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 used cross entropy loss treats each pixel independently and equally, which makes the network sensitive to fuzzy boundaries and heterogeneous pathologies. To address these issues, 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 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. Finally, a conditional random field model is used to refine the segmentation results. 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 structure difference.**

I. Introduction

ACCURATE liver segmentation from three dimensional (3D) computed tomography (CT) is critical in many clinical applications, such as radiotherapy planning and post-operative 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, automatic liver segmentation is highly demanded [2].

Automatic liver segmentation from CT images is a very challenging task due to the wide variety of liver shapes, fuzzy boundaries, and the presence of various pathologies and high-intensity intrahepatic veins. 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.* [3] and Mharib *et al.* [4]. Heimann *et al.* [5] also presented a detailed comparison study between 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 [6].

Gray level intensity is the most obvious feature of CT images. Many gray level based segmentation algorithms are developed, including region growing methods [7], [8], active contours [9], [10], graph cuts [11], [12], clustering based algorithms [13], [14] and so on. For example, Rusko *et al.* [15] first determined a seed region based on intensity histogram and separated the heart from the liver to eliminate over-segmented regions. Finally, they employed an advanced region growing method to segment the liver region followed by various postprocessing steps to prevent under-segmentation. Lim *et al.* [16] extracted the initial liver volume by exploiting prior information about the location of liver and the distribution of liver intensity 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 contour. Finally, contour-based segmentation using the labeling-based search algorithm was applied to find the final liver contour. Massoptier *et al.* [17] 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.* [14] 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, such as: 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.* [18] 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.* [19] presented

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).

a fully automatic multi-tiered statistical shape model for the liver that combined learned local shape constraints with observed shape deviation during adaptation. Van *et al.* [20] 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, a major challenge that these methods need to address is modeling the large shape variations with limited training data.

In texture based segmentations, handcrafted features are extracted first and trained classification models are then employed to label unseen images. Luo *et al.* [21] 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. Finally, integrated morphological operations were designed to remove noise and delineate the liver. Ling *et al.* [22] 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 [23], [24], [25], [26], [27], many studies have used the representative features learned by CNNs to deal with the segmentation of liver. Dou *et al.* [28] presented a novel 3D deeply supervised fully convolutional network for automatic liver segmentation. They further employed a fully connected conditional random field (CRF) [29] 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 [5]. Hu *et al.* [30] 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 [2], CRF [28], 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 three major novelties in the proposed framework:

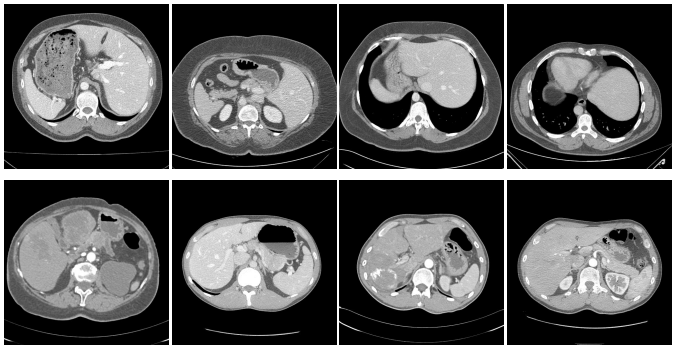

Fig. 1. Examples of contrast-enhanced CT images illustrating the challenges for accurate liver segmentation. Each row shows the examples form the CHAOS challenge, and the SLIVER07 challenge, respectively.

- We use an adaptively weighted loss to pay more attention to pixels that are difficult to classify, and design an edge-preserved smoothness loss as a regularizer to constrain neighbouring pixels with the same label to have similar outputs while neighbouring pixels with different labels to have dissimilar outputs.
- Our system also introduces a high-level loss for the output of the middle layer of network, which is a shape constraint that better guides the learned features for segmentation.
- We validated the proposed method on two separate clinical databases.

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, a total of 70 abdominal CT scans from two datasets were used for model training (40) and testing (30). 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 contains CT images of 40 different patients, who have healthy liver. 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 [5]. 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 the Fig. 2. Two networks are incorporated: a liver shape autoencoder and a 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 next sections, we 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 [5]. As thus, we use an autoencoder to model the liver shape.

The shape autoencoder is designed based on the structure of DenseNet [27], which uses densely connected blocks (dense blocks) to ensure maximum information flow between layers. The dense block architecture 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. This construction of dense block has several advantages: alleviating the vanishing-gradient problem, strengthening feature propagation, and encouraging feature reuse.

