BLU-Net: A neural network for abdominal multi-organ segmentation based on bidirectional convolution LSTM and 3D U-Net

Weisheng Li[1], Hongchuan Zhang[1]

[1]Chongqing Key Laboratory of Image Cognition, Chongqing University of Posts and Telecommunications, Chongqing, China
liws@cqupt.edu.cn

Abstract. Abdominal organ segmentation plays an important role in clinical diagnosis and follow-up treatment. With the successful application of current deep learning methods in western image segmentation tasks, the segmentation accuracy of various abdominal organs has made significant progress. However, the current methods often only target one organ, and are accompanied by the consumption of a large amount of computing resources. Moreover, medical image data is very precious. The images to be segmented may be collected from different hospitals and different medical equipment, which leads to these images often have large heterogeneity, which tests the robustness of the model. Therefore, it is of great significance to propose a robust and lightweight method that can segment multiple abdominal organs at once. Based on Long short-term memory (LSTM) with convolution and 3D U-Net, we use the training set of FLARE21 to develop a lightweight abdominal multi-organ segmentation network.

Keywords: organ segmentation, heterogeneity, LSTM

1. Introduction

Multi-organ, multi-site segmentation is a very challenging segmentation task. For CT images, for the segmentation of a single organ, the preprocessing operation on the HU value can be performed first, but if multiple organs are to be segmented at the same time, the upper and lower bounds of the HU value have to be enlarged. Moreover, different organs have great differences in size and shape. In contrast, the volume of the pancreas is much smaller than that of the liver, so the training of the pancreas will face the challenge of serious unevenness of positive and negative samples. Moreover, the images acquired by different sites are also quite heterogeneous, and the inconsistencies in the number of training image slices and the changes in intensity differences will make it difficult for the network to learn a robust representation.

In the field of medical image segmentation, U-Net [1] has become a commonly used baseline with its classic encoding-decoding structure and good performance. But for volume images, U-Net can only convert them into slices and then perform semantic segmentation, which will lose valuable spatial information. In order to make up for this shortcoming, 3D U-Net [12] was proposed. Due to the low contrast of the soft tissue between the liver and its surrounding organs and the highly deformable shape, fully automated segmentation of the liver is challenging, there are several advanced liver segmentation methods [13-15]. The main challenge of kidney segmentation comes from the uneven intensity, there are also some excellent solutions for kidney segmentation [16-17]. Spleen segmentation is useful for not only measuring tissue volume and biomarkers but also for monitoring interventions [18]. The volume of the pancreas is very small, and the shape of the head and tail changes significantly, some existing pancreas segmentation methods are also trying to achieve better performance [19, 20].

We propose an automatic segmentation method called BLU-Net. It uses 3D U-Net as the baseline and uses bidirectional convolution LSTM block to assist the encoder to better learn contextual features.

2. Method

We use 3D U-Net [12] as a baseline to construct our method, as shown in Figure 1. The number of channels in each layer of the network is marked in the figure . Specifically, we reconstructed its encoder and connected the skip connection feature to the middle of the first and second convolutional layers of the decoder.

