Accurate Liver segmentation using 3D CNNs with high level shape constraints

Man Tan, Xiongwei Mao, Fa Wu, and Dexing Kong*

***Abstract*—Automatic liver segmentation from abdominal computed tomography (CT) images is a fundamental task in computer-assisted liver surgery programs. Recently, deep convolutional neural networks (CNNs) are served as the first choice in many volumetric segmentation tasks. However, the most commonly used cross-entropy loss treats each pixel independently and equally, which makes the network sensitive to fuzzy boundaries and heterogeneous pathologies, especially when the data is scarce. In this work, we propose an automatic segmentation framework based on a 3D CNN with a hybrid loss function. The hybrid loss function consists of three parts. The first part is an adaptively weighted cross-entropy loss, which pays more attention on misclassified pixels. The second part is an edge-preserved smoothness loss, which guarantees that neighbouring pixels with the same label have similar outputs, while neighbouring pixels with different labels have dissimilar outputs. The third part of loss is a shape constraint used to model high level structure differences. In our experiments, data augmentation is performed both in the training stage and the test stage. We extensively evaluated our method on two datasets: the Segmentation of the Liver Competition 2007 (Sliver07), and the Combined (CT-MR) Healthy Abdominal Organ Segmentation (CHAOS) Challenge.**

***Index Terms*—automatic liver segmentation, convolutional neural networks, hybrid loss, high level shape constraint.**

I. Introduction

Accurate liver segmentation on three dimensional (3D) computed tomography (CT) is critical in many clinical applications, such as treatment planning and postoperative assessment. However, the manual delineation on each slice of liver is a laborious and huge time-consuming process. As a result, manual segmentation is not suited for a busy clinical practice in high volume settings [1]. In order to accelerate and facilitate diagnosis, therapy planning and monitoring, fast and accurate automatic liver segmentation is highly demanded.

Automatic liver segmentation from CT images is a very challenging task due to the wide variety of liver shapes, fuzzy boundaries, the presence of various pathologies and high-intensity intrahepatic veins. Fig. 1 shows some examples of CT images illustrating the challenges. To tackle these difficulties, extensive works have been proposed. Comprehensive surveys on liver CT image segmentation methods and techniques were presented by Campadelli *et al.* [2] and Mharib *et al.* [3]. Heimann *et al.* [4] also presented a detailed comparison study among different methods for liver segmentation based on results from the "MICCAI 2007 Grand Challenge" workshop. Generally, all liver segmentation algorithms can be categorized into three classes according to the image features they work on, including gray level based methods, structure based methods and texture based methods [5].

Gray level intensity is the most obvious feature of CT images. Many gray level based segmentation algorithms are developed, including region growing methods [6], [7], active contours [8], [9], graph cuts [10], [11], clustering based algorithms [12], [13] and so on. For example, Rusko *et al.* [14] first determined a seed region based on intensity histogram and separated the heart from the liver to eliminate over-segmented regions. Then, they employed an advanced region growing method to segment the liver region followed by various postprocessing steps to prevent under-segmentation. Lim *et al.* [15] extracted the initial liver volume by exploiting prior information from manually segmented CT samples. Next, they utilized multiscale morphological filters with region-labeling and clustering to detect the search range and generate the initial liver boundaries. Finally, contour-based segmentation was applied to find the final liver contour. Massoptier *et al.* [16] used the mean shift filter to remove the noise from homogenous areas while keeping clear and sharp edges. And then, they applied a graph cut based method initialized by an adaptive threshold to segment the liver. Zhao *et al.* [13] employed a fuzzy C-means clustering algorithm and morphological reconstruction filtering to segment the initial liver CT image. Then, a neural network was trained to classify the candidate regions. Although some of the aforementioned approaches have achieved promising performance, there are some drawbacks in gray level based methods: they need additional algorithms for initial conditions (seed points, initial contours/regions), and may be sensitive to initial conditions; they are challenging to prevent over-segmentation caused by similar intensities between target and background regions, and avoid under-segmentation caused by inhomogeneous target regions.

The central hypothesis of structure based methods is that structures of interested objects have a repetitive form of geometry. Generally, deformable models, statistical shape models (SSMs), and probabilistic atlases built by a set of examples of shape are employed to generate segmentations. Kainmuller *et al.* [17] presented a fully automatic 3D segmentation method for the liver based on a combination of a constrained free-form and statistical deformable model. Erdt *et al.* [18] presented a fully automatic multi-tiered statistical shape model for the liver that combined learned local shape constraints with observed

Asterisk indicates corresponding author.

M. Tan and F. Wu are with the School of Mathematical Sciences, Zhejiang University, Hangzhou, Zhejiang, China.

X. Mao is with the Zhejiang University School Hospital, Hangzhou, Zhejiang, China. He is also with the First Affiliated Hospital, School of Medicine, Zhejiang University, Hangzhou, Zhejiang, China.

*D. Kong is with the School of Mathematical Sciences, Zhejiang University, Hangzhou, Zhejiang, China (e-mail: dkong@zju.edu.cn).

shape deviation during adaptation. Van *et al.* [19] used a statistical classifier and two types of features, gray-level features and location features obtained from a multi-atlas registration procedure, to label pixels. Structure based methods are more robust by capturing anatomical knowledge about the shape, size and position of liver. However, structure based methods may not model the large shape variations well with limited training data.

In texture based segmentations, a feature-based classifier is trained to label unseen images. Luo *et al.* [20] used wavelet coefficients as texture descriptors and implemented support vector machines (SVMs) to classify the data into pixel-wised liver area or non-liver area. Then, integrated morphological operations were designed to remove noise and delineate the liver. Ling *et al.* [21] presented a hierarchical framework to efficiently and effectively monitor the accuracy propagation in a coarse-to-fine fashion. And marginal space learning and steerable features were applied for robust boundary inference. Unfortunately, these methods heavily rely on handcrafted features and do not take full advantage of 3D spatial information.