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 followed by 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 [25]. 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 belongs to liver. Except for the first convolutional layer is with a stride of 2, all other convolutional layers are with the same 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 in the liver mask y, y_i^{SA} denotes the output probability of pixel i belongs to liver from the shape autoencoder, and n denotes the total number of pixels in y.

B. SC-SegNet

To segment liver from raw CT images, we develop a shape-constrained densely connected segmentation network (SC-SegNet). The architecture of SC-SegNet illustrated in Fig. 2(a) is designed based on FCN [25], DenseNet [27], 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 SC-SegNet, other architecture settings are same as the shape autoencoder.

Let $x \in R^{W \times H \times S}$ denote a input training sample with a ground-truth mask $y \in \{0,1\}^{W \times H \times S}$, f_{le} denote the encoder of SC-SegNet, and f_{ls} denote the 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, and the sensitivity to fuzzy boundaries and heterogeneous pathologies. To address these issues, a loss function composed of four parts is designed to train the SC-SegNet. The first part of loss is the typical cross entropy loss,

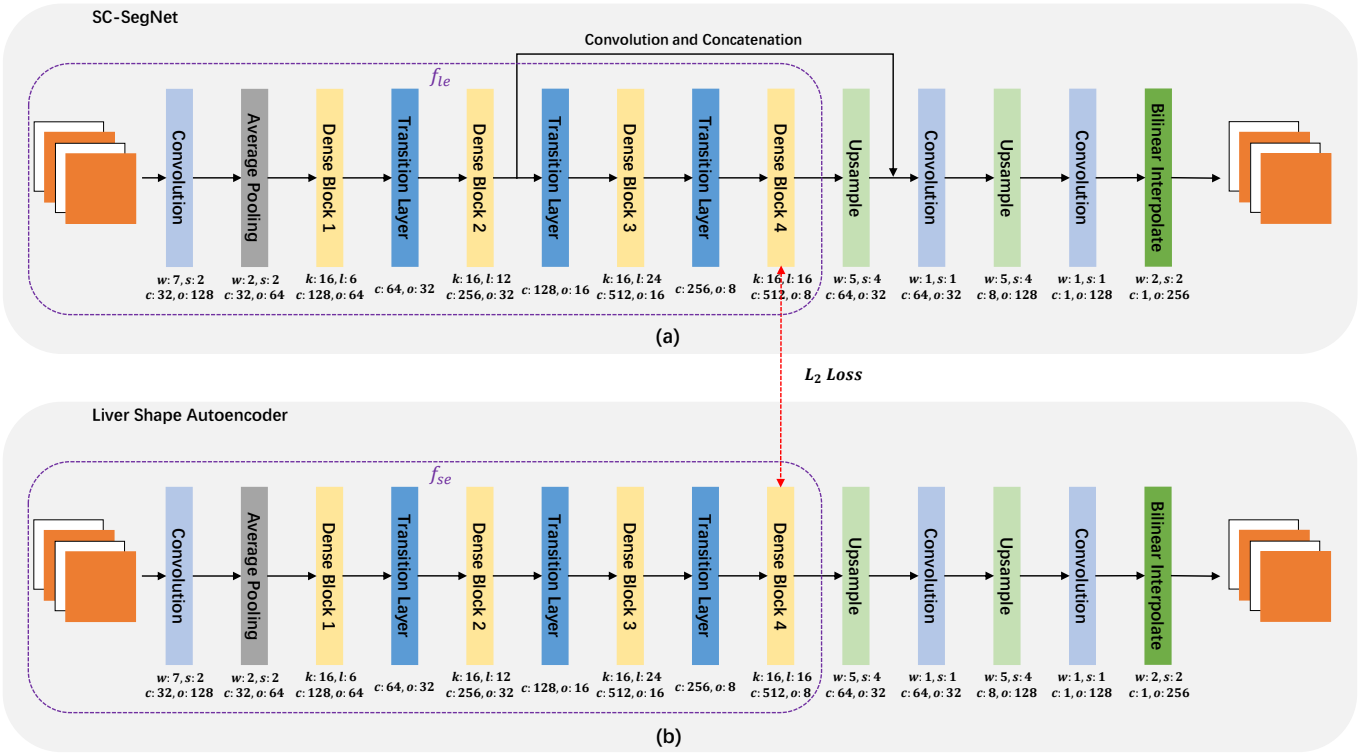

Fig. 2. The framework of the proposed liver segmentation method. The shape autoencoder is first trained to obtain the liver shape codes with the liver masks as input. Then, the SC-SegNet is trained under the supervision of both the liver masks and the shape codes from the trained shape autoencoder. In 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.

