} is the euclidean distance between pixels i and j. $\mu(x_i, x_j)$ is the compatibility between the label of i and that of j represented by a matrix of size $L \times L$. $\tilde{\mathbf{f}}_{ij} = \mathbf{f}_i - \mathbf{f}_j$, where $\mathbf{f}_i$ and $\mathbf{f}_j$ represent the feature vectors of i and j, respectively. The feature vectors can either be learned by a network or be derived from image features such as spatial location with intensity values. For experiments, we used the latter one, as in [8], [45], [50] for simplicity and efficiency. $f(\cdot)$ is a function in terms of $\tilde{\mathbf{f}}_{ij}$ and d_{ij}. Instead of defining $f(\cdot)$ as a single Gaussian function [8] or a combination of several Gaussian functions [45], [50], we set it as a freeform function represented by a fully connected neural network (Pairwise-Net) which can be learned during training. The structure of Pairwise-Net is shown in Fig. 4. The input is a vector composed of $\tilde{\mathbf{f}}_{ij}$ and d_{ij}. There are two hidden layers and one output layer.

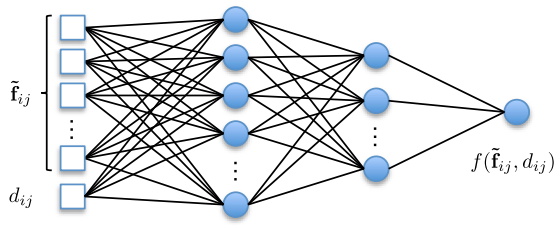


Fig. 4. The Pairwise-Net for pairwise potential function $f(\tilde{\mathbf{f}}_{ij}, d_{ij})$. $\tilde{\mathbf{f}}_{ij}$ is the difference of features between a pixel pair i and j. d_{ij} is the euclidean distance between them.

Graph Cuts [8], [46] can be used to minimize Eq. (6) when $\psi_p(\cdot)$ is submodular [56] such as when the segmentation is binary with $\mu(\cdot)$ being the delta function and $f(\cdot)$ being positive. However, this is not the case for our method since we learn $\mu(\cdot)$ and $f(\cdot)$ where $\mu(\cdot)$ may not be the delta function and $f(\cdot)$ could be negative. Continuous max-flow [43] can also be used for the minimization, but its parameters are manually designed. Alternatively, mean-field approximation [45], [50], [52] is often used for efficient inference of the CRF while allowing learning parameters by back-propagation. Instead of computing $P(\mathbf{X}|\mathbf{I})$ directly, an approximate distribution $Q(\mathbf{X}|\mathbf{I}) = \prod_i Q_i(x_i|\mathbf{I})$ is computed so that the KL-divergence $\mathbf{D}(Q||P)$ is minimized. This yields an iterative update of $Q_i(x_i|\mathbf{I})$ [45], [50], [52].

$$Q_i(x_i|\mathbf{I}) = \frac{1}{Z_i} e^{-E(x_i)} = \frac{1}{Z_i} e^{-\psi_u(x_i) - \phi_p(x_i)} \tag{8}$$

$$\phi_p(x_i = l|\mathbf{I}) = \sum_{l'\in\mathcal{L}} \mu(l, l') \sum_{(i,j)\in\mathcal{N}} f(\tilde{\mathbf{f}}_{ij}, d_{ij}) Q_j(l'|\mathbf{I}), \tag{9}$$

where $\mathcal{L}$ is the label set. i and j are a pixel pair. For the proposed CRF-Net(fu), with the set of user-provided scribbles $\mathcal{S}_{fb} = \mathcal{S}_f \cup \mathcal{S}_b$, we force the probability of pixels in the scribble set to be 1 or 0. The following equation is used as the update rule for each iteration:

$$Q_i(x_i|\mathbf{I}) = \begin{cases} 1 & \text{if } i \in \mathcal{S}_{fb} \text{ and } x_i = s_i \\ 0 & \text{if } i \in \mathcal{S}_{fb} \text{ and } x_i \neq s_i \\ \frac{1}{Z_i} e^{-E(x_i)} & \text{otherwise,} \end{cases} \tag{10}$$

where s_i denotes the user-provided label of a pixel i that is in the scribble set $\mathcal{S}_{fb}$. We follow the implementation in [50] to update Q through a multi-stage mean-field method in an RNN. Each mean-field layer splits Eq. (8) into four steps including message passing, compatibility transform, adding unary potentials and normalizing [50].

3.4 Implementation Details

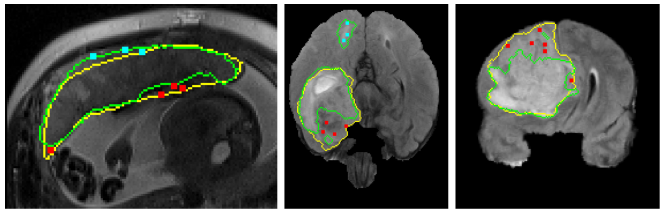
The raster-scan algorithm [36] was used to compute geodesic distance transforms by applying a forward pass scanning and a backward pass scanning with a 3×3 kernel for 2D and

a $3 \times 3 \times 3$ kernel for 3D. It is fast due to accessing the image memory in contiguous blocks. For the proposed CRF-Net with freeform pairwise potentials, two observations motivate us to use pixel connections based on local patches instead of full connections within the entire image. First, the permutohedral lattice implementation [45], [50] allows efficient computation of fully connected CRFs only when pairwise potentials are Gaussian functions. However, a method that relaxes the requirement of pairwise potentials as freeform functions represented by a network (Fig. 4) cannot use that implementation and therefore would be inefficient for fully connected CRFs. Suppose an image with size $M \times N$, a fully connected CRF has $MN(MN\text{-}1)$ pixel pairs. For a small image with $M = N = 100$, the number of pixel pairs would be almost 10^8, which requires not only a huge amount of memory but also long computational time. Second, though long-distance dependency helps to improve segmentation in most RGB images [22], [45], [50], this would be very challenging for medical images since the contrast between the target and background is often low [57]. In such cases, long-distance dependency may lead the label of a target pixel to be corrupted by the large number of background pixels with similar appearances. Therefore, to maintain a good efficiency and avoid long-distance corruptions, we define the pairwise connections for one pixel within a local patch centered on that. In our experiments, the patch size is set to 7×7 for 2D images and $5 \times 5 \times 3$ for 3D images.

We initialize $\mu(\cdot)$ as $\mu(x_i, x_j) = [x_i \neq x_j]$, where $[\cdot]$ is the Iverson Bracket [50]. A fully connected neural network (Pairwise-Net) with two hidden layers is used to learn the freeform pairwise potential function (Fig. 4). The first and second hidden layers have 32 and 16 neurons, respectively. In practice, this network is implemented by an equivalent fully convolutional neural network with 1×1 kernels for 2D or $1 \times 1 \times 1$ kernels for 3D. We use a pre-training step to initialize the Pairwise-Net with an approximation of a contrast sensitive function [8]:

$$f_0(\tilde{\mathbf{f}}_{ij}, d_{ij}) = \exp\left(-\frac{||\tilde{\mathbf{f}}_{ij}||^2}{2\sigma^2 \cdot F}\right) \cdot \frac{\omega}{d_{ij}}, \tag{11}$$

where F is the dimension of the feature vectors $\mathbf{f}_i$ and $\mathbf{f}_j$, and ω and σ are two parameters controlling the magnitude and shape of the initial pairwise function respectively. In this initialization step, we set σ to 0.08 and ω to 0.5 based on experience. Similar to [45], [50], [58], we set $\mathbf{f}_i$ and $\mathbf{f}_j$ as values in input channels (i.e, image intensity in our case) of P-Net for simplicity of implementation and for obtaining contrast-sensitive pairwise potentials. To pre-train the Pairwise-Net we generate a training set $T' = \{X', Y'\}$ with 100k samples, where X' is the set of features simulating the concatenated $\tilde{\mathbf{f}}_{ij}$ and d_{ij}, and Y' is the set of prediction values simulating $f_0(\tilde{\mathbf{f}}_{ij}, d_{ij})$. For each sample s in T', the feature vector x'_s has a dimension of $F + 1$ where the first F dimensions represent the value of $\tilde{\mathbf{f}}_{ij}$ and the last dimension denotes d_{ij}. The cth channel of x'_s is filled with a random number k', where $k' \sim Norm(0, 2)$ for $c \leq F$ and $k' \sim U(0, 8)$ for $c = F + 1$. The ground truth of prediction value y'_s for x'_s is obtained by Eq. (11). After generating X' and Y', we use a Stochastic Gradient Descent (SGD) algorithm with a quadratic loss function to pre-train the Pairwise-Net.