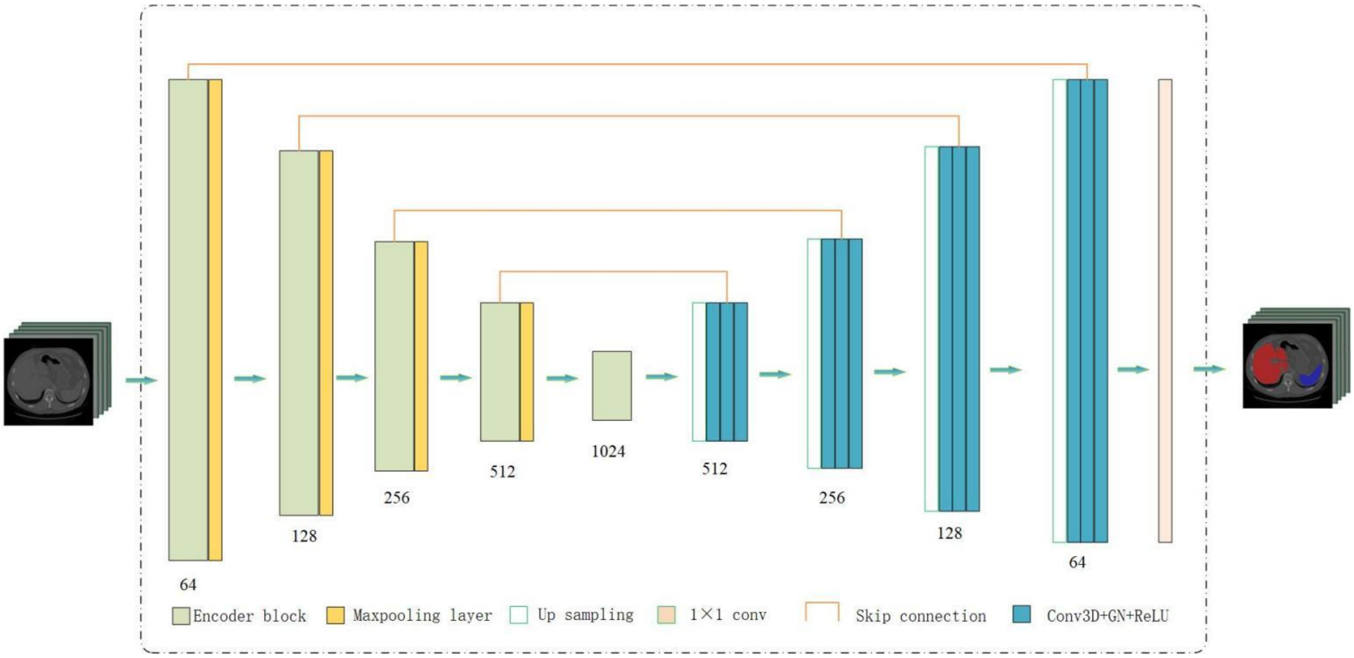

Figure 1. Network architecture. Where GN represents the GroupNorm layer in pytorch.

