Neural Computing and Applications (2020) 32:6769–6778
https://doi.org/10.1007/s00521-019-04700-0

S.I. : BRAIN INSPIRED COMPUTING &MACHINE LEARNING APPLIED RESEARCH-BISMLARE

A two-stage approach for automatic liver segmentation with Faster R-CNN and DeepLab

Wei Tang[1] · Dongsheng Zou[1] · Su Yang[1] · Jing Shi[1] · Jingpei Dan[1] · Guowu Song[2]

Received: 4 January 2019 / Accepted: 20 December 2019 / Published online: 16 January 2020

Abstract
Proper liver segmentation is a key step in many clinical applications, including computer-assisted diagnosis, radiation therapy and volume measurement. However, liver segmentation is still challenging due to fuzzy boundary, complex liver anatomy, present of pathologies, and diversified shape. This paper presents a novel two-stage liver detection and segmentation model DSL. The first stage uses improved Faster Regions with CNN features (Faster R-CNN) to detect approximate position of liver. The obtained images are processed and input into DeepLab to obtain the contour of liver. The proposed approach is validated on two datasets MICCAI-Sliver07 and 3Dircadb. Experimental results reveal that the proposed method outperforms the state-of-the-art solutions in terms of volume overlap error, average surface distance, relative volume difference, and total score.

Keywords Liver segmentation · DeepLab · Faster R-CNN

1 Introduction

As the largest substantive organ in human viscera, liver is rich in blood vessels, complex in structure, and has many types of liver diseases and high incidence, which greatly endangers human health. Currently, about 33% of people in the world are suffering from non-alcoholic liver disease [29]. Liver disease has become one of the leading reasons of premature death. The patients need to receive scientific and effective treatment. A common treatment for liver diseases is liver surgery. It needs to segment the liver from CT images, perform processing analysis, obtain information on pathology, physics and anatomy, etc., and provide a theoretical basis for surgical plans. Due to the complex background, blurred boundaries, and different shapes, even manually sketching the liver by experienced experts can be quite time-consuming and subjective. Therefore, the study of automatic liver segmentation is the primary task of liver surgery and has important practical significance.

Many researchers study medical image segmentation to improve the accuracy and effectiveness of diagnosis and treatment. Many tasks, such as liver transplantation and 3D localization in radiotherapy procedures, require automatic liver segmentation. In recent years, semiautomatic or fully automatic liver CT image segmentation methods have been proposed [4, 16], which have achieved certain results, but precise liver segmentation is still quite difficult. Figure 1 shows four reasons. Firstly, the liver intensity is highly similar to that of its neighboring organs (heart and

✉ Dongsheng Zou
dszou@cqu.edu.cn

Wei Tang
WeiTang@cqu.edu.cn

Su Yang
yangsu@cqu.edu.cn

Jing Shi
shijing@cqu.edu.cn

Jingpei Dan
danjingpei@cqu.edu.cn

Guowu Song
guowu.song@changhong.com

[1] College of Computer Science, Chongqing University, No.174 ShaZheng Street, ShaPingBa District, Chongqing, China

[2] Sichuan Changhong Elextronics Co. Ltd., No.35 EastMianyangRoad, High-Tech, Mianyang, Sichuan, China

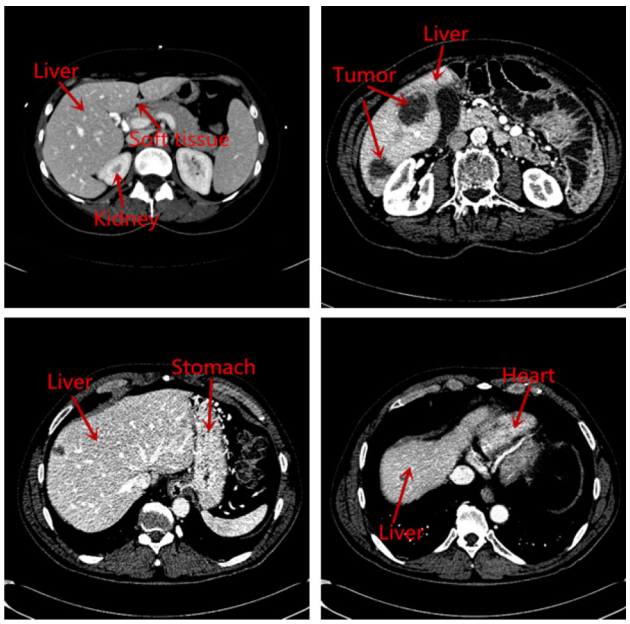

Fig. 1 Four challenges in liver segmentation

stomach). Secondly, partial volume effects blur the liver contour. Thirdly, severe pathological changes, such as cirrhosis and large tumors, often occur in clinical images. However, its intensity is obviously different from normal liver. Fourthly, the liver shape varies from person to person. The above factors make it difficult for existing methods to segment the liver with small size and complex contour and cannot effectively apply the automatic liver segmentation method to clinical diagnosis and treatment.

This paper proposes a two-stage automatic liver segmentation method DSL. It adapts Faster R-CNN algorithm for location of approximate area. Then, we input the obtained images into DeepLab to gain the contour of liver. By first detecting the liver region, the novel method solves the problem that the liver tissue is highly similar to its neighboring organs. This method can effectively segment liver with complex shape and small volume, which has not been found in current research methods. Experiments show that our approach achieves good results in terms of RVD, ASD, VOE, and total score. Therefore, this method has important application value.

2 Literature review

In this section, previous works on liver segmentation are introduced. Considering the degree of human involvement, widely used liver segmentation methods can be simply categorized as interactive segmentation [8, 28], semiautomatic segmentation [8, 19, 34], and automatic segmentation [1, 16, 35].