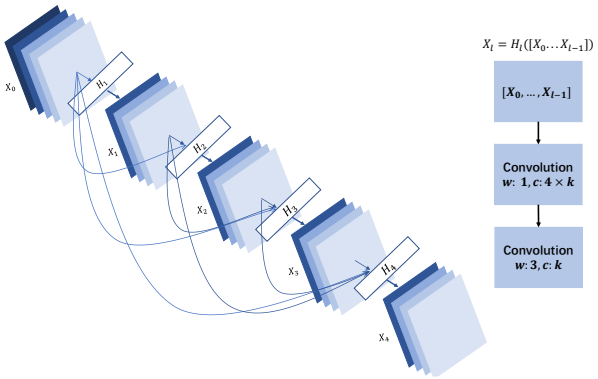

Fig. 3. A 4-layer dense block. The input has 5 feature maps, and each layer in the block produces 4 feature maps. Finally, the block outputs 21 feature maps (the same symbol definition as in Fig. 2).

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

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 $f_{se}(y)$,

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

To put it simply, we try to make the features learned by the SC-SegNet consistent with the shape coding 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

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

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. The w_{ij} is defined in terms of pixel intensities I_i and I_j, labels y_i and y_j, and network output $f_{ls}(x)_j$:

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

In the above function, $\mathbf{1}_A$ 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. Due to the first indicator, 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,

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

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

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

where α, β, γ are hyperparameters used to balance these four parts of loss.

C. Refinement with CRF

Although the SC-SegNet can generate high-quality probability maps, directly using a threshold-based method to obtain liver boundaries can lead to incorrect segmentations. In this section, we consider using a fully connected CRF model [29] to refine the segmentation results. To segment images, the model minimizes the energy function:

$$E(y|x) = \sum_{i} \psi_u(y_i) + \sum_{i<j} \psi_p(y_i, y_j). \tag{11}$$

The unary potential $\psi_u(y_i)$ is computed independently for each pixel by a classifier that produces a distribution over the label assignment y_i. In our experiment, the unary potential is defined as $\psi_u(y_i) = -\log(p(y_i))$, where $p(y_i)$ is produced by the SC-SegNet. The pairwise potential $\psi_p(y_i, y_j)$ introduces a penalty for nearby similar pixels that are assigned different labels. Compared to other CRF-based methods, the model establishes pairwise potentials on all pairs of pixels in the image. It is defined as $\psi_p(y_i, y_j) = \mathbf{1}_{y_i \neq y_j} \{ w^{(1)} \exp(-\frac{|p_i - p_j|^2}{2\theta_\alpha^2} - \frac{|I_i - I_j|^2}{2\theta_\beta^2}) + w^{(2)} \exp(-\frac{|p_i - p_j|^2}{2\theta_\gamma^2}) \}$, where p_i is the position of pixel i, $w^{(1)}$, $w^{(2)}$, θ_α, θ_β, and θ_γ are five parameters.

IV. Experimental Setting

In this section, we present the details of our experimental setup.

A. Preprocessing

We applied image preprocessing, including several steps. First, the pixel intensity range was normalized from (-250,250 Houndsfield Unit) to (0,1). Intensity > 250 was set to 1, and < -250 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 and the slice number remained unchanging. All the preprocessed steps were applied to both training and test datasets. In the test stage, segmentation results were resampled back to the original scale (512×512).

B. Data Augmentation

In our experiments, data augmentation was applied to prevent overfitting. We adopted random rotation, cropping and flipping for all training samples in both networks. First, we randomly resampled 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 raw CT images such that the distance between the center slice of the liver and the center slice of the patch was less than 8. If the slice number was less than 256, we appended zeros on both sides. Because the shape autoencoder is easier to overfit, we augmented more samples to train the network. For the shape autoencoder, we cropped patches of size $256 \times 256 \times 256$ such that the proportion of slices of liver covered by the patch was more than 3/4. Last, we randomly flipped samples with respect to the three axes and rotated samples 90, 180, 270 degrees in the axial.