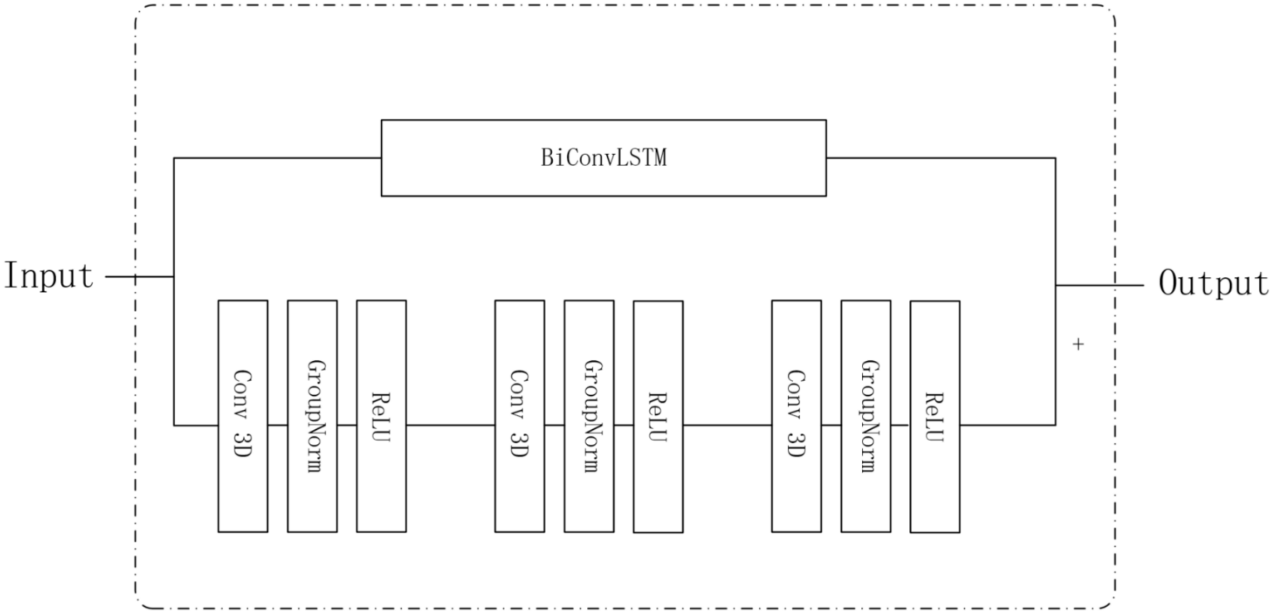

Figure 2. The network structure of the encoder block.

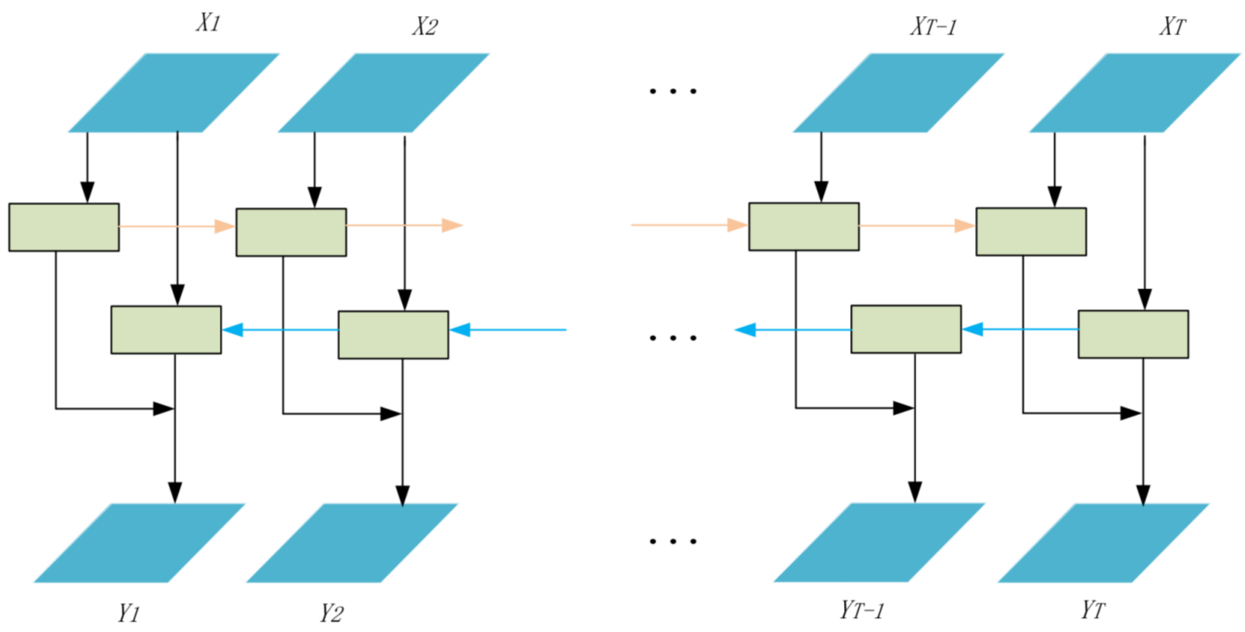

Figure 3. The network structure of BiConvLSTM.

2.1. Preprocessing

The baseline method includes the following preprocessing steps:

In the training phase, we crop the image according to the label, and remove the slices that are all background. Then every 5 slices are made into a patch. For the obtained patches, we use bilinear interpolation to down sample their size from 5*512*512 to 5*256*256 and normalize each patch so that the mean is 0 and the variance is 1.

In the inference stage, after reading the volume data, the data is divided into a patch form every 5 slices, and their size is reduced to 5×256×256 by bilinear interpolation first, and then the same as the training phase, normalization them, and then send them to the trained network one by one to get the segmentation result of each patch.