Interactive liver segmentation methods Manual segmentation relies entirely on experienced experts to artificially delineate liver boundaries based on prior knowledge. Dong et al. [9] raised a random walk method. Wang et al. [32] presented a fast hybrid segmentation method. The key to the success of this method lies in four aspects: linear stretch and anisotropic diffusion data processing, small regions of watershed transform segmentation, region graph constructed using typical Gaussian weighted energy functions.

Semiautomatic segmentation methods They combine computer's fast computing ability with human subjective initiative to complete image-processing tasks. Compared with manual segmentation, it not only reduces manual intervention, but also improves the accuracy and speed. However, it is still tedious. Yang et al. [36] used a combination of two level-set methods. According to the size of liver region, the seed contour point was manually selected and marked. Then, the candidate point close to the seed point was automatically calculated, so as to achieve the goals of adaptive target contour in interaction process. Liao [24] also demonstrated an efficient method based on bottleneck detection and graph cut.

Automatic segmentation methods At present, related works mainly focus on automatic segmentation. Automatic segmentation completely relies on computer to segment independently. It can achieve ideal results without human intervention, and lays a foundation for quantitative measurement and analysis of relevant indicators of medical imaging. Automatic segmentation methods mainly include region growing, rule-based, graphic cutting, statistical shape model and convolutional neural network, etc. Although regional growth is a low-level image processing technique, it has remarkable results in liver segmentation, especially in segmentation rates. Gambino et al. [12] demonstrated a texture-based method that automatically calculates the seed points and thresholds for region growth, and finally the liver organs can be correctly identified and segmented. A set of rules defined with dedicated scripting language was introduced by Schmidt et al. [31], which can be used for object detection. On the basis of BoyKov et al. [2], Burkhardt et al. [3] presented a graph cutting method to automatically select seed points of both liver and background. Subsequently, Li et al. [25] raised a deformable graph cut method. Graph cutting method is especially effective for segmentation of large CT images.

Until recently, DCNNs (deep convolutional neural networks) [17, 27, 35] have revolutionized and dominated detection and segmentation tasks. Dou et al. [10] raised a supervision network so that the model has faster convergence speed and more powerful recognition ability. Lu et al. [27] integrated CNN and graphic cutting method. Yang et al. [35] presented an adversarial network. The

innovation of this method was to utilize the adversarial network to discriminate model output from the ground truth.

Although researchers have proposed various methods of liver segmentation, there are still problems such as inaccurate segmentation, over-segmentation and under-segmentation. The existing methods cannot segment small liver and solve the liver boundary blurring problem well. Therefore, these methods cannot be directly used for clinical diagnosis and treatment. To address these problems, this paper proposes a two-stage liver segmentation model DSL.

3 Method

3.1 Overview

Figure 2 shows processing flow of the proposed DSL method that narrows the scope of semantic segmentation. In the first stage of DSL model, we use improved Faster R-CNN algorithm for approximate location. The obtained image is processed as input of the second stage to segment liver with DeepLab.

In model training, CT images need to be manually annotated with bounding boxes that accurately mark the location of liver. We train Faster R-CNN to utilize the training data with annotation. The pixels in bounding box remain unchanged, and the pixels outside the target frame are set to 0, representing the non-liver region. Subsequently, the processed images are used for DeepLab training. In model testing, we use Faster R-CNN to obtain the liver region of testing images. Finally, DeepLab is utilized to acquire the segmentation results of testing images.

3.2 Liver detection with improved Faster R-CNN

Faster R-CNN [30] merges region proposal network (RPN) and Fast R-CNN [14, 15]. It has evolved into the state-of-the-art framework. Faster R-CNN contains two modules. One is RPN proposing regions, and another is the fast R-CNN detector. This paper uses Faster R-CNN for liver detection. First, we input the whole image into VGG16 for feature extraction. Then, unlike Fast R-CNN using the selective search algorithm, Faster R-CNN adapts RPN for candidate regions generation, and approximately 300 candidate region proposals are generated per image. Next, we map region proposals to feature map of the last convolutional layer of VGG16. The feature maps are not of the same size. Therefore, region proposal needs to be input into the ROI (Region of interest) pooling layer for processing. The characteristics of the unified dimension are finally input to fully connected layer to complete tasks. Faster R-CNN reduces the number of region proposal from the original 2000 to 300. It makes the quality of the region proposal better, and improves the accuracy of detection.

The core of Faster R-CNN is RPN, which greatly reduces the number of regional proposal generated. RPN finds a pixel in the form of sliding window on feature map. The pixel point is at the center of the sliding window of 3×3 convolution kernel. Through the receptive field, we map the pixel to the input image. Then, the pixel is used as the center point to extract nine rectangular regions with three scales (128^2, 256^2, 512^2) and three aspect ratios (1:3, 1:2, 1:1), which are called anchors. It can solve the problem of multi-scale