In recent years, with the remarkable success of deep convolutional neural networks (CNNs) in nature image processing [22]–[26], many studies have used the representative features learned by CNNs to deal with the segmentation of liver. Dou *et al.* [27] presented a novel 3D deeply supervised fully convolutional network for automatic liver segmentation. They further employed a fully connected conditional random field (CRF) [28] to refine the segmentation results. Finally, they achieved a volumetric overlap error (VOE) of 5.42% and an average symmetric surface distance (ASSD) of 0.79 mm on the Sliver07 dataset [4]. Hu *et al.* [29] proposed an automatic segmentation framework based on a 3D CNN and globally optimized surface evolution. They first used a trained deep 3D CNN to learn a subject-specific probability map of liver that was acted as a shape prior. Then, both global and local appearance information from the prior segmentation were adaptively incorporated into a segmentation model, which was globally optimized in a surface evolution way. Finally, they achieved a mean Dice similarity coefficient (DSC) of 97.25%, and an ASSD of 0.84 mm on the Sliver07 dataset. Compared to previous methods, these methods are superior as they can automatically produce a subject-specific segmentation probability map without difficult handcrafted features, complex registration or shape deformation. However, these methods heavily rely on image intensities. Thus, the probability maps still suffer from some limitations of gray level based methods. To solve these shortcomings, these methods all take further postprocessing steps to improve the segmentation results. It's worth noting that CNNs usually take only a few seconds to generate the probability maps, while the postprocessing process (e.g. graph cut [30], CRF [27], level set [31]) often takes tens or even hundreds of seconds.

In this work, we propose a novel end-to-end system, called shape-constrained densely connected segmentation network (SC-SegNet). Compared with other existing algorithms, there are two major novelties in the proposed framework:

- We design a hybrid loss to train the segmentation network. The loss consists of three parts, including an adaptively weighted cross entropy, an edge-preserved smoothness loss and a high-level shape constraint.
- We validated the proposed method on two separate clinical databases.

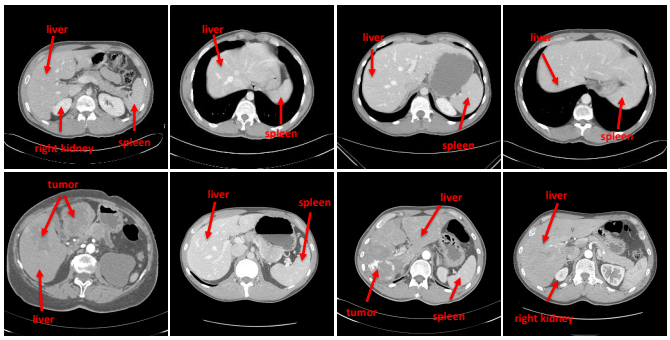

Fig. 1. Examples of contrast-enhanced CT images illustrating the challenges for accurate liver segmentation. Each row shows the examples form the CHAOS challenge, and the Sliver07 challenge, respectively.

This paper is organized in the following manner. We start by introducing the data used in our study in Section II and explaining the details of our system in Section III. In Section IV, we present the details of our experimental setup. And in Section V, we report the results of a set of experiments and compare our method with other related works. Further discussion on some key issues is presented in Section VI. Finally, the summary is given in Section VII.

II. Materials

In our experiments, two different datasets are separately processed and validated. The first dataset is from the CHAOS challenge [32]. There are five competition categories in this challenge, we only consider the task of liver segmentation from CT images. This dataset only contains healthy livers aligned in the same direction and patient position. Among then, there are 20 clinical images with reference segmentation and 20 test images without available ground-truths. All images are acquired from upper abdomen area at portal venous phase after contrast agent injection and have the same axial dimensions of 512×512 with slice number varying from 77 to 105. The pixel spacing varies from 0.7 to 0.8 mm in x-y direction, and slice distance varies from 3.0 to 3.2 mm. The second dataset is from the Sliver07 challenge [4]. It includes 20 clinical images with reference segmentation and 10 test images without available ground-truths for participants. All images are acquired contrast-dye-enhanced in the central venous phase and have the same axial dimensions of 512×512 with slice number varying from 64 to 502. The pixel spacing varies from 0.55 to 0.8 mm in x-y direction, and slice distance varies from 1.0 to 3.0 mm. Most images in the study are pathologic and include tumors, metastasis and cysts of different sizes. Some examples of CT scans are shown in Fig. 1.

III. Methods

The framework of our method is presented in Fig. 2. Two networks are incorporated: a liver shape autoencoder and a