2.2. Proposed Method

We propose an end-to-end segmentation method based on 3Dunet, which consists of four up and down sampling. Compared with the baseline method, we have improved its encoder part, as shown in Figure 2. The encoder is composed of two learning paths. We use the bidirectional convolution LSTM block (BiConvLSTM) and the conventional 3D convolution module to extract features, and combine them by linear addition. We think that compared to a single learning path , The proposed method can better adapt to changes in image styles, and improve the robustness of the network to adapt to the heterogeneity of scanned data from different sites.

We believe that a 3D medical scan image can be viewed as multiple continuous top view slices, so it can be treated as a continuous video frame. For a volume input X $\in$ (T, W, H), we transform it into {X1, X2, …, XT}, where Xi (i $\in$1,2…T) is regarded as a frame and its size is (W, H). The LSTM module can establish a connection between the frames of the video, and our medical images also contain rich context, and the information of adjacent frames can assist the network in better segmentation tasks. In order to allow the characteristics of LSTM to be applied to image segmentation tasks, the convolution LSTM block (ConvLSTM) [10] was proposed. ConvLSTM unit consists of a storage unit c_t, an input gate i_t, an output gate o_t, and a forget gate f_t. ConvLSTM can be described as:

$$\begin{aligned}
&\mathrm{i}_t = \sigma(W_i^X * X_t + W_i^H * H_{t-1}),\\
&f_t = \sigma(W_f^X * X_t + W_f^H * H_{t-1}),\\
&o_t = \sigma(W_o^X * X_t + W_o^H * H_{t-1}),\\
&c_t = f_t \circ c_{t-1} + i_t \circ \tanh(W_c^X * X_t + W_c^H * \mathrm{H}_{t-1}),\\
&H_t = o_t \circ \tanh(c_t)
\end{aligned} \tag{2.1}$$

Where $*$ denotes the convolution layer and $\circ$ denotes the Hadamard product. All the gates i_t, f_t, o_t, memory cell c_t, hidden state H and the learnable weights W are 3D tensors.

The frame behind ConvLSTM can perceive the information of the previous frame, but not vice versa, and we hope that they can perceive each other, so we introduced the BiConvLSTM [13] module, which is used to capture two-way time characteristics, as shown in Figure 3. It can be described as:

$$Y_t = \tanh(W_y^{H^f} * H_t^f + W_y^{H^b} * H_{t-1}^b) \tag{2.2}$$

Where H^f and H^b indicates the hidden states from forward and backward ConvLSTM units, and Y_t indicates the final output considering bidirectional spatiotemporal information.

We use the summation between Dice loss and binary cross entropy loss as the loss function of the entire network. The total number of parameters of the network is 67,458,739 and its flops is 968704000. It occupies less than 8 G of GPU memory in the training phase. In the stable stage of inference, it is observed that its GPU memory occupancy is only 2887M. So we think this is a lightweight medical image segmentation method.

2.3. Post-processing

The output of our network is a one-hot encoding segmentation probability map. First apply bilinear interpolation to up sample the probability map of 5×256×256 to 5×512×512 , then apply the sigmoid function to map the generated probability value between 0-1, and then use a threshold to determine whether each pixel belongs to a certain category, here we set the threshold to 0.5, finally, decode the one-hot encoding. In this way, we get the segmentation result of a patch. For a subject, we stack these patches to get the overall result.

For the generated results, we first calculate the predicted connectivity components of the liver, spleen, and pancreas, and then only retain the largest connectivity component to remove the subtle noise area, because the kidney itself has more than one connected component, it will not be processed. It should be noted that for images with a large number of slices, the post-processing process is very wasteful. In order to prevent timeout, we only apply this post-processing to subjects with less than 100 slices.

3. Dataset and Evaluation Metrics

3.1. Dataset

- A short description of the dataset used:
  The dataset used of FLARE2021 is adapted from MSD [2] (Liver [3], Spleen, Pancreas), NIH Pancreas [4, 5, 6], KiTS [7, 8], and Nanjing University under the license permission. For more detail information of the dataset, please refer to the challenge website and [9].