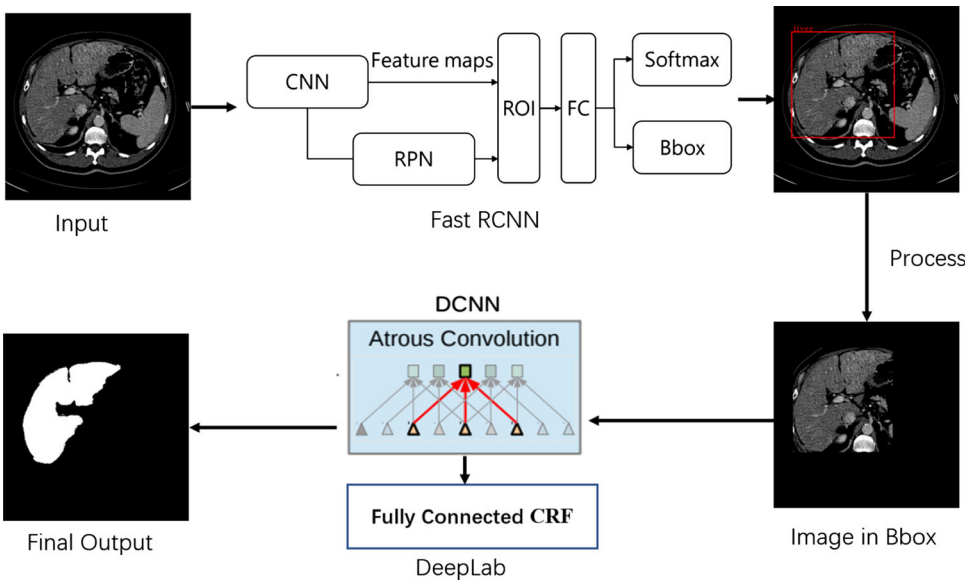

Fig. 2 Overview of the proposed DSL model

objects. On feature map, there are multiple pixels, and one pixel corresponds to nine anchors, which will generate a large number of anchors. Obviously, it is impossible for all anchors to participate in training, but to screen a certain number of samples. The region proposal is generated by inputting the image into the RPN network for a forward calculation to obtain the classification probability and the frame regression parameter of the anchor. On this basis, the coordinates of the region proposal are calculated, and the extracted region proposal subjected to a NMS (non-maximum suppression) yields a high-quality candidate region for final output of the target detection phase.

The loss function is computed as follows [30]:

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{\text{cls}}}\sum_i L_{\text{cls}}(p_i, p_i^*) + \lambda\frac{1}{N_{\text{reg}}}\sum_i p_i^* L_{\text{reg}}(t_i, t_i^*) \quad (1)$$

Here, i is index of an anchor. p_i is the probability of predicting anchor i as liver. p_i^* is the ground-truth label. t_i is a vector representing four parameterized coordinates of predicted box, and t_i^* is that of the ground truth box. $L_{\text{cls}}(p_i, p_i^*)$ and $L_{\text{reg}}(t_i, t_i^*)$ represent classification loss and regression loss, respectively.

$$L_{\text{cls}}(p_i, p_i^*) = -\log[p_i^* p_i + (1-p_i^*)(1-p_i)] \quad (2)$$

$$L_{\text{reg}}(t_i, t_i^*) = R(t_i - t_i^*) \quad (3)$$

where R denotes smooth L1 function. The term $p_i^* L_{\text{reg}}$ means the regression loss is activated only for positive anchors ($p_i^* = 1$) and is disabled otherwise ($p_i^* = 0$). $\{p_i\}$ and $\{t_i\}$ are outputs of classification and regression layer. The two terms are normalized by N_{cls} and N_{reg} weighted by a balancing parameter λ.

N_{cls} is normalized by the minibatch size. N_{reg} is normalized by the number of anchor locations. By default we set $\lambda = 10$, and thus the two terms are equally weighted.

Figure 3 shows the liver region detection results by Faster R-CNN for liver region detection.

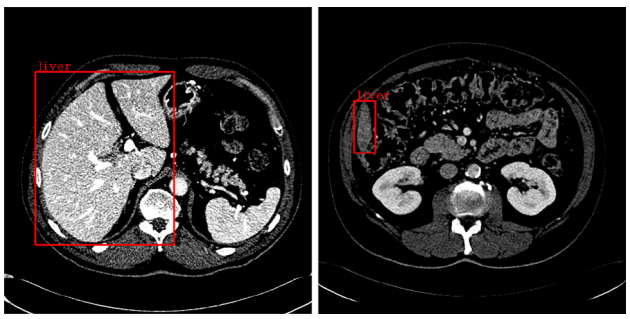

Fig. 3 Detection results by Faster R-CNN

A common problem in many medical image-processing problems is that datasets are unbalanced. When most are negative samples, the learning model is biased toward the negative samples. It can lead to problems where many negative samples are mistakenly divided into positive cases. The method used by Faster R-CNN is to artificially set the proportion of positive and negative samples. In addition, Faster R-CNN adopts a strategy of randomly selecting samples, which makes it possible to select many simple samples, especially many invalid negative samples, resulting in poor expression ability.

To address this problem, we introduce OHEM (Online Hard Example Mining) [38] to improve the Faster R-CNN. The OHEM algorithm first classifies all samples and then puts the misclassified samples into sample set and continues training the classifier, to enhance the discriminant ability of the classifier. With the improved Faster R-CNN algorithm, we first put the liver CT images to the CNN network to calculate the feature map; then generate all possible candidate regions through the RPN network; then we adapt OHEM algorithm to find the hard samples to train network. The search process is to calculate the loss value of all candidate regions through Faster R-CNN, sort according to the loss value, and then utilize NMS algorithm to choose the partial samples with the largest loss value as hard samples. Finally, specific category and location of candidate regions are obtained by classification and regression operations.

In the liver detection stage, the IOU (Intersection Over Union) of candidate region and real box exceeding the threshold of 0.5 is considered as positive sample (foreground); the IOU below the threshold of 0.5 and above 0.1 is considered as negative sample (background). Our improved Faster R-CNN does not need to randomly select 128 candidate regions from the image to form a minibatch, set the ratio of negative and positive samples to 3–1, and then input detection network. Instead, it uses the samples with large loss values for training.

In general, the improved Faster R-CNN algorithm avoids artificially setting the proportion of samples and uses samples with large loss value to train the network effectively. It can reduce the scope of subsequent liver semantic segmentation.