(a) (b) (c)

Fig. 5. Simulated user interactions on training images for placenta (a) and brain tumor (b, c). Green: automatic segmentation given by P-Net with CRF-Net(f). Yellow: ground truth. Red (cyan): simulated clicks on under-segmentation (over-segmentation).

For pre-processing, all the images are normalized by the mean value and standard deviation of the training set. We apply data augmentation by vertical or horizontal flipping, random rotation with angle range [$-\pi/8$, $\pi/8$] and random zoom with scaling factor range [0.8, 1.25]. We use the cross entropy loss function and SGD algorithm for optimization with minibatch size 1, momentum 0.99 and weight decay 5×10^{-4}. The learning rate is halved every 5k iterations. Since a proper initialization of P-Net and CRF-Net(f) is helpful for a faster convergence of the joint training, we train the P-Net with CRF-Net(f) in three steps. First, the P-Net is pre-trained with initial learning rate 10^{-3} and maximal number of iterations 100k. Second, the Pairwise-Net in the CRF-Net (f) is pre-trained as described above. Third, the P-Net and CRF-Net(f) are jointly trained with initial learning rate 10^{-6} and maximal number of iterations 50k.

After the training of P-Net with CRF-Net(f), we automatically simulate user interactions to train R-Net with CRF-Net (fu). First, P-Net with CRF-Net(f) is used to obtain an automatic segmentation for each training image. It is compared with the ground truth to find mis-segmented regions. Then the user interactions on each mis-segmented region are simulated by randomly sampling n pixels in that region. Suppose the size of one connected under-segmented or over-segmented region is N_m, we set n for that region to 0 if $N_m < 30$ and $\lceil N_m/100 \rceil$ otherwise based on experience. Examples of simulated user interactions on training images are shown in Fig. 5. With these simulated user interactions on the initial segmentation of training data, the training of R-Net with CRF-Net(fu) is implemented through SGD, which is similar to the training of P-Net with CRF-Net(f).

We implemented our 2D networks by Caffe[1] [59] and 3D networks by Tensorflow[2] [60] using NiftyNet[3] [14]. Our training process was done via two 8-core E5-2623v3 Intel Haswells and two K80 NVIDIA GPUs and 128 GB memory. The testing process with user interactions was performed on a MacBook Pro (OS X 10.9.5) with 16 GB RAM and an Intel Core i7 CPU running at 2.5 GHz and an NVIDIA GeForce GT 750M GPU. A Matlab and PyQt GUI were developed for 2D and 3D interactive segmentation tasks, respectively.(See supplementary videos, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2018.2840695)

1. http://caffe.berkeleyvision.org
2. https://www.tensorflow.org
3. http://niftynet.io

4 Experiments

4.1 Comparison Methods and Evaluation Metrics

We first present the results obtained in Stage 1 of our method, then present the results obtained in Stage 2. For Stage 1, we compared our P-Net with FCN [11] and DeepLab [58] for 2D segmentation and DeepMedic [15] and HighRes3DNet [14] for 3D segmentation. Pre-trained models of FCN[4] and DeepLab[5] based on ImageNet were fine-tuned for 2D placenta segmentation. Since the input of FCN and DeepLab should have three channels, we duplicated each of the gray-level images twice and concatenated them into a three-channel image as the input. DeepMedic and HighRes3DNet were originally designed for multi-modality or multi-class 3D segmentation. We adapted them for single modality binary segmentation. We also compared 2D/3D P-Net with 2D/3D P-Net(b5) that only uses the features from block 5 (Fig. 3) instead of the concatenated multi-scale features. The proposed CRF-Net(f) with freeform pairwise potentials was compared with: 1). Dense CRF as an independent post-processing step for the output of P-Net. We followed the implementation in [15], [45], [58]. The parameters of this CRF were manually tuned based on a coarse-to-fine search scheme as suggested by [58], and 2). CRF-Net(g) which refers to the CRF that can be trained jointly with CNNs by using Gaussian pairwise potentials [50].

For Stage 2, which is the interactive refinement part, we compared three methods to deal with user interactions. 1). Min-cut user-editing [9], where the initial probability map (output of P-Net in our case) is combined with user interactions to solve an energy minimization problem with min-cut [8]; 2). Using the euclidean distance of user interactions in R-Net, which is referred to as R-Net (Euc), and 3). The proposed R-Net with the geodesic distance of user interactions.

We also compared DeepIGeoS with several other interactive segmentation methods. For 2D slices, DeepIGeoS was compared with: 1). Geodesic Framework [35] that computes a probability based on the geodesic distance from user-provided scribbles for pixel classification; 2). Graph Cuts [8] that models segmentation as a min-cut problem based on user interactions; 3). Random Walks [7] that assigns a pixel with a label based on the probability that a random walker reaches a foreground or background seed first, and 4). SlicSeg [6] that uses Online Random Forests to learn from the scribbles and predict the remaining pixels. For 3D images, DeepIGeoS was compared with GeoS [36] and ITK-SNAP [5]. Two users (an Obstetrician and a Radiologist) respectively used these interactive methods to segment every test image until the result was visually acceptable.

For quantitative evaluation, we measured the Dice score and the average symmetric surface distance (ASSD).

$$\text{Dice} = \frac{2|\mathcal{R}_a \cap \mathcal{R}_b|}{|\mathcal{R}_a| + |\mathcal{R}_b|}, \tag{12}$$

where $\mathcal{R}_a$ and $\mathcal{R}_b$ represent the region segmented by the algorithm and the ground truth, respectively.

4. https://github.com/shelhamer/fcn.berkeleyvision.org
5. https://bitbucket.org/deeplab/deeplab-public

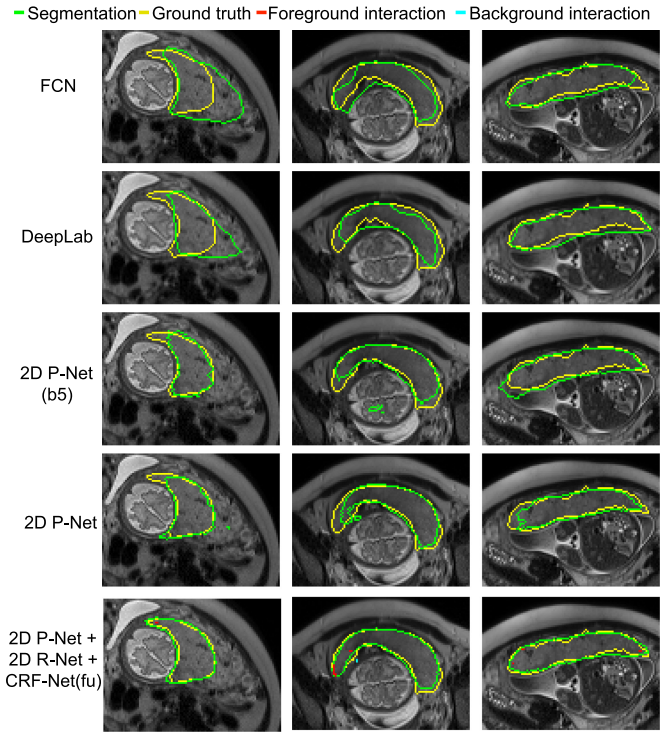


Fig. 6. Visual comparison of different networks in Stage 1 of 2D placenta segmentation. The last row shows interactively refined results by DeepIGeoS.

$$\text{ASSD} = \frac{1}{|\mathcal{S}_a| + |\mathcal{S}_b|}\left(\sum_{i \in \mathcal{S}_a} d(i, \mathcal{S}_b) + \sum_{i \in \mathcal{S}_b} d(i, \mathcal{S}_a)\right), \tag{13}$$

where $\mathcal{S}_a$ and $\mathcal{S}_b$ represent the set of surface points of the target segmented by the algorithm and the ground truth, respectively. $d(i, \mathcal{S}_b)$ is the shortest euclidean distance between i and $\mathcal{S}_b$. We used the Student's t-test to compute the p-value in order to see whether the results of two algorithms significantly differ from each other.