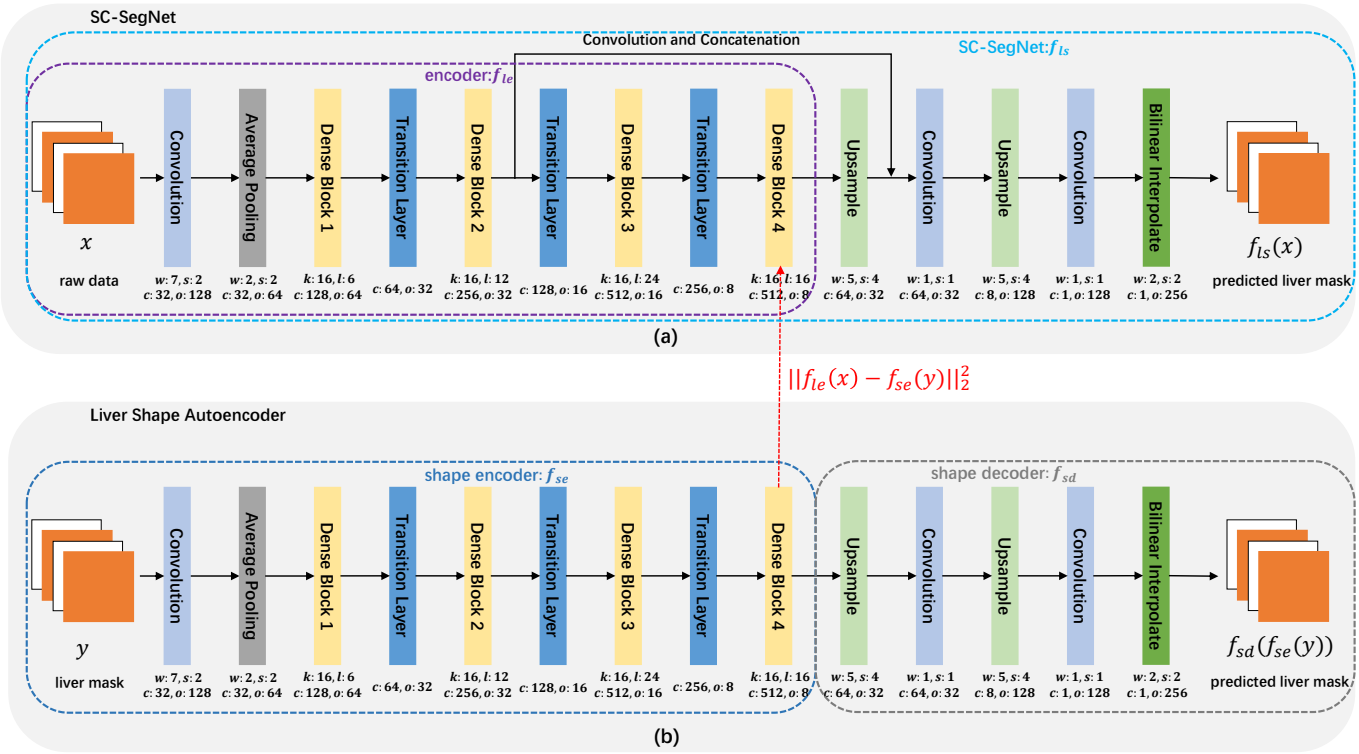

Fig. 2. The framework of the proposed liver segmentation method. The shape autoencoder is first trained to obtain the liver shape codes with the liver masks as input. Then, the SC-SegNet is trained under the supervision of both the liver masks and the shape codes. In the figure, 'w' denotes the kernel size of filter, 's' indicates the stride, 'c' indicates the number of output channels, 'o' indicates the size of output feature maps, 'k' denotes the growth rate of dense block, and 'l' indicates the number of layers in each dense block. If not mentioned, all parameters are the same in three axes.

shape-constrained densely connected segmentation network (SC-SegNet). We employ the cascaded learning strategy to train the system. First, the liver shape autoencoder is trained to obtain compressed codes of liver shapes. Then, the SC-SegNet is trained under the supervision of both the segmentation masks and the learned shape codes. In the following subsections, we will describe the system in detail.

A. Liver Shape Autoencoder

In order to obtain compressed codes of liver shapes, many methods can be used, such as SSMs and autoencoders. SSMs are widely used to analyse shape variations. However, due to the large variations of shape, the liver is a very challenging structure to describe with SSMs [4]. As thus, we use an autoencoder to model the liver shape.

The shape autoencoder is designed based on the structure of DenseNet [26], which uses densely connected blocks (dense blocks) to ensure maximum information flow between layers. The architecture of the dense block is shown in Fig. 3. For each layer in the block, the feature maps of all preceding layers are used as input, and its own feature maps are used as input for all subsequent layers. If the input of this block has k_0 feature maps and each layer produces k (k is called growth rate) feature maps, it follows that a dense block with l layers produces $k_l = k_0 + k \times (l-1)$ feature maps. Due to its dense connectivity pattern, dense block has several advantages: alleviating the vanishing-gradient problem, strengthening feature propagation, and encouraging feature reuse.

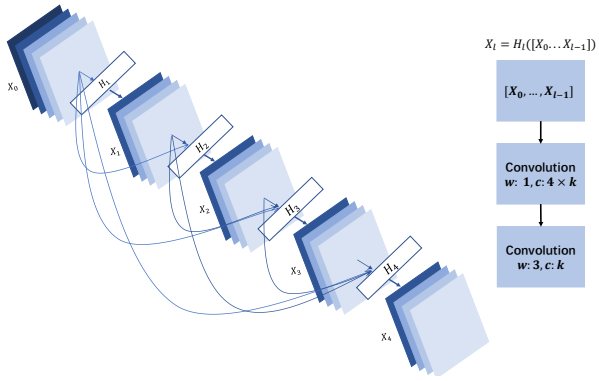

Fig. 3. A 4-layer dense block. The input has 5 feature maps, and each layer in the block produces $k = 4$ feature maps. Finally, the block outputs 21 feature maps (the same symbol definition as in Fig. 2).

The architecture of the shape autoencoder is illustrated in Fig.2(b), which consists of a shape encoder and a decoder. The shape encoder is a typical 3D DenseNet, which is composed of a convolutional (Conv) layer, a pooling layer, three transition layers, and four dense blocks. In order to reduce the spatial resolution of feature maps, an average pooling layer and three transition layers are used. The transition layer consists of an $1 \times 1 \times 1$ convolutional layer and a $2 \times 2 \times 2$ average pooling layer. To further improve model compactness, the number of feature maps is reduced by half at transition layers. The shape decoder consists of two upsampling layers, two convolutional

layers, and a bilinear interpolation operation. Each upsampling is the backwards strided (stride $4 \times 4 \times 4$) convolution to expand the size of feature maps [24]. Batch normalization (BN) [33] and Rectified linear units (ReLU) are employed after all convolutional layers and upsampling layers except the last convolutional layer. The bilinear interpolation is followed by logistic units to predict the probability of each pixel that belongs to liver. Except for the first convolutional layer is with a stride of 2, all other convolutional layers are with the stride of 1. In order to preserve the resolution of feature maps, we set all convolutional layers with padding in three axes. In our experiments, we set the operation H of each layer in all dense blocks to be BN-ReLU-Conv($1 \times 1 \times 1$)-BN-ReLU-Conv($3 \times 3 \times 3$). This further helps reduce the number of parameters. The dense blocks in the shape encoder are all with $k = 16$. The numbers of layers in four dense blocks are 6, 12, 24 and 16, respectively. Most importantly, all operations are implemented in 3D style.

Let f_{se} denote the shape encoder with a liver mask $y \in \{0,1\}^{W \times H \times S}$ as input (W, H, S indicate the width, height, and number of slices of input), f_{sd} denote the shape decoder with $f_{se}(y)$ as input, and g denote the logistic operation. Then, the output of the shape autoencoder can be written as

$$y^{SA} = g(f_{sd}(f_{se}(y))). \tag{1}$$

To train the network, we use the negative log-likelihood as the loss function, which is described as

$$loss_{SA} = -\sum_{y} \frac{1}{n} \sum_{i} y_i \log y_i^{SA} + (1-y_i) \log(1-y_i^{SA}), \tag{2}$$

where y_i indicates the i-th pixel of the liver mask y, y_i^{SA} denotes the output probability of pixel i that belongs to liver from the shape autoencoder, and n denotes the total number of pixels of y.