3.3 Liver segmentation with DeepLab

The second stage addresses the task of liver segmentation with DeepLab V2. It is based on DeepLab V1 [5]. DeepLab V2 introduced atrous convolution and ASPP (Atrous Spatial Pyramid Pooling) for object segmentation at multiple scales. It can handle two defects of FCN (Fully Convolutional Networks) [37]. First, FCN does not fully consider the relationship between pixels and pixels in pixel classification. It ignores the spatial regularization steps

commonly used in pixel classification and lacks spatial consistency. Second, the results obtained from FCN are still not effective enough. Although 8× upsampling is much better than 32×, the results of upsampling are still smooth and blurred. More technical details can refer to the original paper [6].

DeepLab V2 removes the last maximum pooling and subsampling layers in DCNN and replaces them with atrous convolution to solve the problem of spatial resolution degradation. Atrous convolution is used to recover feature maps of full resolution. They are significantly reduced by consequent combination of maximum and striding. In addition, atrous convolution does not increase the amount of computation or the number of parameters. The output feature map obtained by atrous convolution is much larger than that of traditional convolution. Yet atrous convolution will lead to high computational cost and consume a large amount of memory resources for processing high-resolution feature maps.

Hence, DeepLab V2 uses bilinear interpolation for some feature mapping and atrous convolution for others. In semantic segmentation, DeepLab V2 uses ASPP to handle scale variability. The feature map finally obtained by DCNN is rich in semantic information, which is very suitable for classification tasks, but it is not applicable to positioning tasks that require a lot of detailed information. It can only locate the approximate position and cannot really delineate borders of objects. To address localization problem, we need an accurate semantic segmentation structure. DeepLab V2 employs a fully connected CRF (Conditional Random Field) [23]. The advantages of fully connected CRF are fast speed, high accuracy, and simple structure. It finds the most likely segmentation by minimizing the energy function

$$E(x) = \sum_i x_i + \sum_{ij} \theta_{ij}(x_i, x_j) \tag{4}$$

where x is the label assignment for pixels. $\theta_i(x_i) = -\log P(x_i)$ is used as unary potential function, where $P(x_i)$ is the label assignment probability computed independently for each pixel by a DCNN. Pairwise potential $\theta_{ij}(x_i, x_j)$ is a linear combination of Gaussian kernels. The following expression is used:

$$\theta_{ij}(x_i, x_j) = \mu(x_i, x_j)\left[w_1 \exp\left(-\frac{\|p_i - p_j\|^2}{2\sigma_\alpha^2} - \frac{\|I_i - I_j\|^2}{2\sigma_\beta^2}\right) + w_2 \exp\left(-\frac{\|p_i - p_j\|^2}{2\sigma_\gamma^2}\right)\right] \tag{5}$$

where $\mu(x_i, x_j) = 1$ if $x_i \neq x_j$, and zero otherwise. The remaining expression combines two Gaussian kernels. The first kernel depends on both pixel positions (denoted as p) and RGB color (denoted as I). It forces pixels with similar color and position to have similar labels. The second kernel only depends on pixel positions. The scale of Gaussian kernels is controlled by hyper parameters σ_α, σ_β and σ_γ.

4 Experiment and analysis

We divide the training procedure of Faster R-CNN into four steps: training RPN, VGG-16, RPN, and VGG-16. We optimize the objective function by the standard SGD (Stochastic Gradient Descent) procedure for 70,000 iterations in training RPN and 50,000 iterations in training VGG-16. Then we fine-tune model weights. Following the procedure of [6], the weights are adopted to the segmentation task. The 1000-way ImageNet classifier in the last layer of VGG-16 is replaced with a two-class (object and background) classifier. The SGD solver is run 100,000 iterations.

Our method is evaluated on two standard benchmarks. The first is the 3Dircadb dataset (source: https://www.ircad.fr/research/3dircadb/). The second is the MICCAI-Sliver07 (source: http://www.sliver07.org) dataset. We first introduce the evaluation measures and then show the experimental results on MICCAI-Sliver07, followed by the results on 3Dircadb.

4.1 Evaluation

We adopt VOE, RVD, ASD, root mean surface distance (RMSD) and max surface distance (MSD) as the evaluation measures. Among them, VOE and ASD are commonly adopted to measure segmentation accuracy. Two sets of voxels A and B represent the real and predicted segmentation result of liver, respectively. If the segmentation is perfect, VOE equals 0. If the prediction and reference do not overlap at all, VOE equals 1. vol means the volumetric of a set of voxels. The definition is as shown in equation:

$$\mathrm{VOE}(A, B) = 1 - \mathrm{vol}(A \cap B)/\mathrm{vol}(A \cup B) \tag{6}$$

RVD is used to determine whether the predicted liver segmentation results are in an over-segmentation or incomplete segmentation state. The value of RVD is 0, which means that the volume of predicted segmentation results is same as that of the real segmentation results instead of exactly the same. Therefore, RVD needs to be mixed with other indicators. The definition is shown in formula:

$$\mathrm{RVD}(A, B) = \frac{\mathrm{vol}(B - A)}{\mathrm{vol}(A)} \tag{7}$$

The ASD value is 0, indicating that liver is perfectly segmented. $S(A)$ denotes the set of surface voxels of A.