4.2 2D Placenta Segmentation from Fetal MRI

4.2.1 Clinical Background and Experiments Setting

Fetal MRI is an emerging diagnostic tool complementary to ultrasound due to its large field of view and good soft tissue contrast. Segmenting the placenta from fetal MRI is important for fetal surgical planning such as in the case of twin-to-twin transfusion syndrome [61]. Clinical fetal MRI data are often acquired with a large slice thickness for good contrast-to-noise ratio. Movement of the fetus can lead to inhomogeneous appearances between slices. In addition, the location and orientation of the placenta vary largely between individuals. These factors make automatic and 3D segmentation of the placenta a challenging task [62]. Interactive 2D slice-based segmentation is expected to achieve more robust results [6], [55]. The 2D segmentation results can also be used for motion correction and high-resolution volume reconstruction [63].

We collected clinical T2-weighted MRI scans of 25 pregnant women in the second trimester with Single-shot Fast Spin-echo (SSFSE). The data were acquired in axial view with pixel size between 0.7422 mm $\times$ 0.7422 mm and 1.582 mm $\times$ 1.582 mm and slice thickness 3-4 mm. Each slice was resampled with a uniform pixel size of 1 mm $\times$ 1 mm

TABLE 1
Quantitative Comparison of Different Networks and CRFs in Stage 1 of 2D Placenta Segmentation

Method	Dice(%)	ASSD(pixels)
FCN [11]	81.47 ± 11.40	2.66 ± 1.39
DeepLab [58]	83.38 ± 9.53	2.20 ± 0.84
2D P-Net(b5)	83.16 ± 13.01	2.36 ± 1.66
2D P-Net	84.78 ± 11.74	2.09 ± 1.53
2D P-Net + Dense CRF	84.90 ± 12.05	2.05 ± 1.59
2D P-Net + CRF-Net(g)	**85.44 ± 12.50**	**1.98 ± 1.46**
2D P-Net + CRF-Net(f)	**85.86 ± 11.67**	**1.85 ± 1.30**

CRF-Net(g) [50] constrains pairwise potential as Gaussian functions. CRF-Net(f) is our proposed CRF that learns freeform pairwise potential functions. Significant improvement from 2D P-Net (p-value < 0.05) is shown in bold font.

and cropped by a box of size 172 × 128 containing the placenta. We used 17 volumes with 624 slices for training, three volumes with 122 slices for validation and five volumes with 179 slices for testing. The ground truth was manually delineated by an experienced Radiologist.

4.2.2 Stage 1: Automatic Segmentation by 2D P-Net with CRF-Net(f)

Fig. 6 shows the automatic segmentation results obtained by different networks in Stage 1. It shows that FCN is able to capture the main region of the placenta. However, the segmentation results are blob-like with smooth boundaries. DeepLab is better than FCN, but its blob-like results are similar to those of FCN. This is mainly due to the downsampling and upsampling procedure employed by these methods. In contrast, 2D P-Net(b5) and 2D P-Net obtain more detailed results. It can be observed that 2D P-Net achieves better results than the other three networks. However, there are still some obvious mis-segmented regions by 2D P-Net. Table 1 presents a quantitative comparison of these networks based on all the testing data. 2D P-Net achieves a Dice score of 84.78 ± 11.74 percent and an ASSD of 2.09 ± 1.53 pixels, and it performs better than the other three networks.

Based on 2D P-Net, we compared different CRFs in Stage 1. A visual comparison between Dense CRF, CRF-Net(g) with Gaussian pairwise potentials and CRF-Net(f) with freeform pairwise potentials is shown in Fig. 7. In the first column, the placenta is under-segmented by 2D P-Net. Dense CRF leads to very small improvements on the result. CRF-Net(g) and CRF-Net(f) improve the result by preserving more placenta regions, and the later shows a better segmentation. In the second column, 2D P-Net obtains an over-segmentation of adjacent fetal brain and maternal tissues. Dense CRF does not improve the segmentation noticeably, but CRF-Net(g) and CRF-Net(f) remove more over-segmentated areas. CRF-Net(f) shows a better performance than the other two CRFs. The quantitative evaluation of these three CRFs is presented in Table 1, which shows Dense CRF leads to a result that is very close to that of 2D P-Net (p-value > 0.05), while the last two CRFs significantly improve the segmentation (p-value $<$

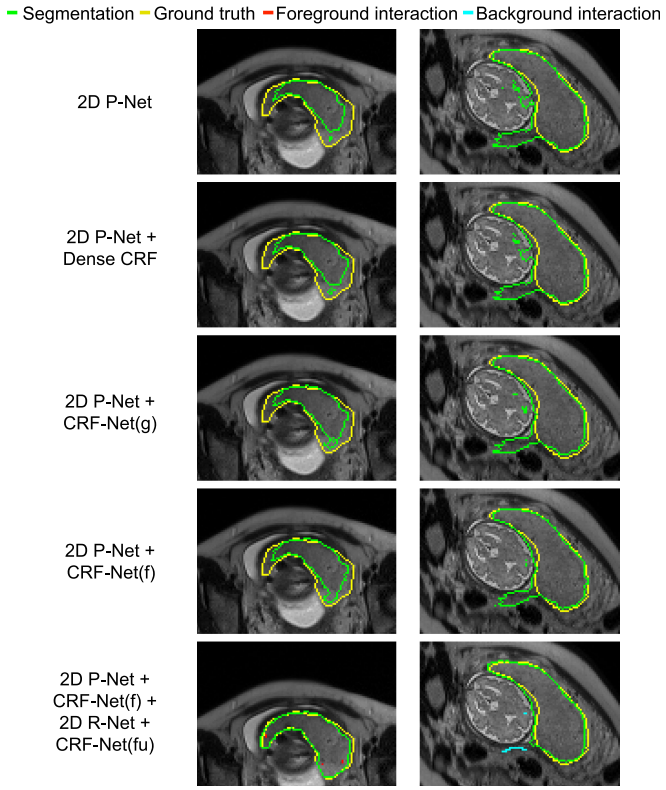


Fig. 7. Visual comparison of different CRFs in Stage 1 of 2D placenta segmentation. The last row shows interactively refined results by DeepIGeoS.

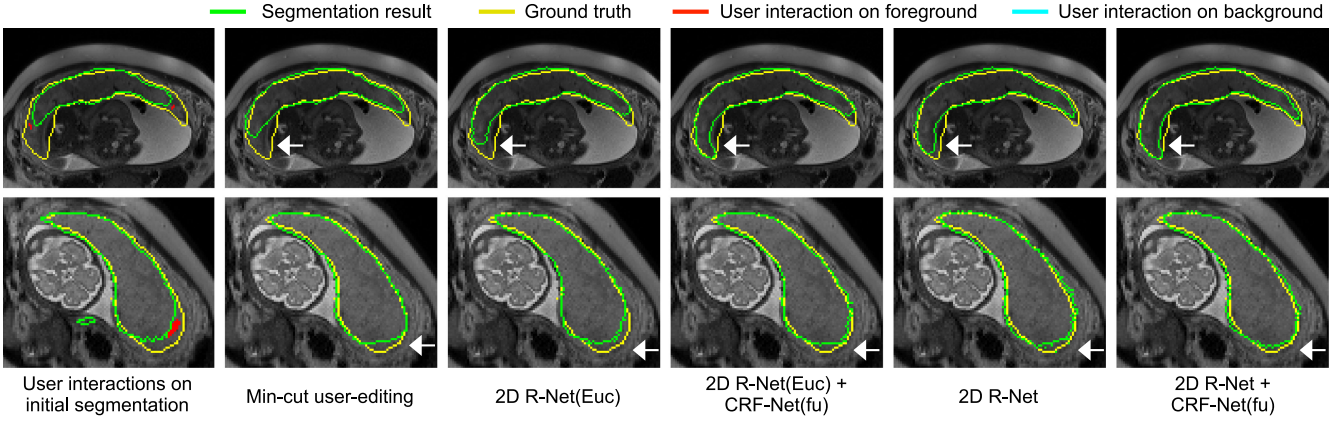


Fig. 8. Visual comparison of different refinement methods in Stage 2 of 2D placenta segmentation. The first column shows the initial automatic segmentation obtained by 2D P-Net + CRF-Net(f), on which user interactions are added for refinement. The remaining columns show refined results. 2D R-Net(Euc) is a counterpart of the proposed 2D R-Net and it uses euclidean distance. White arrows show the difference in local details.