C. Training Details and Parameter Setting

In the training stage, all hyperparameters were determined based on the validation set (8 CT images from the training set). Once all the hyperparameters were determined, we used all the data to retrain the networks. For both networks, the convolutional weights and upsampling weights were initialized using the method proposed by He *et al.* [36]. The biases were initialized with zeros. 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. Both networks were trained with 60000 iterations using stochastic gradient descent algorithm [37]. For the shape autoencoder, 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-SegNet, the initial learning rate was set to 0.1, and decreased to 0.00001 by the same policy used in the shape autoencoder. 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 both networks, we applied "gradient scaling" to update the weights [39]. The parameters of the CRF model were optimized using a grid search on the training set. 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 and resampled to have a resolution of 256×256 in the axial. If the number of slices of the scan was bigger than 256, we resampled it to 256. Then, SC-SegNet outputed the coarse segmentation result with the resampled data as input. 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 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 performed on each cropped patch by rotating 90, 180, 270 in the axial and flipping in three axes. And then, each augmented patch was independently processed by SC-SegNet. To get the segmentation results, we averaged the predictions computed from the augmented data and the CRF model was then applied based on the probability maps. Finally, we resampled the segmentation results back to the original scale. To avoid isolated segments, a largest connected component labeling was performed to refine the segmentation results. By implementing in C++ with parallelization and using a GPU-based algorithm, the total processing time of a single scan depended on the number of slices, ranging from 0.7; minutes to 3 minutes (about 8.5 seconds for SC-SegNet, the rest for postprocessing).

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, RVD, ASSD, and MSSD. The results of these four metrics were converted to grades at 0-100 scale further, and combined into a final score.

TABLE I
Evaluation of SC-SegNet on the SLIVER07 challenge (10 volumes).

Metric	VOE (%)	RVD (%)	ASSD (mm)	RMSD (mm)	MSSD (mm)	DSC (%)	Score
case 1	4.86	2.22	0.79	1.61	17.65	97.51	80.8
case 2	5.52	4.57	0.86	1.98	25.78	97.16	74.2
case 3	8.90	-5.03	1.68	4.70	36.87	95.34	56.5
case 4	5.05	0.82	0.80	1.60	12.86	97.41	83.4
case 5	4.52	1.63	0.73	1.52	19.66	97.69	81.7
case 6	4.95	-0.21	0.77	1.73	18.03	97.46	82.5
case 7	3.74	2.81	0.53	1.00	15.58	98.09	84.6
case 8	4.37	0.91	0.67	1.21	13.31	97.77	84.3
case 9	4.96	2.56	0.66	1.49	20.33	97.46	80.6
case 10	5.70	-2.34	0.79	1.45	10.73	97.07	82.3
Avg	5.26	0.90	0.83	1.83	19.08	97.30	79.09
Std	1.40	2.79	0.31	1.04	7.60	0.75	8.46

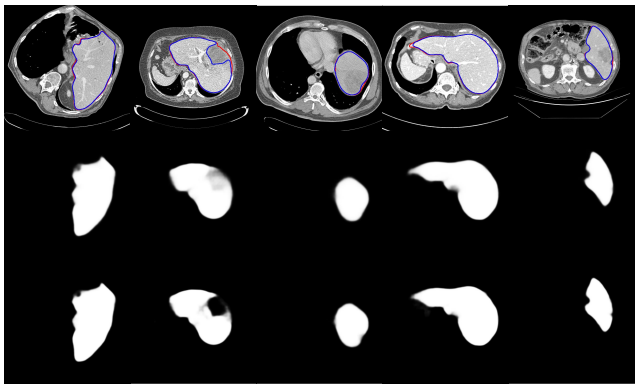

Fig. 4. Examples of segmentation results on the SLIVER07 challenge. The first row shows the segmentation results in the axial based on the probability maps with a threshold of 0.5. The red lines delineate the results obtained from SC-SegNet. The blue lines delineate the results obtained form a network with the same architecture but using cross entropy as the loss (SegNet). The probability maps in the second row are produced by the SC-SegNet. The probability maps in the last row are produced by the SegNet.

B. Segmentation Results of SC-SegNet

A total of 30 volumes from two different datasets were used to evaluate the final segmentations of the SC-SegNet. The segmentations of 10 test scans from the SLIVER07 challenge were evaluated by the organizers of the SLIVER07 website. Table I summarizes the detail results of each scan with a total score of 79.09 ± 8.46. The final mean results of VOE, RVD, ASSD, RMSD, and MSSD are $5.26 \pm 1.40\%$, $0.90 \pm 2.79\%$, 0.83 ± 0.31 mm, 1.83 ± 1.04 mm, and 19.08 ± 7.60 mm, respectively. Fig. 4 shows some examples of our segmentation results in the axial plane. As can be seen, the SC-SegNet can deal well with the presence of pathologies and inhomogeneous appearances.

The results of 20 scans from the CHAOS challenge were also evaluated by the organizers. Table II describes the final result of each metric and the total score. The total score reaches an average of 81.64 ± 4.77. The mean results of DSC, RVD, ASSD, and MSSD are $97.45 \pm 0.36\%$, $1.89 \pm 0.96\%$, 0.87 ± 0.15 mm, and 16.44 ± 4.75 mm, respectively. Fig. 5

TABLE II
EVALUATION OF SC-SEGNET ON THE CHAOS CHALLENGE (20 VOLUMES).