Table 1 Evaluation of DeepLab V2 method on MICCIA-Sliver07 test set

Test case	VOE (%)	Score	RVD (%)	Score	ASD (mm)	Score	RMSD (mm)	Score	MSD (mm)	Score	Total score
1	10.06	60.7	5.44	71.1	1.61	59.75	2.86	60.3	20.06	73.6	65.1
2	6.03	76.4	3.73	80.2	1.02	74.5	2.21	69.3	21.50	71.7	74.4
3	4.99	80.5	2.87	84.7	0.73	81.8	1.56	78.3	15.30	79.9	81.0
4	4.77	81.4	0.38	98.0	0.81	79.8	2.34	67.5	24.96	67.2	78.8
5	5.84	77.2	3.38	82.0	0.95	76.3	2.54	64.7	28.00	63.2	72.7
6	7.25	71.7	− 0.22	98.8	1.16	71.0	2.42	66.4	26.18	65.6	74.7
7	7.59	70.4	5.47	70.9	1.26	68.5	2.15	70.1	23.65	68.9	69.8
8	6.72	73.8	1.56	91.7	1.11	72.3	1.82	74.7	25.03	67.1	75.9
9	4.67	81.8	− 0.35	98.1	0.71	82.3	1.57	78.2	26.08	65.7	81.2
10	5.87	77.1	− 0.88	95.3	1.14	71.5	2.91	59.6	29.66	61.0	72.9
Avg	6.38	75.1	2.14	87.1	1.05	73.8	2.24	68.9	24.04	68.4	74.7

Table 2 Evaluation of our proposed method on MICCIA-Sliver07 test set

Test case	VOE (%)	Score	RVD (%)	Score	ASD (mm)	Score	RMSD (mm)	Score	MSD (mm)	Score	Total score
1	8.18	68.0	1.84	90.2	1.24	69.0	2.26	68.6	18.63	75.5	74.3
2	4.67	81.8	0.43	97.7	0.70	82.5	1.71	76.3	32.83	56.8	79.0
3	3.64	85.8	0.94	95.0	0.47	88.3	1.22	83.1	18.32	75.9	85.6
4	3.27	87.2	− 0.09	99.5	0.43	89.3	1.02	85.8	12.02	84.2	89.2
5	4.60	82.0	0.34	98.2	0.68	83.0	1.68	76.7	18.80	75.3	83.0
6	5.25	79.5	− 1.62	91.4	0.70	82.5	1.45	79.9	22.34	70.6	80.8
7	5.46	78.7	1.54	91.8	0.88	78.0	1.70	76.4	18.32	75.9	80.2
8	5.85	77.1	− 1.31	93.0	1.00	75.0	1.92	73.3	31.04	59.2	75.5
9	4.21	83.6	− 1.58	91.6	0.60	85.0	1.15	84.0	11.99	84.2	85.7
10	5.51	78.5	− 1.42	92.4	1.09	72.8	2.87	60.1	29.66	61.0	73.0
Avg	5.06	80.2	1.11	94.1	0.78	80.5	1.70	76.4	23.42	71.9	80.6

$d(s_A, S(B))$ represents the Euclidean distance from the surface voxel of A to the nearest surface voxel of B.

$$d(v, S(B)) = \min_{s_A \in S(B)} \|v - s_B\| \tag{8}$$

where $\|\cdot\|$ denotes the Euclidean distance.

The definition is as shown in formula:

$$\mathrm{ASE}(A, B) = \frac{\sum_{s_A \in S(A)} d(s_A, S(B)) + \sum_{s_B \in S(B)} d(s_B, S(A))}{|S(A)| + |S(B)|} \tag{9}$$

The MSD calculates the maximum Euclidean distance of the surface voxel. The definition is as shown in formula:

$$\mathrm{MSD}(A, B) = \max\left\{ \max_{s_A \in S(A)} d^2(s_A, S(B)), \max_{S_B \in S(B)} d^2(s_B, S(A)) \right\} \tag{10}$$

RMSD is similar to the ASD calculation method. RMSD needs to square, average, and square the shortest Euclidean distance between surface voxels. If the value of RMSD is smaller, liver segmentation is more accuracy. The definition is as shown in the formula:

$$\mathrm{RMSD}(A, B) = \frac{\sqrt{\sum_{s_A \in S(A)} d^2(s_A, S(B)) + \sum_{s_B \in S(B)} d^2(s_B, S(A))}}{\sqrt{|S(A)| + |S(B)|}} \tag{11}$$

score is 75 when each evaluation criterion reaches a given manual assessment reference. A_i and B_i are the true and predicted segmentation results of ith term, respectively. The score S_i is defined as follows:

$$S_i = \max\left(100 - 25 \times \frac{B_i}{A_i}, 0\right) \tag{12}$$

According to Ginneken [13], $\sigma_{\mathrm{VOE}} = 6.4\%$, $\sigma_{\mathrm{RVD}} = 4.7\%$, $\sigma_{\mathrm{ASD}} = 1.0\,\mathrm{mm}$, $\sigma_{\mathrm{RMSD}} = 1.8\,\mathrm{mm}$ and $\sigma_{\mathrm{MSD}} = 19\,\mathrm{mm}$ were used as the reference values when score is 75. The corresponding score is calculated as [18].

Table 3 Comparision with other state-of-the-art methods on MICCIA-Sliver07 test set

Method	VOE (%)	Score	RVD (%)	Score	ASD (mm)	Score	RMSD (mm)	Score	MSD (mm)	Score	Total score
Li et al. [25]	6.24	–	1.18	–	1.03	–	2.11	–	18.82	–	–
Alshaikhli et al. [1]	6.44	74.9	1.53	89.7	0.95	76.3	**1.58**	**78.1**	**15.92**	**79.1**	79.6
Kainmüller et al. [20]	6.09	76.2	− 2.86	84.7	0.95	76.3	1.87	74.0	18.69	75.4	77.3
Wimmer et al. [33]	6.47	74.7	1.04	86.4	1.02	74.5	2.00	72.3	18.32	75.9	76.8
Linguraru et al. [26]	6.37	75.1	2.26	85.0	1.00	74.9	1.92	73.4	20.75	72.7	76.2
Heimann et al. [18]	7.73	69.8	1.66	87.9	1.39	65.2	3.25	54.9	30.07	60.4	67.6
Kinda et al. [21]	8.91	65.2	1.21	80.0	1.52	61.9	3.47	51.8	29.27	61.5	64.1
Fang et al. [27]	5.90	77.0	2.70	85.6	0.91	77.3	1.88	73.8	18.94	75.1	77.8
Only DeepLab	6.38	75.1	2.14	87.1	1.05	73.8	2.24	68.9	24.04	68.4	74.7
The proposed	**5.06**	**80.2**	**1.11**	**94.1**	**0.78**	**80.5**	1.70	76.4	23.42	71.9	**80.6**