TABLE 2
Quantitative Evaluation of Different Refinement Methods in Stage 2 of 2D Placenta Segmentation

Method	Dice(%)	ASSD(pixels)
Before refinement	85.86 ± 11.67	1.85 ± 1.30
Min-cut user-editing	87.04 ± 9.79	1.63 ± 1.15
2D R-Net(Euc)	88.26 ± 10.61	1.54 ± 1.18
2D R-Net	88.76 ± 5.56	1.31 ± 0.60
2D R-Net(Euc) + CRF-Net(fu)	88.71 ± 8.42	1.26 ± 0.59
2D R-Net + CRF-Net(fu)	**89.31 ± 5.33**	**1.22 ± 0.55**

The initial segmentation is obtained by 2D P-Net + CRF-Net(f). 2D R-Net (Euc) uses euclidean distance instead of geodesic distance. Significant improvement from 2D R-Net (p-value < 0.05) is shown in bold font.

0.05). In addition, CRF-Net(f) is better than CRF-Net(g). Fig. 7 and Table 1 indicate that large mis-segmentation exists in some images, therefore we use 2D R-Net with CRF-Net(fu) to refine the segmentation interactively in the following section.

4.2.3 Stage 2: Interactive Refinement by 2D R-Net with CRF-Net(fu)

Fig. 8 shows examples of interactive refinement based on 2D R-Net with CRF-Net(fu) in Stage 2. The first column in Fig. 8 shows initial segmentation results obtained by 2D P-Net + CRF-Net(f). The user provides clicks/scribbles to indicate the foreground (red) or the background (cyan). The second to last column in Fig. 8 show the results for five variations of refinement. These refinement methods correct most of the mis-segmented areas but perform at different levels in dealing with local details, as indicated by white arrows. Fig. 8 shows 2D R-Net with geodesic distance performs better than min-cut user-editing and 2D R-Net(Euc) that uses euclidean distance. CRF-Net(fu) can further improve the segmentation. For quantitative comparison, we measured the segmentation accuracy after the first iteration of user refinement (giving user interactions to mark all the main mis-segmented regions and applying refinement once), in which the same initial segmentation and the same set of user interactions were used by the five refinement methods. The results are presented in Table 2, which shows the combination of the proposed 2D R-Net using geodesic distance and CRF-Net (fu) leads to more accurate segmentations than the other refinement methods with the same set of user interactions.

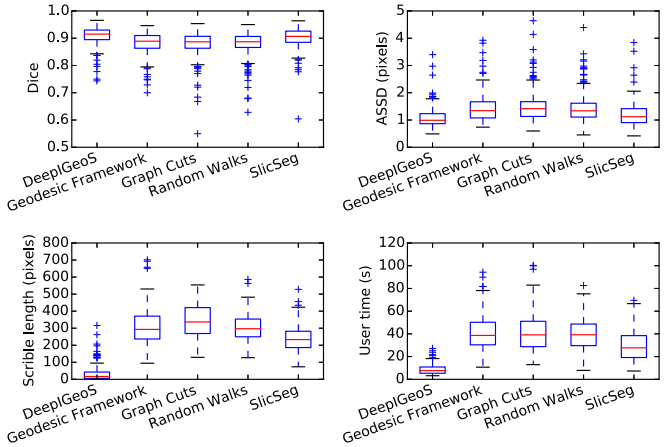


Fig. 10. Quantitative comparison of 2D placenta segmentation by different interactive methods in terms of Dice, ASSD, total interactions (scribble length) and user time.

The Dice score and ASSD of 2D R-Net + CRF-Net(fu) are 89.31 ± 5.33 percent and 1.22 ± 0.55 pixels, respectively.

4.2.4 Comparison with Other 2D Interactive Methods

Fig. 9 shows a visual comparison between DeepIGeoS and Geodesic Framework [35], Graph Cuts [8], Random Walks [7] and SlicSeg [6] for 2D placenta segmentation. The first row shows the initial scribbles and the resulting segmentation. Notice no initial scribbles are needed for DeepIGeoS. The second row shows refined results, where DeepIGeoS only needs two short strokes to get an accurate segmentation, while the other methods require far more scribbles to get similar results. Quantitative comparison of these methods based on the final segmentation given by the two users is presented in Fig. 10. It shows these methods achieve similar accuracy, but DeepIGeoS requires far fewer user interactions and less user time. (See supplementary video 1, available online.)

4.3 3D Brain Tumor Segmentation from FLAIR Images

4.3.1 Clinical Background and Experiments Setting

Gliomas are the most common brain tumors in adults with little improvement in treatment effectiveness despite

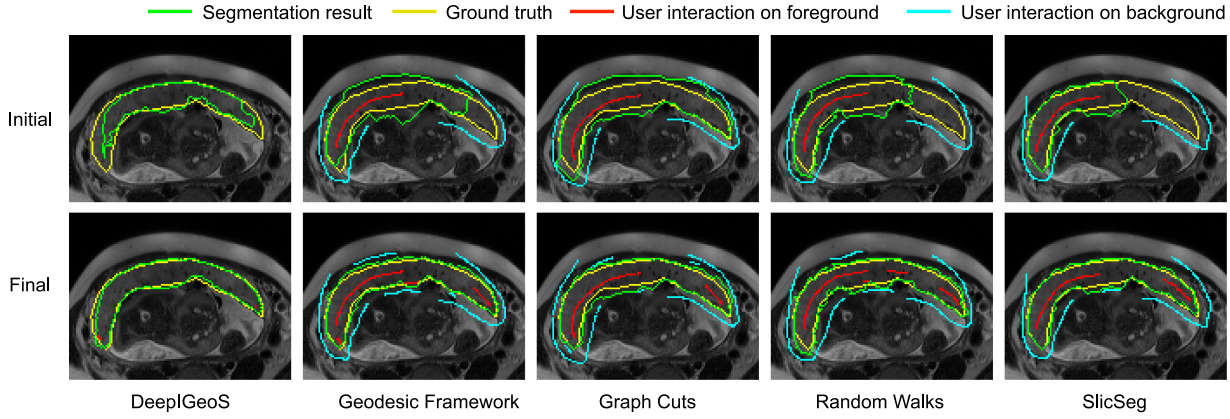


Fig. 9. Visual comparison of DeepIGeoS and other interactive methods for 2D placenta segmentation. The first row shows initial scribbles (except for DeepIGeoS) and the resulting segmentation. The second row shows final refined results with the entire set of scribbles. The user decided on the level of interaction required to achieve a visually acceptable result.