B. SC-SegNet

With the help of the shape codes learned by the proposed shape autoencoder, we develop a shape-constrained densely connected segmentation network (SC-SegNet). The architecture of the SC-SegNet illustrated in Fig. 2(a) is designed based on FCN [24], DenseNet [26], and U-Net [34]. The network consists of an encoder and a decoder. The encoder has the same architecture as the shape encoder f_{se}. The decoder contains two steps. The first step consists of an upsampling, a concatenation operation with the correspondingly convoluted feature maps from the contracting path, and a convolution. The second step consists of an upsampling, a convolution, and a bilinear interpolation. In the SC-SegNet, other architecture settings are same as the shape autoencoder.

Let $x \in R^{W \times H \times S}$ denote a training sample with a ground-truth mask $y \in \{0,1\}^{W \times H \times S}$, f_{le} denote the encoder of the SC-SegNet, and f_{ls} denote the whole SC-SegNet. Then the output of the SC-SegNet can be written as

$$y^{SC} = g(f_{ls}(x)). \tag{3}$$

Since the most used cross-entropy loss treats each pixel independently and equally, it may not be able to handle the imbalance between different categories, fuzzy boundaries and heterogeneous pathologies. To address these issues, a loss function composed of four parts is proposed to train the SC-SegNet. The first part of loss is the typical cross-entropy loss,

$$\begin{aligned} loss_{ce} &= \sum_{(x,y)} \frac{1}{n} \sum_{i} CE(y_i, y_i^{SC}) \\ &\equiv -\sum_{(x,y)} \frac{1}{n} \sum_{i} y_i \log y_i^{SC} + (1-y_i) \log(1-y_i^{SC}), \end{aligned} \tag{4}$$

where y_i^{SC} denotes the output probability of pixel i from the SC-SegNet. The second part of loss is a shape constraint, which minimizes the difference between $f_{le}(x)$ and $f_{se}(y)$,

$$loss_{sc} = \sum_{(x,y)} \frac{1}{n} ||f_{le}(x) - f_{se}(y)||_2^2. \tag{5}$$

To put it simply, we try to make the features learned by the SC-SegNet consistent with the shape codes produced by the liver shape autoencoder. In order to make the output probability maps smoother inside and outside the liver, we design an edge-preserved smoothness regularizer. The regularizer penalizes nearby similar pixels that are assigned different outputs inside and outside the liver, and is designed as

$$loss_{sr} = \sum_{(x,y)} \frac{1}{n} \sum_{i} \frac{1}{w_i} \sum_{j \in \Omega_i \setminus i} w_{ij} (f_{ls}(x)_j - f_{ls}(x)_i)^2. \tag{6}$$

where $f_{ls}(x)_i$ indicates the value of pixel i in $f_{ls}(x)$, Ω_i is the $5 \times 5 \times 5$ neighborhood of i, w_{ij} is the contribution from the pixel j to the pixel i, and w_i is the sum of the weights of all pixels in the neighborhood. Based on pixel intensities I_i and I_j, labels y_i and y_j, and network output $f_{ls}(x)_j$, w_{ij} is defined as

$$w_{ij} = \mathbf{1}_{(\mathbf{1}_{f_{ls}(x)_j > 0} = y_j)} (-1)^{\mathbf{1}_{y_i \neq y_j}} DM(I_i, I_j), \tag{7}$$

where $\mathbf{1}_A$ is an indicator function that returns 1 when A is true, otherwise returns 0. The first term $\mathbf{1}_{(\mathbf{1}_{f_{ls}(x)_j > 0} = y_j)}$ means that only the adjacent pixels whose current prediction is the same as its true label will be used for calculation. It makes sure that each pixel is updated in the right direction. The second term $(-1)^{\mathbf{1}_{y_i \neq y_j}}$ equals to 1 when $y_i = y_j$, while -1 when $y_i \neq y_j$. This is inspired by the fact that adjacent pixels with same label should have similar outputs, adjacent pixels with different labels should have dissimilar outputs. The third term is the intensity difference measure of adjacent pixels, and is defined as

$$DM(I_i, I_j) = \begin{cases} 1 - |I_i - I_j|^{\frac{1}{2}} & y_i = y_j, \\ |I_i - I_j|^{\frac{1}{2}} & y_i \neq y_j. \end{cases} \tag{8}$$

When $y_i = y_j$, pixels with similar intensities will have greater weights, and when $y_i \neq y_j$, pixels with similar intensities will have smaller weights. Fig. 4 shows a simple example of the loss of a pixel in the 3×3 neighborhood.

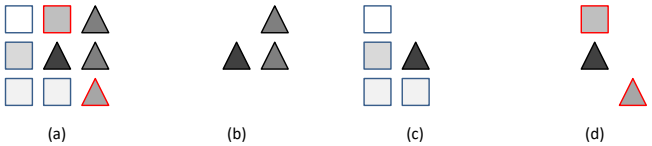

Fig. 4. An example of $loss_{sr}$. In the 3×3 neighborhood image of the central pixel, different shapes indicate different class, and red edges indicate misclassified pixels (a). Given whether the network correctly classifies each pixel and whether the nearby pixels are in the same class, all adjacent pixels can be divided into three groups. Correctly classified adjacent pixels with the same label should have similar outputs, and similar intensities lead to bigger weights (b). Correctly classified adjacent pixels with different labels should have dissimilar outputs, and similar intensities lead to smaller weights (c). Misclassified pixels are ignored because they will update the output of the central pixel in the wrong direction (d).