Bold values indicate the highest rank of the current indicator

The total score S_{total} is the overall score, which is defined as follows:

$$S_{\text{total}} = \sum_{i=1}^{n} S_i \Big/ n \tag{13}$$

4.2 MICCIA-Sliver07

Tables 1 and 2 present experimental results with DeepLab V2 and our method, respectively. We calculate five metrics according to the method in [18]. Table 2 reveals that the score of VOE, RVD, ASD, RMSD, and MSD are 80.2, 94.1, 80.5, 76.4, and 71.9, respectively. In Table 3, we compare the proposed methods with other eight representative segmentation methods [1, 18, 20, 21, 25–27, 33]. The results reveal that the proposed method yields improvement over DeepLab V2 model. Figure 4 presents the segmentation results with our method and DeepLab. The segmentation results with the ground truth are the area circled by green lines. The results by our proposed method are represented by red lines. The region covered by blue lines is obtained by DeepLab. Figure 4 demonstrates that our method segments small liver effectively. Obviously, liver segmentation is more accurate with the proposed method. The total score reaches 80.6, which exceeds all comparison methods. Over all, the proposed method achieves better performance. The reasons are as follows:

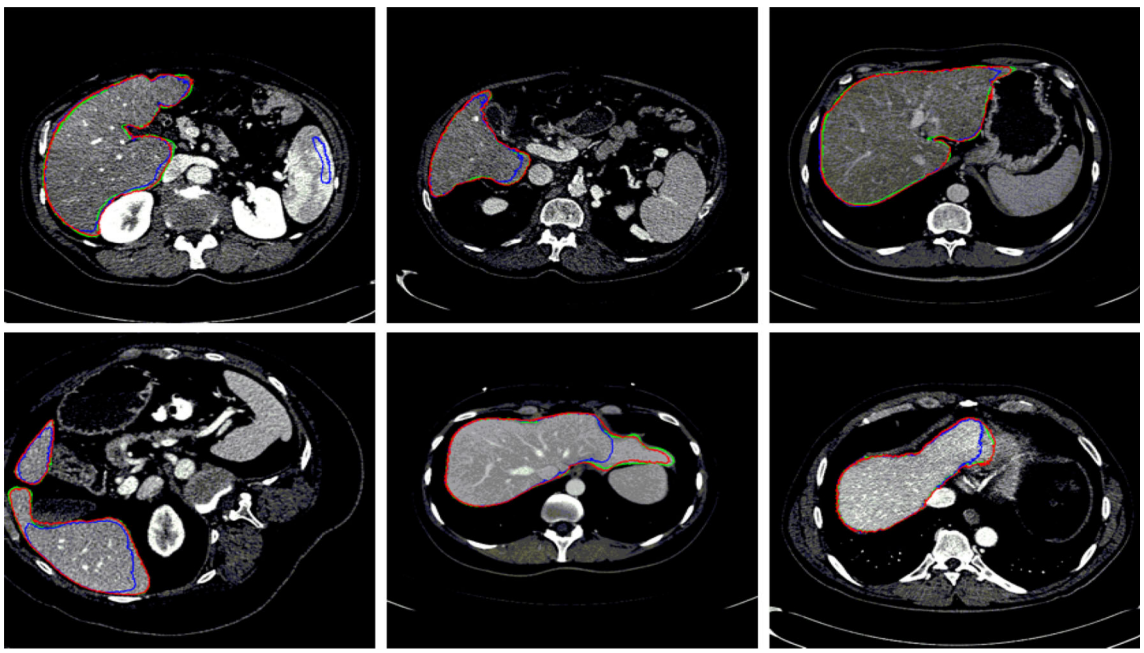

Fig. 4 Example of liver segmentation results with the ground truth in green. The result by our proposed method is in red, and the result by the only DeepLab is in blue

Table 4 Comparision with other state-of-the-art methods on 3Dircadb dataset

3Dircadb	VOE (%)	RVD (%)	ASD (mm)	RMSD (mm)	MSD (mm)
Chuang et al. [7]	12.99 ± 5.04	− 5.66 ± 5.59	2.24 ± 1.08	–	25.74 ± 8.85
Kirscher et al. [22]	–	− 3.62 ± 5.50	1.94 ± 1.10	4.47 ± 3.30	34.60 ± 17.70
Li et al. [25]	9.15 ± 1.44	− 0.07 ± 3.64	1.55 ± 0.39	3.15 ± 0.98	28.22 ± 8.31
Erdt et al. [11]	10.34 ± 3.11	1.55 ± 6.49	1.74 ± 0.59	3.51 ± 1.16	26.83 ± 8.87
Fang Lu et al. [27]	9.36 ± 3.34	0.97 ± 3.26	1.89 ± 1.08	4.15 ± 3.16	33.14 ± 16.36
The proposed	8.67 ± 0.815	0.57 ± 2.53	1.37 ± 0.41	3.56 ± 0.81	27.01 ± 7.28

Firstly, the detection of liver region by Faster R-CNN can reduce the segmentation scope, and solve the problem of grayscale similarity. Secondly, because DeepLab is a semantic segmentation method, segmentation results are affected by serious pathological phenomena in pathological images. Finally, fully connected CRF can solve partial volume effect problem. In summary, the proposed method is superior to others.