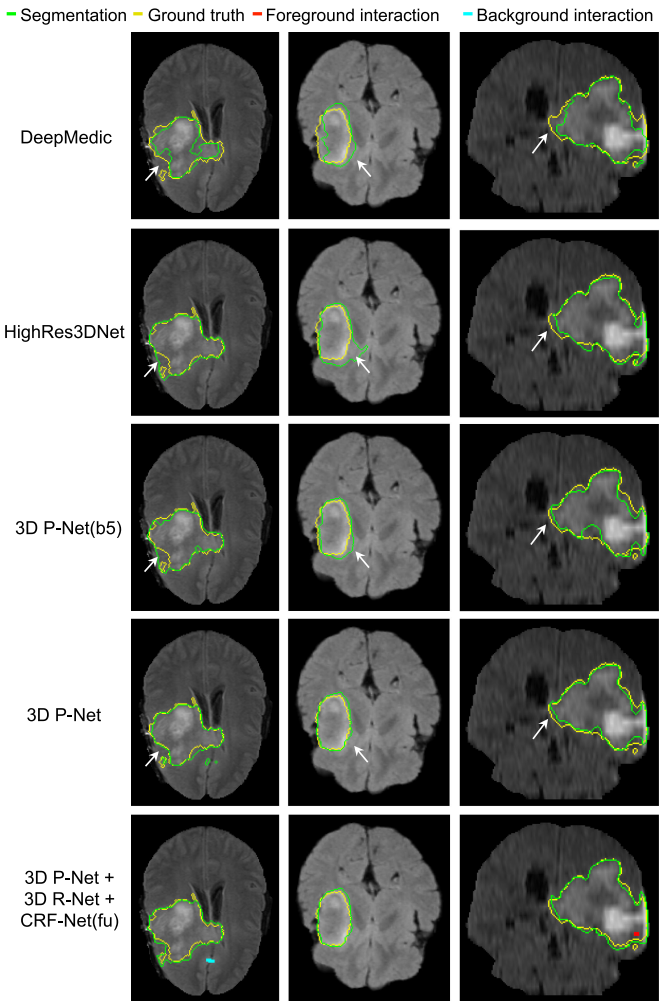


Fig. 11. Visual comparison of different networks in Stage 1 of 3D brain tumor segmentation. The last row shows interactively refined results by DeepIGeoS.

TABLE 3
Quantitative Comparison of Different Networks and CRFs in Stage 1 of 3D Brain Tumor Segmentation

Method	Dice (%)	ASSD (pixels)
DeepMedic [15]	83.87 ± 8.72	2.38 ± 1.52
HighRes3DNet [14]	85.47 ± 8.66	2.20 ± 2.24
3D P-Net(b5)	85.36 ± 7.34	2.21 ± 2.13
3D P-Net	86.68 ± 7.67	2.14 ± 2.17
3D P-Net + Dense CRF	87.06 ± 7.23	2.10 ± 2.02
3D P-Net + CRF-Net(f)	$\mathbf{87.55 \pm 6.72}$	$\mathbf{2.04 \pm 1.70}$

Significant improvement from 3D P-Net (p-value < 0.05) is shown in bold font.

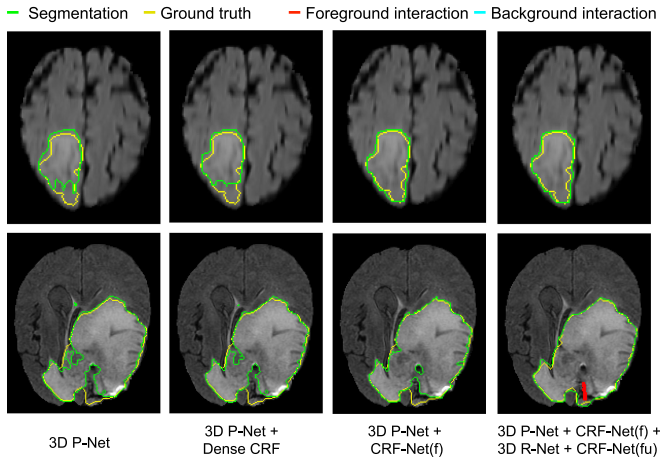


Fig. 12. Visual comparison of different CRFs in Stage 1 of 3D brain tumor segmentation. The last column shows interactively refined results by DeepIGeoS.

considerable research works [64]. With the development of medical imaging, brain tumors can be imaged by different MR protocols with different contrasts. For example, T1-weighted images highlight enhancing part of the tumor and FLAIR acquisitions highlight the peritumoral edema. Segmentation of brain tumors can provide better volumetric measurements and therefore has enormous potential value for improved diagnosis, treatment planning, and follow-up of individual patients. However, automatic brain tumor segmentation remains technically challenging because 1) the size, shape, and localization of brain tumors have considerable variations among patients; 2) the boundaries between adjacent structures are often ambiguous.

In this experiment, we investigate interactive segmentation of the whole tumor from FLAIR images. We used the 2015 Brain Tumor Segmentation Challenge (BraTS) [64] training set with images of 274 cases. The ground truth were manually delineated by several experts. Differently from previous works using this dataset for multi-label and multi-modality segmentation [15], [65], as a first demonstration of deep interactive segmentation in 3D, we only use FLAIR images in the dataset and only segment the whole tumor. We randomly selected 234 cases for training and used the remaining 40 cases for testing. All these images had been skull-stripped and resampled to size of $240 \times 240 \times 155$ with isotropic resolution 1 mm^3. We cropped each image based on the bounding box of its non-zero region. The feature channel number of 3D P-Net and R-Net was $C = 16$.

4.3.2 Stage 1: Automatic Segmentation by 3D P-Net with CRF-Net(f)

Fig. 11 shows examples of automatic segmentation by different networks in Stage 1, where 3D P-Net is compared with DeepMedic [15], HighRes3DNet [14] and 3D P-Net (b5). In the first column, DeepMedic segments the tumor roughly, with some missed regions near the boundary. HighRes3DNet reduces the missed regions but leads to some over-segmentation. 3D P-Net(b5) obtains a similar result to that of HighRes3DNet. In contrast, 3D P-Net achieves a more accurate segmentation, which is closer to the ground truth. More examples in the second and third column in Fig. 11 also show 3D P-Net outperforms the other networks. Quantitative evaluation of these four networks is presented in Table 3. DeepMedic achieves an average dice score of 83.87 percent. HighRes3DNet and 3D P-Net(b5) achieve similar performance, and they are better than DeepMedic. 3D P-Net outperforms these three counterparts with 86.68 ± 7.67 percent in terms of Dice and 2.14 ± 2.17 pixels in terms of ASSD. Note that the proposed 3D P-Net has far fewer parameters compared with HighRes3DNet. It is more memory efficient and therefore can perform inference on a 3D volume in interactive time.

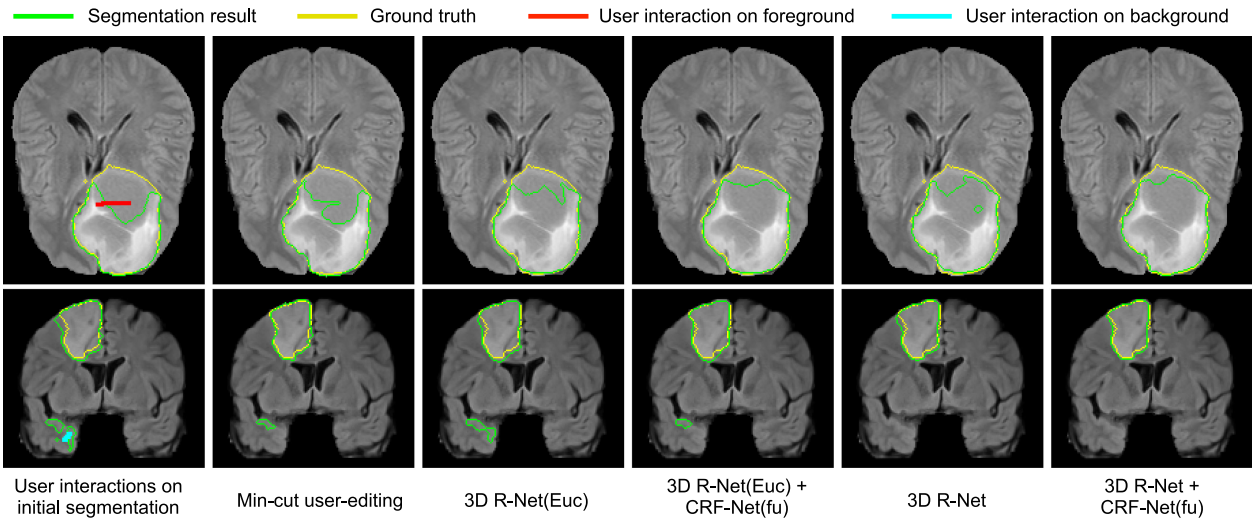


Fig. 13. Visual comparison of different refinement methods in Stage 2 of 3D brain tumor segmentation. The initial segmentation is obtained by 3D P-Net + CRF-Net(f), on which user interactions are given. 3D R-Net(Euc) is a counterpart of the proposed 3D R-Net and it uses euclidean distance.