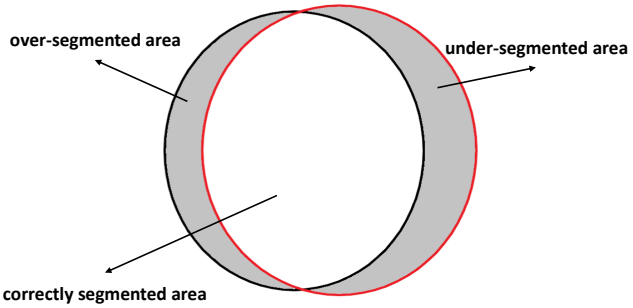

Fig. 5. An example of the pixels processed by $loss_{ec}$. Suppose the black line delineates the segmentation results produced by the network, and the red line delineates the reference segmentation results. It can be seen that $loss_{ec}$ focuses on under- and over-segmented areas (in gray color).

Due to the first indicator in (6), we find that pixels in the misclassification area do not participate in this part of loss. As thus, we introduce the fourth part of loss as

$$loss_{ec} = \sum_{(x,y)} \frac{1}{n} \sum_{i} \mathbf{1}_{(\mathbf{1}_{f_{ls}(x)_i > 0 \neq y_i})} CE(y_i, y_i^{SC}). \quad (9)$$

This part of loss can be seen as a technique of hard negative mining that allows the network to focus on pixels that are misclassified [35]. Fig. 5 shows an example of the pixels processed by $loss_{ec}$.

Combined with the four parts of loss, the final loss function can be written as:

$$\begin{aligned} loss = & \sum_{(x,y)} \frac{1}{n} \sum_{i} (1 + \gamma \mathbf{1}_{(\mathbf{1}_{f_{ls}(x)_i > 0 \neq y_i})}) CE(y_i, y_i^{SC}) \\ & + \beta \sum_{(x,y)} \frac{1}{n} \sum_{i} \frac{1}{w_i} \sum_{j \in \Omega_i \setminus i} w_{ij} (f_{ls}(x)_j - f_{ls}(x)_i)^2 \\ & + \alpha \sum_{(x,y)} \frac{1}{n} ||f_{le}(x) - f_{se}(y)||_2^2, \end{aligned} \quad (10)$$

where α, β, γ are hyperparameters used to balance these four parts of loss. The final hybrid loss can be considered to contain three parts, including an adaptively weighted cross entropy, an edge-preserved smoothness loss and a high-level shape constraint.

IV. Experimental Setting

In this section, we present the details of our experimental setup. Importantly, all settings used in both datasets are the same.

A. Preprocessing

We applied image preprocessing, including several steps. First, the pixel intensity range was normalized from (-110,190 Houndsfield Unit) to (0,1). Intensity > 190 was set to 1, and < -110 was set to 0. In order to reduce computational complexity and memory usage, all images were resampled to have the same resolution of 256×256 in the axial. In the test stage, segmentation results were resampled back to the original scale (512×512 in the axial). All the preprocessed steps were applied to both training and test datasets.

B. Data Augmentation

In our experiments, data augmentation was applied to prevent overfitting. We used random scaling, rotation, cropping and flipping for all training samples in both networks. First, we randomly interpolated data so that the number of slices occupied by the liver ranged from 64 to 256 for both networks. Second, we randomly cropped the data. For the SC-SegNet, we cropped patches of size $256 \times 256 \times 256$ ($W = H = S = 256$) from data such that the number of slices between the center slice of the liver and the center slice of the patch was less than 8. If the number of slices of data was less than 256, we appended zeros on both sides. Since the shape autoencoder is easier to overfit, we augmented more samples to train the network. For the shape autoencoder, we cropped patches of size $256 \times 256 \times 256$ so that the ratio of liver slices covered by the patch was greater than 0.75. Finally, we randomly flipped samples with respect to the three axes and rotated samples 90, 180, 270 degrees in the axial.

C. Parameter Setting and Training Details

In the training stage, all hyperparameters were determined based on the validation set (4 CT images from the training set). We performed a 5-fold cross validation in our experiment. Once all the hyperparameters were determined, we used all the training data to retrain the networks. For both networks, the weights were initialized using the method proposed by He *et al.* [36]. The batch size was set to 1 because of memory limitation. The momentum was set to 0.9, and the weight decay was 0.0001. All networks were trained using stochastic gradient descent algorithm [37]. The shape autoencoders were trained with 150000 iterations, and the Sc-SegNets were trained with 30000 iterations. For the shape autoencoders, the initial learning rate was set to 1, and it was decreased to 0.0001 by a "poly" learning rate policy where the initial learning rate was multiplied by $(1 - \frac{\texttt{iteration}}{\texttt{max\_iteration}})^{\texttt{power}}$ with `power = 0.9` [38]. For the SC-SegNets, the initial learning rate was set to 0.1, and decreased to 0.0001 by the same policy used to train the shape autoencoders. The hyperparameter α was initialized to 100, and decreased linearly to 0 as the number of training steps increased. The hyperparameters β

and γ were set to 0.1 and 4, respectively. In order to avoid exploding gradients when training the networks, we applied "gradient scaling" to update the weights [39]. The experiments were conducted on a desktop computer with Intel Xeon E5-2686 CPU (2.30 GHz) and a graphics card (NVDIA TITAN V). The networks were implemented in C++ based on the deep learning library of cuda-convnet [40]. It took about fourteen hours to train each network.

D. Inference Schemes

In the test stage, each CT scan was first preprocessed. If the number of slices of the scan was bigger than 256, it was resampled to 256. Then, the SC-SegNet took the resampled data as input and outputted a coarse segmentation result. Since the number of slices and the position of liver varied widely, we calculated the initial position p_i and the final position p_f of the liver based on the coarse segmentation result. Subsequently, we resampled the data with scale $(p_f - p_i)/160$ in z axis and then cropped a patch such that the liver lied in the center along z axis. Following that, data augmentation was applied to each cropped patch by rotating 90, 180, 270 in the axial and flipping in three axes. Each augmented patch was then independently processed by the SC-SegNet. To obtain the segmentation result, we averaged the predictions computed from the augmented data, and resampled the segmentation result back to the original scale. To avoid isolated segments, a largest connected component labeling and hole filling were finally performed to refine the segmentation result. By implementing in C++ and using a GPU-based algorithm, the total processing time for a single scan depended on the number of slices, ranging from 7 to 10 seconds.