4.3 3Dircadb

The comparison of our method with the state-of-the-art methods [7, 11, 21, 24, 26] on 3Dircadb dataset is reported in Table 4. It reveals that our explored method substantially outperforms the concurrent work in terms of VOE, RVD and ASD. However, Erdt's and Chuang's methods show a little better than our approach in terms of RMSD and MSD metrics. The segmentation result is shown in Fig. 5. Apparently, our proposed method can segment liver with small volume and complex contour more accurately than DeepLab. Overall, the DSL model proposed in this work achieves good results in various measures, which are better than those of concurrent works.

5 Conclusions

In this work, we proposed a two-stage automatic liver segmentation method. For the first time, we combine Faster R-CNN and DeepLab to form an automatic segmentation

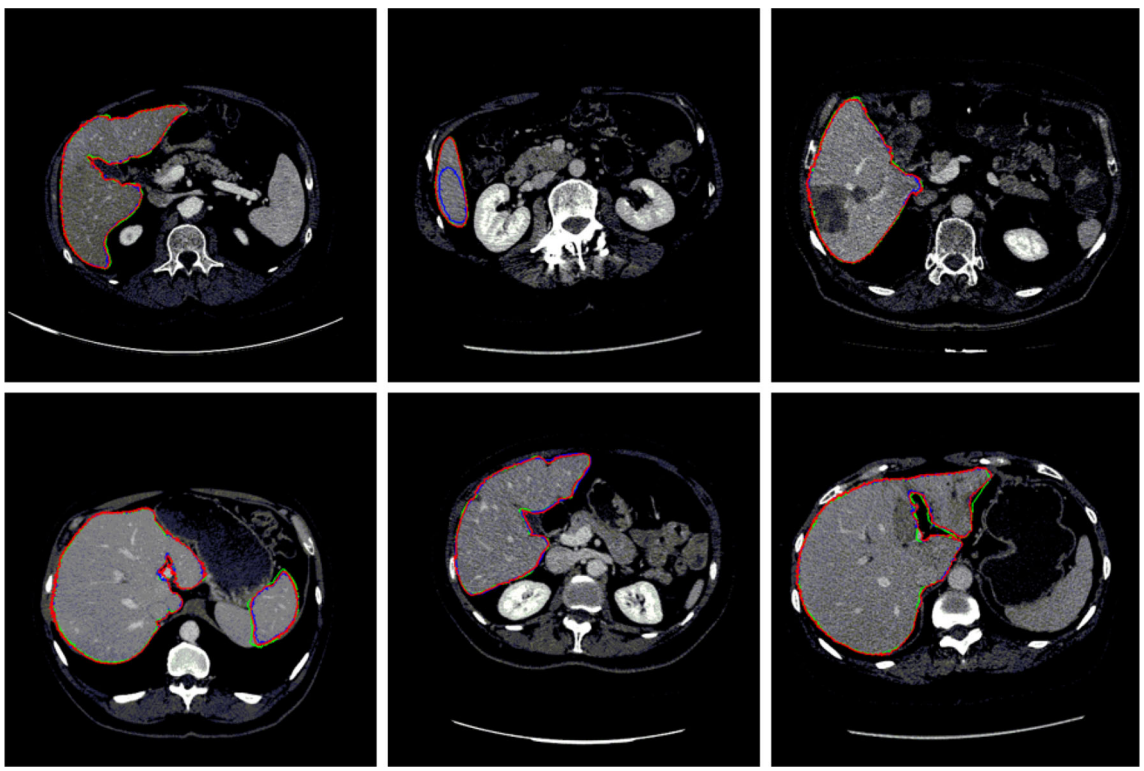

Fig. 5 Example of liver segmentation results with the ground truth in green. The result by our proposed method is in red, and the result by the only DeepLab is in blue

method that eliminates the need to manually extract features or user interaction in the whole procedure. The highlight of the proposed method is that it can segment liver with small volume and complex contour accurately. Compared with other concurrent works, our DSL model has good results mainly for three reasons: First, deep learning methods are used in liver detection and segmentation stages to minimize the impact of human factors on the segmentation effect. Second, in the segmentation stage, with the idea of residual learning, the hop structure is used to fuse the multi-level information of network, and the useful information of middle layer is reused to extract more abundant and significant context information, which can balance the two contradictory goals of location and classification of liver semantic segmentation. Thirdly, the proposed model inputs the processed detection result image into the segmentation model, which narrows the range of subsequent semantic segmentation. Consequently, it not only ensures that the segmentation range is as small as possible, but also covers the liver region, which can solve the troubles caused by volume effects and adjacent tissues.

Compared with the state-of-the-art methods, DSL method achieved the best performance in terms of total score. In particular, VOE, RVD, and ASD metrics are significantly higher than other methods, which proves the effectiveness of the proposed method. How to share parameter in two stages is left as future work.

Acknowledgements This work was partly supported by the National Nature Science Foundation of China (No. 61309013 and No. 51608070) and Chongqing Basic and frontier research projects (No. CSTC2014JCYJA40042 and No. CSTC2016JCYJA0022).