Since CRF-RNN [50] was only implemented for 2D, in the context of 3D segmentation we only compared 3D CRF-Net(f) with 3D Dense CRF [15] that uses manually tuned parameters. Visual comparison between these two types of CRFs working with 3D P-Net in Stage 1 is shown in Fig. 12. It can be observed that CRF-Net(f) achieves more noticeable improvement compared with Dense CRF that is used as post-processing without end-to-end learning. Quantitative measurement of Dense CRF and CRF-Net(f) is listed in Table 3. It shows that only CRF-Net(f) obtains significantly better segmentation than 3D P-Net with p-value < 0.05.

4.3.3 Stage 2: Interactive Refinement by 3D R-Net with CRF-Net(fu)

Fig. 13 shows examples of interactive refinement results in Stage 2 of 3D brain tumor segmentation. The initial segmentation is obtained by 3D P-Net + CRF-Net(f) in Stage 1. With the same set of user interactions, we compared the refined results of min-cut user-editing and four variations of 3D R-Net: using geodesic or euclidean distance transforms with or without CRF-Net(fu). Fig. 13 shows that min-cut user-editing achieves a small improvement. It can be found that more accurate results are obtained by using geodesic distance than using euclidean distance, and CRF-Net(fu) can further help to improve the segmentation. For quantitative comparison, we measured the segmentation accuracy after the first iteration of refinement, in which the same set of scribbles were used for different refinement methods. The quantitative evaluation is listed in Table 4, showing that the proposed 3D R-Net with geodesic distance and CRF-Net(fu) achieves higher accuracy than the other variations with a Dice score of 89.93 $\pm$ 6.49 percent and ASSD of 1.43 $\pm$ 1.16 pixels.

TABLE 4
Quantitative Comparison of Different Refinement Methods in Stage 2 of 3D Brain Tumor Segmentation

Method	Dice(%)	ASSD(pixels)
Before refinement	87.55 $\pm$ 6.72	2.04 $\pm$ 1.70
Min-cut user-editing	88.41 $\pm$ 7.05	1.74 $\pm$ 1.53
3D R-Net(Euc)	88.82 $\pm$ 7.68	1.60 $\pm$ 1.56
3D R-Net	89.30 $\pm$ 6.82	1.52 $\pm$ 1.37
3D R-Net(Euc) + CRF-Net(fu)	89.27 $\pm$ 7.32	1.48 $\pm$ 1.22
3D R-Net + CRF-Net(fu)	**89.93 $\pm$ 6.49**	**1.43 $\pm$ 1.16**

The segmentation before refinement is obtained by 3D P-Net + CRF-Net(f). 3D R-Net(Euc) uses euclidean distance instead of geodesic distance. Significant improvement from 3D R-Net (p-value < 0.05) is shown in bold font.

4.3.4 Comparison with Other 3D Interactive Methods

Fig. 14 shows a visual comparison between GeoS [36], ITK-SNAP [5] and DeepIGeoS. In the first row, the tumor has a good contrast with the background. All the compared methods achieve very accurate segmentations. In the second row, a lower contrast makes it difficult for the user to identify the tumor boundary. Benefited from the initial tumor boundary that is automatically identified by 3D P-Net, DeepIGeoS outperforms GeoS and ITK-SNAP. Quantitative comparison is presented in Fig. 15. It shows DeepIGeoS achieves higher accuracy compared with GeoS and ITK-SNAP. In addition, the user time for DeepIGeoS is about one third of that for the other two methods. Supplementary video 2, available

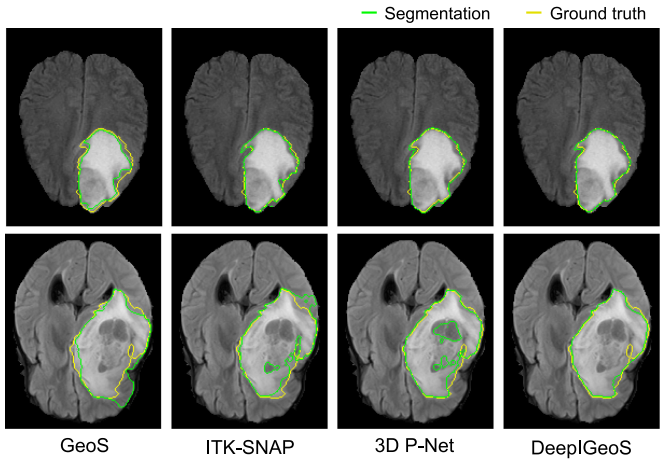


Fig. 14. Visual comparison of 3D brain tumor segmentation using GeoS, ITK-SNAP, and DeepIGeoS that is based on 3D P-Net.

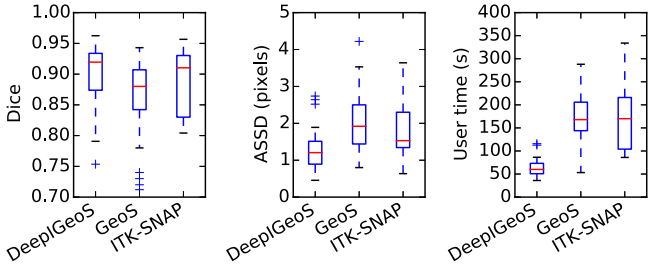


Fig. 15. Quantitative evaluation of 3D brain tumor segmentation by DeepIGeoS, GeoS and ITK-SNAP.

online, shows more examples of DeepIGeoS for 3D brain tumor segmentation.

5 Conclusion

In this work, we presented a deep learning-based interactive framework for 2D and 3D medical image segmentation. We proposed a two-stage framework with a P-Net to obtain an initial automatic segmentation and an R-Net to refine the result based on user interactions that are transformed into geodesic distance maps and then integrated into the input of R-Net. We also proposed a resolution-preserving network structure with dilated convolution for dense prediction, and extended the existing RNN-based CRF so that it can learn freeform pairwise potentials and take advantage of user interactions as hard constraints. Segmentation results of the placenta from 2D fetal MRI and brain tumors from 3D FLAIR images show that our proposed method achieves better results than automatic CNNs. It requires far less user time compared with traditional interactive methods and achieves higher accuracy for 3D brain tumor segmentation. The framework can be extended to deal with multiple organs in the future.

Acknowledgments

This work was supported through an Innovative Engineering for Health award by the Wellcome Trust (WT101957); Engineering and Physical Sciences Research Council (EPSRC) (NS/A000027/1, EP/H046410/1, EP/J020990/1, EP/K005278), Wellcome/EPSRC [203145Z/16/Z], the National Institute for Health Research University College London Hospitals Biomedical Research Centre (NIHR BRC UCLH/UCL), the Royal Society [RG160569], a UCL Overseas Research Scholarship and a UCL Graduate Research Scholarship, hardware donated by NVIDIA and of Emerald, a GPU-accelerated High Performance Computer, made available by the Science & Engineering South Consortium operated in partnership with the STFC Rutherford-Appleton Laboratory.

References

[1] D. Withey and Z. Koles, "Medical image segmentation: Methods and software," in *Proc. Joint Meeting 6th Int. Symp. Noninvasive Functional Source Imaging Brain Heart Int. Conf. Functional Biomedical Imaging*, 2007, pp. 140–143.

[2] N. Sharma and L. M. Aggarwal, "Automated medical image segmentation techniques," *J. Medical Physics*, vol. 35, no. 1, pp. 3–14, 2010.

[3] F. Zhao and X. Xie, "An overview of interactive medical image segmentation," *Ann. BMVA*, vol. 2013, no. 7, pp. 1–22, 2013.

[4] A. Hamamci, N. Kucuk, K. Karaman, K. Engin, and G. Unal, "Tumor-Cut: Segmentation of brain tumors on contrast enhanced MR images for radiosurgery applications," *IEEE Trans. Medical Imaging*, vol. 31, no. 3, pp. 790–804, Mar. 2012.

[5] P. A. Yushkevich, J. Piven, H. C. Hazlett, R. G. 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.

[6] G. Wang, M. A. Zuluaga, R. Pratt, M. Aertsen, T. Doel, M. Klusmann, A. L. David, J. Deprest, T. Vercauteren, and S. Ourselin, "Slic-Seg: A minimally interactive segmentation of the placenta from sparse and motion-corrupted fetal MRI in multiple views," *Med. Image Anal.*, vol. 34, pp. 137–147, 2016.