V. Results

A. Evaluation Metrics

According to previous studies in literature, it is not possible to define a single evaluation metric for the image segmentation problem. In the Sliver07 challenge, five different performance measures were computed, including the volumetric overlap error (VOE) in percent, the relative volume difference (RVD), the average symmetric surface distance (ASSD), the root mean square symmetric surface distance (RMSD) and the maximum symmetric surface distance (MSSD). Each error measure was translated to a score in the range from 0 (lowest possible score) to 100 (perfect result). Finally, the five scores were averaged to obtain one overall score per test case. In addition to these metrics, we also calculated the Dice similarity coefficient (DSC) for ecah scan. According to the evaluation of the CHAOS challenge, four evaluation metrics were utilized, including DSC, relative absolute volume difference (RAVD), ASSD, and MSSD. The results of these four metrics were converted to grades at 0-100 scale further, and combined into a final score.

B. Segmentation Results of the SC-SegNet

A total of 30 volumes from two different datasets were used to evaluate the performance of the SC-SegNet. The segmentations of 10 test scans from the Sliver07 challenge were evaluated by the organizers of the challenge. Fig. 6 shows some examples of our segmentation results on the validation set of the Sliver07 challenge in the axial, sagittal, and coronal plane. As can be seen, the SC-SegNet can deal well with the presence of pathologies and inhomogeneous appearances.

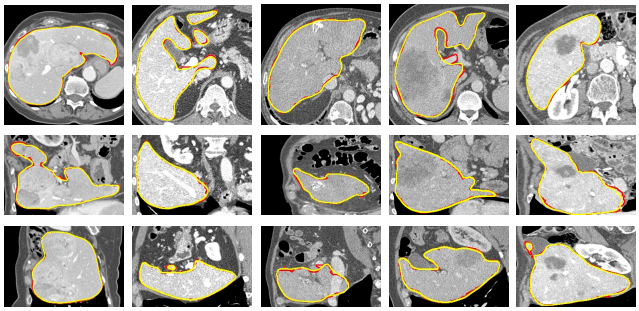

Fig. 6. Examples of segmentation results from the Sliver07 challenge. The three rows display the results generated by the SC-SegNet viewed in the axial plane, sagittal, and coronal plane, respectively. The red lines are the true segmentation results provided by the corresponding challenge organizers. The yellow lines are produced by the SC-SegNet.

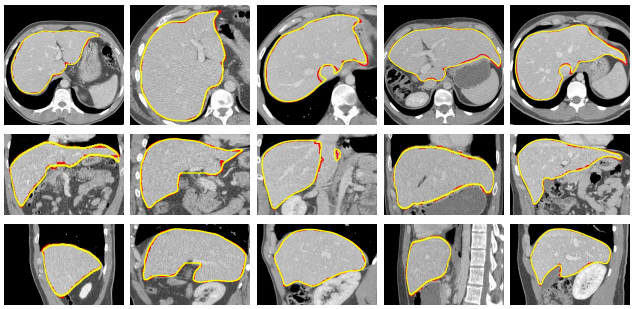

Fig. 7. Examples of segmentation results from the CHAOS challenge (the same settings as in Fig. 6).

The results of 20 scans from the CHAOS challenge were also evaluated by the organizers of the challenge. Fig. 7 shows several typical results on the validation set from the CHAOS challenge. We can observe that livers with various shapes can be well segmented.

C. Effectiveness of the Hybrid Loss

To validate the effectiveness of the hybrid loss, we compared the behaviors of the SC-SegNet and networks which have the same architecture but different loss functions. Table I shows the DSC scores of different configurations on the training data of the Sliver challenge and the CHAOS challenge by a 5-fold cross validation. It can be seen that combining all three parts of the loss can achieve the best average DSC scores of 97.71% and 97.48% on the two training datasets, which indicates the hybrid loss can improve the segmentation performance. We also compared the performance of networks with the same architecture but different loss functions, including the hybrid

TABLE I
COMPARISON OF NETWORKS WITH DIFFERENT CONFIGURATIONS ON THE TRAINING DATASETS OF THE SLIVER07 CHALLENGE AND THE CHAOS CHALLENGE BY A 5-FOLD CROSS VALIDATION. IN EACH TABLE, THE TWO NUMBERS ARE THE DSC SCORES IN THE CORRESPONDING FOLDS OF THE SLIVER CHALLENGE AND THE CHAOS CHALLENGE, RESPECTIVELY. THE BEST AVERAGE DSC SCORE IS MARKED IN BOLD.

(α, β, γ)	(0, 0, 0)	(100, 0, 0)	(0, 0.1, 0)	(0, 0, 4)	(100, 0.1, 0)	(100, 0, 4)	(0, 0.1, 4)	(100, 0.1, 4)
fold 1	97.74/96.74	97.65/97.04	97.81/96.62	97.73/96.55	97.71/96.93	97.77/97.03	97.82/96.82	97.78/97.15
fold 2	97.30/97.44	97.33/97.52	97.40/97.53	97.47/97.55	97.43/97.67	97.52/97.58	97.45/97.60	97.41/97.64
fold 3	97.47/96.96	97.65/97.22	97.43/97.09	97.72/97.26	97.67/97.22	97.73/97.45	97.70/97.37	97.82/97.43
fold 4	97.65/97.26	97.72/97.71	97.45/97.31	97.45/97.64	97.51/97.69	97.46/97.72	97.36/97.57	97.72/97.77
fold 5	97.06/97.10	97.60/97.29	97.08/97.24	97.19/97.32	97.65/97.29	97.67/97.24	97.38/97.34	97.81/97.43
Avg	97.44/97.10	97.59/97.36	97.43/97.16	97.51/97.27	97.59/97.36	97.63/97.41	97.54/97.34	**97.71/97.48**

TABLE II
COMPARISON OF NETWORKS WITH THE SAME ARCHITECTURE BUT DIFFERENT LOSS FUNCTIONS. IN EACH TABLE, THE TWO NUMBERS ARE THE DSC SCORES IN THE CORRESPONDING FOLDS OF THE SLIVER CHALLENGE AND THE CHAOS CHALLENGE, RESPECTIVELY. THE BEST AVERAGE DSC SCORE IS MARKED IN BOLD.