References

1. Alshaikhli SDS, Yang MY, Rosenhahn B (2015) Automatic 3D liver segmentation using sparse representation of global and local image information via level set formulation. arXiv:1508.01521
2. Boykov Y, Funka-Lea G (2006) Graph cuts and efficient N-D image segmentation. Int J Comput Vis 70(2):109–131
3. Burkhardt H (2010) Integration of morphology and graph-based techniques for fully automatic liver segmentation. Majlesi J Electr Eng 4(3):59–66
4. Campadelli P, Casiraghi E (2009) Liver segmentation from CT scans: a survey. Artif Intell Med 45(2):185–196
5. Chen LC, Papandreou G, Kokkinos I et al (2014) Semantic image segmentation with deep convolutional nets and fully connected CRFs. Comput Sci 4:357–361
6. Chen LC, Papandreou G, Kokkinos I et al (2017) DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Trans Pattern Anal Mach Intell 40(4):834–848
7. Chung F, Delingette H (2013) Regional appearance modeling based on the clustering of intensity profiles. Comput Vis Image Underst 117(6):705–717
8. Dawant BM, Li R, Lennon B et al (2007) Semi-automatic segmentation of the liver and its evaluation on the MICCAI 2007 grand challenge data set. In: Proceedings of MICCAI workshop 3D segmentation clinic: a grand challenge, pp 215–221
9. Dong C, Chen YW, Tateyama T et al (2016) A knowledge-based interactive liver segmentation using random walks. In: International conference on fuzzy systems and knowledge discovery. IEEE, pp 1731–1736
10. Dou Q, Chen H, Jin Y et al (2016) 3D Deeply supervised network for automatic liver segmentation from CT volumes, pp 149–157
11. Erdt M, Steger S, Kirschner M et al (2010) Fast automatic liver segmentation combining learned shape priors with observed shape deviation. In: IEEE, international symposium on computer-based medical systems. IEEE, pp 249–254
12. Gambino O, Vitabile S, Re GL et al (2010) Automatic volumetric liver segmentation using texture based region growing. In: International conference on complex, intelligent and software intensive systems. IEEE Computer Society, pp 146–152
13. Ginneken BV, Heimann T, Styner M (2007) 3D Segmentation in the clinic: a grand challenge. In: Proceedings of 3D segmentation in the clinic: a grand challenge. Springer, Brisbane, pp 7–15
14. Girshick R (2015) Fast R-CNN. In: IEEE international conference on computer vision. IEEE, pp 1440–1448
15. Girshick R, Donahue J, Darrell T et al (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: IEEE international conference on computer vision and pattern recognition, pp 580–587
16. He B, Huang C, Sharp G et al (2016) Fast automatic 3D liver segmentation based on a three-level AdaBoost-guided active shape model. Med Phys 43(5):2421
17. Heimann T, Meinzer HP, Wolf I (2010) A statistical deformable model for the segmentation of liver CT volumes. In: MICCAI workshop on 3D Segmentation in the Clinic
18. Heimann T, Van GB, Styner MA et al (2009) Comparison and evaluation of methods for liver segmentation from CT datasets. IEEE Trans Med Imaging 28(8):1251–1265
19. Jansen J, Schreurs R, Dubois L et al (2015) Orbital volume analysis: validation of a semi-automatic software segmentation method. Int J Comput Assist Radiol Surg 11(1):11–18
20. Kainmüller D, Lange T, Lamecker H (2007) Shape constrained automatic segmentation of the liver based on a heuristic intensity model. In: MICCAI workshop on 3D Segmentation in the Clinic, pp 109–116
21. Saddi KA, Rousson M et al (2007) Global-to-local shape matching for liver segmentation in CT imaging
22. Kirschner M (2013) The probabilistic active shape model: from model construction to flexible medical image segmentation. PhD dissertation
23. Krähenbühl P, Koltun V (2011) Efficient inference in fully connected CRFs with Gaussian edge potentials. In: Proceedings of advances in neural information processing systems, vol 24, pp 109–117
24. Liao M, Zhao YQ, Wang W et al (2016) Efficient liver segmentation in CT images based on graph cuts and bottleneck detection. Phys Med 32(11):1383
25. Li G, Chen X, Shi F et al (2015) Automatic liver segmentation based on shape constraints and deformable graph cut in CT Images. IEEE Trans Image Process 24(12):5315
26. Linguraru MG, Richbourg WJ, Watt JM et al (2011) Liver and tumor segmentation and analysis from CT of diseased patients via a generic affine invariant shape parameterization and graph cuts. In: International conference on abdominal imaging: computational and clinical applications. Springer, Berlin, pp 198–206
27. Lu F, Wu F, Hu P et al (2017) Automatic 3D liver location and segmentation via convolutional neural network and graph cut. Int J Comput Assist Radiol Surg 12(2):171–182
28. Lu J, Shi L, Deng M et al (2011) An interactive approach to liver segmentation in CT based on deformable model integrated with

attractor force. In: International conference on machine learning and cybernetics. IEEE, pp 1660–1665
29. Njg W (2017) Alternative RNA splicing in the pathogenesis of liver disease. Front Endocrinol 8:133
30. Ren S, He K, Girshick R et al (2015) Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell 39(6):1137
31. Schmidt G, Athelogou MA, Schönmeyer R et al (2007) Cognition network technology for a fully automated 3-D segmentation of liver. In: MICCAI workshop on 3D Segmentation in the clinic: a grand challenge
32. Wang N, Huang L L, Zhang B (2010) A fast hybrid method for interactive liver segmentation. In: Chinese conference on pattern recognition, pp 1–5
33. Wimmer A, Soza G, Hornegger J (2009) A generic probabilistic active shape model for organ segmentation. In: Medical image computing and computer-assisted intervention–MICCAI 2009. Springer, Berlin, pp 26–33
34. Yan J, Schwartz LH, Zhao B (2015) Semiautomatic segmentation of liver metastases on volumetric CT images. Med Phys 42(11):6283–6293
35. Yang D, Xu D, Zhou S K et al (2017) Automatic liver segmentation using an adversarial image-to-image network
36. Yang X, Yu HC, Choi Y et al (2014) A hybrid semi-automatic method for liver segmentation based on level-set methods using multiple seed points. Comput Methods Programs Biomed 113(1):69–79
37. Shelhamer E, Long J, Darrell T (2014) Fully convolutional networks for semantic segmentation. IEEE Trans Pattern Anal Mach Intell 39(4):640–651
38. Shrivastava A, Gupta A, Girshick R (2016) Training region-based object detectors with online hard example mining. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 761–769