[7] L. Grady, "Random walks for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 11, pp. 1768–1783, Nov. 2006.

[8] Y. Y. Boykov and M. P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images," in *Proc. 8th IEEE Int. Conf. Comput. Vis.*, 2001, pp. 105–112.

[9] C. Rother, V. Kolmogorov, and A. Blake, "GrabCut: Interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 309–314, 2004.

[10] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. van der Laak, B. V. Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis," *Med. Image Anal.*, vol. 42, pp. 60–88, 2017.

[11] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3431–3440.

[12] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Medical Image Comput. Comput.-Assisted Intervention*, 2015, pp. 234–241.

[13] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: Fully convolutional neural networks for volumetric medical image segmentation," in *Proc. 4th Int. Conf. 3D Vis.*, 2016, pp. 565–571.

[14] W. Li, G. Wang, L. Fidon, S. Ourselin, M. J. Cardoso, and T. Vercauteren, "On the compactness, efficiency, and representation of 3D convolutional networks: brain parcellation as a pretext task," in *Proc. Inf. Process. Medical Imaging*, 2017, pp. 348–360.

[15] K. Kamnitsas, C. Ledig, V. F. J. Newcombe, J. P. Simpson, A. D. Kane, D. K. Menon, D. Rueckert, and B. Glocker, "Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation," *Med. Image Anal.*, vol. 36, pp. 61–78, 2017.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. 25th Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

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

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[19] M. Havaei, A. Davy, D. Warde-Farley, A. Biard, A. Courville, Y. Bengio, C. Pal, P.-M. Jodoin, and H. Larochelle, "Brain tumor segmentation with deep neural networks," *Med. Image Anal.*, vol. 35, pp. 18–31, 2016.

[20] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 580–587.

[21] A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3D U-Net: Learning dense volumetric segmentation from sparse annotation," in *Proc. Medical Image Comput. Comput.-Assisted Intervention*, 2016, pp. 424–432.

[22] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," *arXiv preprint arXiv:1412.7062*.

[23] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*.

[24] P. Ondruska, J. Dequaire, D. Z. Wang, and I. Posner, "End-to-end tracking and semantic segmentation using recurrent neural networks," *arXiv preprint arXiv:1604.05091*.

[25] J. Dai, K. He, Y. Li, S. Ren, and J. Sun, "Instance-sensitive fully convolutional networks," in *Proc. 14th Eur. Conf. Comput. Vis.*, 2016, pp. 534–549.

[26] C. Lea, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks for action segmentation and detection," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 47–54.

[27] G. Lin, C. Shen, I. Reid, and A. van dan Hengel, "Efficient piecewise training of deep structured models for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3194–3203.

[28] P. Pinheiro and R. Collobert, "Recurrent convolutional neural networks for scene labeling," in *Proc. 31st Int. Conf. Int. Conf. Mach. Learn.*, 2014, pp. 82–90.

[29] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Hypercolumns for object segmentation and fine-grained localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 447–456.

[30] C. J. Armstrong, B. L. Price, and W. A. Barrett, "Interactive segmentation of image volumes with live surface," *Comput. Graph.*, vol. 31, no. 2, pp. 212–229, 2007.

[31] J. E. Cates, A. E. Lefohn, and R. T. Whitaker, "GIST: An interactive, GPU-based level set segmentation tool for 3D medical images," *Med. Image Anal.*, vol. 8, no. 3, pp. 217–231, 2004.

[32] M. Rajchl, M. Lee, O. Oktay, K. Kamnitsas, J. Passerat-Palmbach, W. Bai, M. Rutherford, J. Hajnal, B. Kainz, and D. Rueckert, "DeepCut: Object segmentation from bounding box annotations using convolutional neural networks," *IEEE Trans. Medical Imaging*, vol. 36, no. 2, pp. 674–683, Feb. 2017.

[33] S. A. Haider, M. J. Shafiee, A. Chung, F. Khalvati, A. Oikonomou, A. Wong, and M. A. Haider, "Single-click, semi-automatic lung nodule contouring using hierarchical conditional random fields," in *Proc. IEEE 12th Int. Symp. Biomed. Imaging*, 2015, pp. 1139–1142.

[34] C. Xu and J. L. Prince, "Snakes, shapes, and gradient vector flow," *IEEE Trans. Image Process.*, vol. 7, no. 3, pp. 359–369, Mar. 1998.

[35] X. Bai and G. Sapiro, "A geodesic framework for fast interactive image and video segmentation and matting," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, 2007, pp. 1–8.

[36] A. Criminisi, T. Sharp, and A. Blake, "GeoS: Geodesic image segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 99–112.

[37] O. Barinova, R. Shapovalov, S. Sudakov, and A. Velizhev, "Online random forest for interactive image segmentation," in *Proc. Int. Workshop Exp. Economics Mach. Learn.*, 2012, pp. 1–8.

[38] I. Luengo, M. C. Darrow, M. C. Spink, Y. Sun, W. Dai, C. Y. He, W. Chiu, T. Pridmore, A. W. Ashton, E. M. Duke, M. Basham, and A. P. French, "SuRVoS: Super-region volume segmentation workbench," *J. Structural Biol.*, vol. 198, no. 1, pp. 43–53, 2017.

[39] D. Lin, J. Dai, J. Jia, K. He, and J. Sun, "ScribbleSup: Scribble-supervised convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3159–3167.

[40] N. Xu, B. Price, S. Cohen, J. Yang, and T. Huang, "Deep interactive object selection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 373–381.

[41] P. Kontschieder, P. Kohli, J. Shotton, and A. Criminisi, "GeoF geodesic forests for learning coupled predictors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 65–72.

[42] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1124–1137, Sep. 2004.

[43] J. Yuan, E. Bae, and X. C. Tai, "A study on continuous max-flow and min-cut approaches," in *Proc. IEEE Comput. Society Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 2217–2224.

[44] N. Payet and S. Todorovic, "(RF)2 – random forest random field," in *Proc. Neural Inf. Process. Syst. Conf.*, 2010, pp. 1885–1893.

[45] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected CRFs with gaussian edge potentials," in *Proc. Neural Inf. Process. Syst. Conf.*, 2011, pp. 109–117.

[46] M. Szummer, P. Kohli, and D. Hoiem, "Learning CRFs using graph cuts," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 582–595.

[47] J. I. Orlando and M. Blaschko, "Learning fully-connected CRFs for blood bessel segmentation in retinal images," in *Proc. Medical Image Comput. Comput.-Assisted Intervention*, 2014, pp. 634–641.

[48] J. Domke, "Learning graphical model parameters with approximate marginal inference," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 10, pp. 2454–2467, Oct. 2013.

[49] P. Krähenbühl and V. Koltun, "Parameter learning and convergent inference for dense random fields," in *Proc. 30th Int. Conf. Int. Conf. Mach. Learn.*, 2013, pp. 513–521.

[50] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr, "Conditional random fields as recurrent neural networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1529–1537.

[51] A. Adams, J. Baek, and M. A. Davis, "Fast high-dimensional filtering using the permutohedral lattice," *Comput. Graph. Forum*, vol. 29, no. 2, pp. 753–762, 2010.

[52] R. Vemulapalli, O. Tuzel, M.-y. Liu, and R. Chellappa, "Gaussian conditional random field network for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3224–3233.

[53] F. Liu, C. Shen, and G. Lin, "Deep convolutional neural fields for depth estimation from a single image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 5162–5170.

[54] A. Kirillov, S. Zheng, D. Schlesinger, W. Forkel, A. Zelenin, P. Torr, and C. Rother, "Efficient likelihood learning of a generic CNN-CRF model for semantic segmentation," *arXiv preprint arXiv:1511.05067*.

[55] G. Wang, M. A. Zuluaga, R. Pratt, M. Aertsen, T. Doel, M. Klusmann, A. L. David, J. Deprest, T. Vercauteren, and S. Ourselin, "Dynamically balanced online random forests for interactive scribble-based segmentation," in *Proc. Medical Image Comput. Comput.-Assisted Intervention*, 2016, pp. 352–360.

[56] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts?" *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 2, pp. 147–159, Feb. 2004.