Loss	Cross Entropy	DICE Loss	Focal Loss	Our Loss
fold 1	97.74/96.74	97.26/95.99	97.67/96.39	97.78/97.15
fold 2	97.30/97.44	97.15/97.08	97.36/97.40	97.41/97.64
fold 3	97.47/96.96	97.54/97.23	97.40/97.16	97.82/97.43
fold 4	97.65/97.26	96.96/97.34	97.24/97.39	97.72/97.77
fold 5	97.06/97.10	96.77/97.04	96.88/97.22	97.81/97.43
Avg	97.44/97.10	97.13/96.93	97.31/97.11	**97.71/97.48**

loss, the cross-entropy loss (called SegNet), the dice loss [41], and the focal loss [42] in Table II. The networks with the cross-entropy loss were trained using the same learning rate as that used for the SC-SegNet, and were corresponding to the configuration of $(\alpha, \beta, \gamma) = (0, 0, 0)$. The learning rate was set to 0.1, and the momentum was set to 0.9 for the networks with the dice loss. We found that as the hyperparameter γ of the focal loss became larger, the performance deteriorated. Then, the networks with the focal loss were trained with a learning rate of 0.2 and momentum of 0.9, and the hyperparameters of α and γ were set to 0.5 and 0.25, respectively. We can see that our proposed hybrid loss outperformed the cross-entropy loss, the dice loss, and the focal loss with 0.27%/0.38%, 0.58%/0.55%, and 0.40%/0.37% improvement on DSC, respectively.

D. Effectiveness of CRF

Many CNN-based methods take further postprocessing steps to improve the segmentation results, such as CRF, graph cut, active contours, and surface evolution [27], [29]–[31], [43]. To verify the effectiveness of postprocessing steps, we compared the performance of the SC-SegNet (SegNet) with and without a fully connected CRF model [28]. The hyperparameters of CRF were optimized separately for the two challenges using their respective training scans.

VI. DISCUSSION

Accurate segmentation of liver is essential in clinical diagnosis. In this paper, we propose an automatic liver segmentation method based on 3D CNNs. A hybrid loss function consisting of three parts is used to better guide the learned features of network. To demonstrate the effectiveness of the loss, quantitative and qualitative comparisons are performed on two datasets.

Table I shows the detail results of different configurations of the loss. It proves that combining all three parts of the loss improves the segmentation performance the most. Table II compares the performance of different loss functions. It can be seen that our hybrid loss is more suitable for liver segmentation under the situation. We also find that the dice loss and the focal loss achieve no improvement compared to the basic cross-entropy loss. This may be because these loss functions are more suited for imbalanced segmentation. However, the most difficulties in liver segmentation are the highly varied liver shapes, fuzzy boundaries, and the presence of pathologies. Although the focal loss focuses on imbalanced segmentation and hard-sample mining, its performance is also limited when the training dataset is small.

VII. CONCLUSION

In this paper, a fully automatic liver segmentation method is presented. Rather than directly training the segmentation network with the cross-entropy loss, we propose to use a hybrid loss function. The hybrid loss consists of an adaptively weighted cross entropy, an edge-preserved smoothness loss, and a shape constraint.

ACKNOWLEDGMENT

The authors would like to thank organizers of the Sliver07 challenge and the CHAOS challenge for providing liver segmentation images and evaluating our system on the test images. This work was supported in part by the National Natural Science Foundation of China under Grant No. 91630311 and No. 11801511, in part by the Natural Science Foundation of Zhejiang Province under Grant No. LSD19H180005 and No. LQ20H180001, and in part by the Fundamental Research Funds for the Central Universities.

REFERENCES

[1] A. Gotra, L. Sivakumaran, G. Chartrand, K.-N. Vu, F. Vandenbroucke-Menu, C. Kauffmann, S. Kadoury, B. Gallix, J. A. de Guise, and A. Tang, "Liver segmentation: indications, techniques and future directions," Insights into imaging, vol. 8, no. 4, pp. 377–392, 2017.

[2] P. Campadelli, E. Casiraghi, and A. Esposito, "Liver segmentation from computed tomography scans: a survey and a new algorithm," Artificial intelligence in medicine, vol. 45, no. 2-3, pp. 185–196, 2009.

[3] A. M. Mharib, A. R. Ramli, S. Mashohor, and R. B. Mahmood, "Survey on liver ct image segmentation methods," Artificial Intelligence Review, vol. 37, no. 2, pp. 83–95, 2012.

[4] T. Heimann, B. Van Ginneken, M. A. Styner, Y. Arzhaeva, V. Aurich, C. Bauer, A. Beck, C. Becker, R. Beichel, G. Bekes et al., "Comparison and evaluation of methods for liver segmentation from ct datasets," IEEE transactions on medical imaging, vol. 28, no. 8, pp. 1251–1265, 2009.

[5] S. Luo, X. Li, and J. Li, "Review on the methods of automatic liver segmentation from abdominal images," Journal of Computer and Communications, vol. 2, no. 02, p. 1, 2014.

[6] R. Adams and L. Bischof, "Seeded region growing," IEEE Transactions on pattern analysis and machine intelligence, vol. 16, no. 6, pp. 641–647, 1994.

[7] R. Pohle and K. D. Toennies, "Segmentation of medical images using adaptive region growing," in Medical Imaging 2001: Image Processing, vol. 4322. International Society for Optics and Photonics, 2001, pp. 1337–1346.

[8] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," International journal of computer vision, vol. 1, no. 4, pp. 321–331, 1988.

[9] K. Suzuki, R. Kohlbrenner, M. L. Epstein, A. M. Obajuluwa, J. Xu, and M. Hori, "Computer-aided measurement of liver volumes in ct by means of geodesic active contour segmentation coupled with level-set algorithms," Medical Physics, vol. 37, no. 5, pp. 2159–2166, 2010. [Online]. Available: https://aapm.onlinelibrary.wiley.com/doi/abs/10.1118/1.3395579

[10] Y. Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in nd images," in Proceedings eighth IEEE international conference on computer vision. ICCV 2001, vol. 1. IEEE, 2001, pp. 105–112.