Metric	DSC(%)	ASSD(mm)	RVD(%)	MSSD(mm)	Score
Avg	97.45	0.87	1.89	16.44	81.64
Std	0.36	0.15	0.96	4.75	4.77

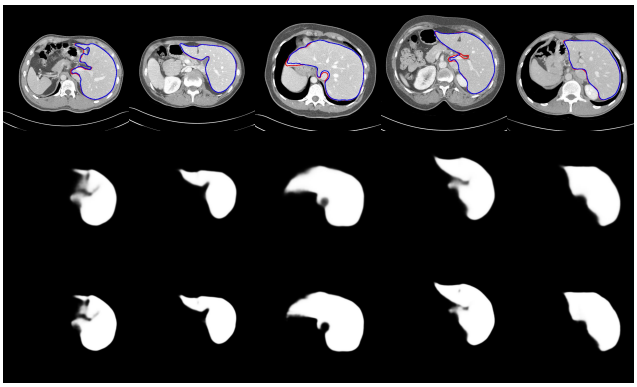

Fig. 5. Examples of segmentation results on the CHAOS challenge (the same settings as in Fig. 4).

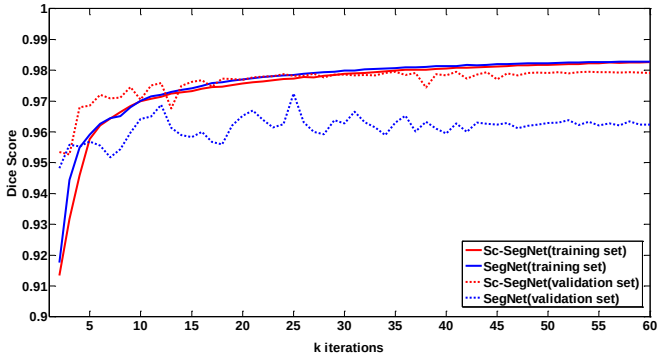

Fig. 6. DSC scores of SC-SegNet and the network which has the same architecture but uses the cross entropy as the loss. Note that, data augmentation was performed on the validation set.

shows several typical results on the CHAOS challenge.

C. Effectiveness of the Hybrid Loss

To validate the effectiveness of the hybrid loss, we compare the behaviors of the SC-SegNet and the network which has the same architecture but uses cross entropy as the loss (called SegNet). Fig. 6 shows the DSC scores on both the training set and the validation set of our proposed SC-SegNet and SegNet. It can be seen that both networks can be trained well, but the SC-SegNet performed much better on the validation set. Table III and Table IV show the quantitative comparisons of SC-SegNet and SegNet on two datasets. It can be seen that SC-SegNet achieves better results on both datasets. Qualitative comparisons are shown in Fig. 4 and Fig. 5. The visual comparison of both networks also illustrates the effectiveness of the hybrid loss.

TABLE III
COMPARISON OF SC-SEGNET AND SEGNET ON THE SLIVER07 CHALLENGE.

Metric	VOE	RVD	ASSD	RMSD	MSSD	Score
Avg(SC-SegNet)	5.26	0.90	0.83	1.83	19.08	79.09
Std(SC-SegNet)	1.40	2.79	0.31	1.04	7.60	8.46
Avg(SegNet)	6.08	-1.17	0.95	2.11	19.96	76.67
Std(SegNet)	3.20	4.28	0.62	1.79	11.45	16.96

TABLE IV
COMPARISON OF SC-SEGNET AND SEGNET ON THE CHAOS CHALLENGE.

Metric	DSC	ASSD	RVD	MSSD	Score
Avg(SC-SegNet)	97.45	0.87	1.89	16.44	81.64
Std(SC-SegNet)	0.36	0.15	0.96	4.75	4.77
Avg(SegNet)	97.13	0.98	2.84	17.37	76.26
Std(SegNet)	0.41	0.17	1.35	5.92	6.96

D. Effectiveness of CRF

E. Comparison with State-of-the-art Automatic Methods

VI. DISCUSSION

Accurate segmentation of liver is essential in clinical diagnosis. Most segmentation errors are due to pathological tissues and fuzzy boundaries. 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. On the SLIVER07 challenge, it achieved an improvement of 2.46 in the final score. And on the CHAOS challenge, it achieved an improvement of 5.38. The results on both datasets validate that the proposed loss is effective.

VII. CONCLUSION

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.

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] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

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

[8] 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.

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

[10] 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

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

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

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[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.