[57] D. Han, J. Bayouth, Q. Song, A. Taurani, M. Sonka, J. Buatti, and X. Wu, "Globally optimal tumor segmentation in PET-CT images: A graph-based co-segmentation method," in *Proc. Inf. Process. Medical Imaging*, 2011, pp. 245–256.

[58] 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 Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–8481, Apr. 2018.

[59] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. 22nd ACM Int. Conf. Multimedia*, 2014, pp. 675–678.

[60] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, X. Zheng, and G. Brain, "TensorFlow: A system for large-scale machine learning," in *Proc. 12th USENIX Conf. Operating Syst. Des. Implementation*, 2016, pp. 265–284.

[61] J. A. Deprest, A. W. Flake, E. Gratacos, Y. Ville, K. Hecher, K. Nicolaides, M. P. Johnson, F. I. Luks, N. S. Adzick, and M. R. Harrison, "The making of fetal surgery," *Prenatal Diagnosis*, vol. 30, no. 7, pp. 653–667, 2010.

[62] A. Alansary, K. Kamnitsas, A. Davidson, M. Rajchl, C. Malamateniou, M. Rutherford, J. V. Hajnal, B. Glocker, D. Rueckert, and B. Kainz, "Fast fully automatic segmentation of the human placenta from motion corrupted MRI," in *Proc. Medical Image Comput. Comput.-Assisted Intervention*, 2016, pp. 589–597.

[63] K. Keraudren, M. Kuklisova-Murgasova, V. Kyriakopoulou, C. Malamateniou, M. A. Rutherford, B. Kainz, J. V. Hajnal, and D. Rueckert, "Automated fetal brain segmentation from 2D MRI slices for motion correction," *NeuroImage*, vol. 101, pp. 633–643, 2014.

[64] B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, J. Kirby, Y. Burren, N. Porz, J. Slotboom, R. Wiest, L. Lanczi, E. Gerstner, M. A. Weber, T. Arbel, B. B. Avants, N. Ayache, P. Buendia, D. L. Collins, N. Cordier, J. J. Corso, A. Criminisi, T. Das, H. Delingette, Ç. Demiralp, C. R. Durst, M. Dojat, S. Doyle, J. Festa, F. Forbes, E. Geremia, B. Glocker, P. Golland, X. Guo, A. Hamamci, K. M. Iftekharuddin, R. Jena, N. M. John, E. Konukoglu, D. Lashkari, J. A. Mariz, R. Meier, S. Pereira, D. Precup, S. J. Price, T. R. Raviv, S. M. Reza, M. Ryan, D. Sarikaya, L. Schwartz, H. C. Shin, J. Shotton, C. A. Silva, N. Sousa, N. K. Subbanna, G. Szekely, T. J. Taylor, O. M. Thomas, N. J. Tustison, G. Unal, F. Vasseur, M. Wintermark, D. H. Ye, L. Zhao, B. Zhao, D. Zikic, M. Prastawa, M. Reyes, and K. Van Leemput , "The multimodal brain tumor image segmentation benchmark (BRATS)," *IEEE Trans. Medical Imaging*, vol. 34, no. 10, pp. 1993–2024, Oct. 2015.

[65] L. Fidon, W. Li, L. C. Garcia-Peraza-Herrera, J. Ekanayake, N. Kitchen, S. Ourselin, and T. Vercauteren, "Scalable multimodal convolutional networks for brain tumour segmentation," in *Proc. Medical Image Comput. Comput.-Assisted Intervention*, 2017, pp. 285–293.

Guotai Wang received the bachelor's and master's degrees in biomedical engineering from Shanghai Jiao Tong University, in 2011 and 2014, respectively. He is working toward the PhD degree in Translational Imaging Group, UCL, working on image segmentation for fetal surgical planning. He won the UCL Overseas Research Scholarship and UCL Graduate Research Scholarship. His research interests include image segmentation, computer vision, and deep learning.

Maria A. Zuluaga received the PhD degree from Universit Claude Bernard Lyon. After a year as a postdoctoral fellow with the European Synchrotron Radiation Facility, Grenoble, France, she joined CMIC, in March 2012, as a research associate to work on cardiovascular image analysis and computer-aided diagnosis (CAD) of cardiovascular pathologies. Since August 2014, she has been a part of the Guided Instrumentation for Fetal Therapy and Surgery (GIFT-Surg) project as a senior research associate.

Wenqi Li received the BSc degree in computer science from the University of Science and Technology Beijing, in 2010, the MSc degree in computing with vision and imaging from the University of Dundee, in 2011, and the PhD degree in computer vision and image processing group from the University of Dundee, in 2015. He is a research associate with the Guided Instrumentation for Fetal Therapy and Surgery (GIFT-Surg) project. His main research interests include anatomy detection and segmentation for presurgical evaluation and surgical planning.

Rosalind Pratt received the degree in medicine from the University of Leeds, before starting clinical training in London, where she is undertaking specialist training in obstetrics and gynaecology. She is a clinical academic training fellow with the UCL Institute for Women's Health. Her main research focus is in novel imaging of the human placenta.

Premal A. Patel received the degree in medicine from the St Bartholomews and the Royal London Hospital School of Medicine and Dentistry, University of London. He is a consultant paediatric interventional radiologist with the Great Ormond Street Hospital for Children NHS Foundation Trust and undertakes a variety of image guided procedures in children. He is also a clinical training fellow within the Translational Imaging Group, UCL. Following a period of training in Surgery and Paediatric Surgery in London, he undertook training in Clinical Radiology in Southampton and subsequently undertook Fellowships in Paediatric Interventional Radiology at Great Ormond Street Hospital for Children, London, United Kingdom and The Hospital for Sick Children (SickKids), University of Toronto, Canada.

Michael Aertsen received the medicine degree from the University of Hasselt and the Katholieke Universiteit Leuven. He is a consultant pediatric radiologist with the University Hospitals of Leuven. He is specialized in fetal MRI and his main research focus is the fetal brain development with advanced MRI techniques.

Tom Doel received the MSc degree in applied mathematics from the University of Bath, the MPhys degree in mathematical physics from the University of Edinburgh, and the PhD degree from the University of Oxford developed computational techniques for multi-modal medical image segmentation, registration, analysis and modelling. He is a research associate with the Translational Imaging Group, UCL and an honorary research fellow with the UCL Hospitals NHS Foundation Trust. He is a professional software engineer and worked for Barco Medical Imaging Systems to develop clinical radiology software. His current research is on novel algorithm development and the robust design and architecture of clinically-focussed software for surgical planning and image-guided surgery as part of the GIFT-Surg project.

Anna L. David is a professor and honorary consultant with the Obstetrics and Maternal Fetal Medicine, Institute for Women's Health, University College London (UCL), London. She has a clinical practice with the UCLH in fetal medicine, fetal therapy and obstetrics. Her main research is in translational medicine. She is head of the Research Department of Maternal Fetal Medicine, UCL Institute for Women's Health.

Jan Deprest is a professor of obstetrics and gynaecology with the Katholieke Universiteit Leuven and consultant obstetrician gynaecologist with the University Hospitals Leuven, Belgium. He is currently the academic chair of his Department and the director of the Centre for Surgical Technologies, Faculty of Medicine. He established the Eurofoetus consortium, which is dedicated to the development of instruments and techniques for minimally invasive fetal and placental surgery.

Sébastien Ourselin is director of the EPSRC Centre for Doctoral Training (CDT) in medical imaging, head of the Translational Imaging Group (TIG) as part of the Centre for Medical Image Computing (CMIC), and professor of Medical Image Computing, UCL. His core skills are in medical image analysis, software engineering, and translational medicine. He is best known for his work on image registration and segmentation, its exploitation for robust image-based biomarkers in neurological conditions, as well as for his development of image-guided surgery systems.

Tom Vercauteren received the graduate degree from Ecole Polytechnique, and Columbia University, and the PhD degree from Inria Sophia Antipolis. He is an associate professor with the UCL. His main research focus is on the development of innovative interventional imaging systems and their translation to the clinic. One key driving force of his work is the exploitation of image computing and the knowledge of the physics of acquisition to move beyond the initial limitations of the medical imaging devices that are developed or used in the course of his research.