[11] H. Yang, Y. Wang, J. Yang, and Y. Liu, "A novel graph cuts based liver segmentation method," in 2010 International Conference of Medical Image Analysis and Clinical Application. IEEE, 2010, pp. 50–53.

[12] B. N. Li, C. K. Chui, S. Chang, and S. H. Ong, "Integrating spatial fuzzy clustering with level set methods for automated medical image segmentation," Computers in biology and medicine, vol. 41, no. 1, pp. 1–10, 2011.

[13] Y. Zhao, Y. Zan, X. Wang, and G. Li, "Fuzzy c-means clustering-based multilayer perceptron neural network for liver ct images automatic segmentation," in 2010 Chinese Control and Decision Conference. IEEE, 2010, pp. 3423–3427.

[14] L. Rusko, G. Bekes, G. Nemeth, and M. Fidrich, "Fully automatic liver segmentation for contrast-enhanced ct images," MICCAI Wshp. 3D Segmentation in the Clinic: A Grand Challenge, vol. 2, no. 7, 2007.

[15] S.-J. Lim, Y.-Y. Jeong, and Y.-S. Ho, "Automatic liver segmentation for volume measurement in ct images," Journal of Visual Communication and Image Representation, vol. 17, no. 4, pp. 860–875, 2006.

[16] L. Massoptier and S. Casciaro, "Fully automatic liver segmentation through graph-cut technique," in 2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE, 2007, pp. 5243–5246.

[17] D. Kainmüller, T. Lange, and H. Lamecker, "Shape constrained automatic segmentation of the liver based on a heuristic intensity model," in Proc. MICCAI Workshop 3D Segmentation in the Clinic: A Grand Challenge, 2007, pp. 109–116.

[18] M. Erdt, S. Steger, M. Kirschner, and S. Wesarg, "Fast automatic liver segmentation combining learned shape priors with observed shape deviation," in 2010 IEEE 23rd International Symposium on Computer-Based Medical Systems (CBMS). IEEE, 2010, pp. 249–254.

[19] E. van Rikxoort, Y. Arzhaeva, and B. van Ginneken, "Automatic segmentation of the liver in computed tomography scans with voxel classification and atlas matching," in Proceedings of the MICCAI Workshop, vol. 3. Citeseer, 2007, pp. 101–108.

[20] S. Luo, Q. Hu, X. He, J. Li, J. S. Jin, and M. Park, "Automatic liver parenchyma segmentation from abdominal ct images using support vector machines," in 2009 ICME International Conference on Complex Medical Engineering. IEEE, 2009, pp. 1–5.

[21] H. Ling, S. K. Zhou, Y. Zheng, B. Georgescu, M. Suehling, and D. Comaniciu, "Hierarchical, learning-based automatic liver segmentation," in 2008 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2008, pp. 1–8.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in International Conference on Neural Information Processing Systems, 2012, pp. 1097–1105.

[23] R. Girshick, "Fast r-cnn," in IEEE International Conference on Computer Vision, 2015, pp. 1440–1448.

[24] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3431–3440.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[26] G. Huang, Z. Liu, L. v. d. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017, pp. 2261–2269.

[27] Q. Dou, H. Chen, Y. Jin, L. Yu, J. Qin, and P.-A. Heng, "3d deeply supervised network for automatic liver segmentation from ct volumes," in International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, 2016, pp. 149–157.

[28] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," in Advances in neural information processing systems, 2011, pp. 109–117.

[29] P. Hu, F. Wu, J. Peng, P. Liang, and D. Kong, "Automatic 3d liver segmentation based on deep learning and globally optimized surface evolution," Physics in Medicine & Biology, vol. 61, no. 24, p. 8676, 2016.

[30] F. Lu, F. Wu, P. Hu, Z. Peng, and D. Kong, "Automatic 3d liver location and segmentation via convolutional neural network and graph cut," International journal of computer assisted radiology and surgery, vol. 12, no. 2, pp. 171–182, 2017.

[31] K. H. Cha, L. Hadjiiski, R. K. Samala, H.-P. Chan, E. M. Caoili, and R. H. Cohan, "Urinary bladder segmentation in ct urography using deep-learning convolutional neural network and level sets," Medical physics, vol. 43, no. 4, pp. 1882–1896, 2016.

[32] A. E. Kavur, M. A. Selver, O. Dicle, M. Barış, and N. S. Gezer, "CHAOS - Combined (CT-MR) Healthy Abdominal Organ Segmentation Challenge Data," Apr. 2019. [Online]. Available: https://doi.org/10.5281/zenodo.3431873

[33] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in Proceedings of the 32nd International Conference on Machine Learning, vol. 37, Jul 2015, pp. 448–456.

[34] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in International Conference on Medical image computing and computer-assisted intervention. Springer, 2015, pp. 234–241.

[35] P. F. Felzenszwalb, R. B. Girshick, D. Mcallester, and D. Ramanan, "Object detection with discriminatively trained part-based models." IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 9, pp. 1627–1645, 2010.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in 2015 IEEE International Conference on Computer Vision (ICCV), Dec 2015, pp. 1026–1034.

[37] G. E. Hinton, S. Osindero, and Y. W. Teh, "A fast learning algorithm for deep belief nets," Neural Computation, vol. 18, no. 7, pp. 1527–1554, 2006.

[38] W. Liu, A. Rabinovich, and A. C. Berg, "Parsenet: Looking wider to see better," CoRR, vol. abs/1506.04579, 2015. [Online]. Available: http://arxiv.org/abs/1506.04579

[39] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in International conference on machine learning, 2013, pp. 1310–1318.

[40] A. Krizhevskey, "Cuda-convnet," 2014. [Online]. Available: http://code.google.com/p/cuda-convnet.

[41] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in 2016 Fourth International Conference on 3D Vision (3DV). IEEE, 2016, pp. 565–571.

[42] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in Proceedings of the IEEE international conference on computer vision, 2017, pp. 2980–2988.

[43] X. Guo, L. H. Schwartz, and B. Zhao, "Automatic liver segmentation by integrating fully convolutional networks into active contour models," Medical Physics, vol. 46, no. 10, pp. 4455–4469, 2019. [Online]. Available: https://aapm.onlinelibrary.wiley.com/doi/abs/10.1002/mp.13735