- Details of training / validation / testing splits:
  The total number of cases is 511. An approximate 70%/10%/20% train/validation/testing split is employed resulting in 361 training cases, 50 validation cases, and 100 testing cases. The detail information is presented in Table 1.

Table 1. Data splits of FLARE2021.

Data Split	Center	Phase	#Num.
Training (361 cases)	The National Institutes of Health Clinical Center	portal venous phase	80
	Memorial Sloan Kettering Cancer Center	portal venous phase	281
Validation (50 cases)	Memorial Sloan Kettering Cancer Center	portal venous phase	5
	University of Minnesota	late arterial phase	25
	7 Medical Centers	various phases	20
Testing (100 cases)	Memorial Sloan Kettering Cancer Center	portal venous phase	5
	University of Minnesota	late arterial phase	25
	7 Medical Centers	various phases	20
	Nanjing University	various phases	50

3.2. Evaluation Metrics

- Dice Similarity Coefficient (DSC)

- Normalized Surface Distance (NSD)

- Running time

- Maximum used GPU memory (when the inference is stable)

4. Implementation Details

4.1. Environments and requirements

We trained our model on a Windows workstation, and then made docker on another Ubuntu server. Table 2 and Table 3 are the environments and requirements of the two computers.

Table 2. Environments and requirements of Windows workstation.

Windows/Ubuntu version	Windows 10
CPU	Intel(R) Core(TM) i9-7820X CPU@3.60GHz
RAM	2×32GB; 2.67MT/s
GPU	Nvidia 2080Ti
CUDA version	10
Programming language	Python3.7
Deep learning framework	Pytorch 1.6.0
Specification of dependencies	None

Table 3. Environments and requirements of Ubuntu server.

Windows/Ubuntu version	Ubuntu 20.04
CPU	Intel(R) Xeon(R) Silver 4210R CPU @ 2.40GHz
RAM	8×16GB; 2.93MT/s
GPU	Nvidia A6000
CUDA version	11
Programming language	Python3.7
Deep learning framework	Pytorch 1.8.0
Specification of dependencies	None

4.2. Training protocols

The training protocols of the baseline method is shown in Table 4.

Table 4. Training protocols.

Data augmentation methods	Sub-patch, Random rotation
Initialization of the network	Random initialization
Patch sampling strategy	Remove all background samples from the training samples
Batch size	1
Patch size	5×256×256
Total epochs	1000
Optimizer	Adam

Initial learning rate	0.0001
Stopping criteria, and optimal model selection criteria	Use the early stopping strategy to verify while training, and stop training when the increase in the verification index is less than the threshold for 4 consecutive epochs
Training time	48 hours

4.3. Testing protocols

- Pre-processing steps of the network inputs: The same strategy is applied as training steps.
- Post-processing steps of the network outputs: For the use case of small slices, for the liver, pancreas, and spleen, only the largest connection component is retained.
- If using patch-based strategy, describing the patch aggregation method:
  When aggregating patches, up sampling is done in the length and width directions, and then the patches are directly stacked along the depth direction

5. Results

5.1. Quantitative results on validation set.

The performance of our method on the validation set is shown in Table 5. For DSC, our method performs well on the liver and spleen, but it performs relatively poorly on the kidney and pancreas. For NSD, the segmentation effect on various organs needs to be improved. We will analyze the segmentation effect on the verification set in the next section.

Table 5. Quantitative results on validation set.

Organ	DSC (%)	NSD (%)
Liver	87.20±14.01	58.00±18.60
Kidney	62.94±24.87	45.16±21.40
Spleen	77.82±26.90	65.25±26.27
Pancreas	48.77±21.76	30.61±16.44

5.2. Qualitative results

Figure 4 and Figure 5 show our partial segmentation results of train set and validation set, respectively. The first column is the two-bit slice of the original image, the second column is label, and the third column is our segmentation result. In the segmentation results, red represents the liver, blue represents the spleen, green represents the kidney, and yellow represents the pancreas. Figures 4 and 5 list some relatively successful segmentation cases.

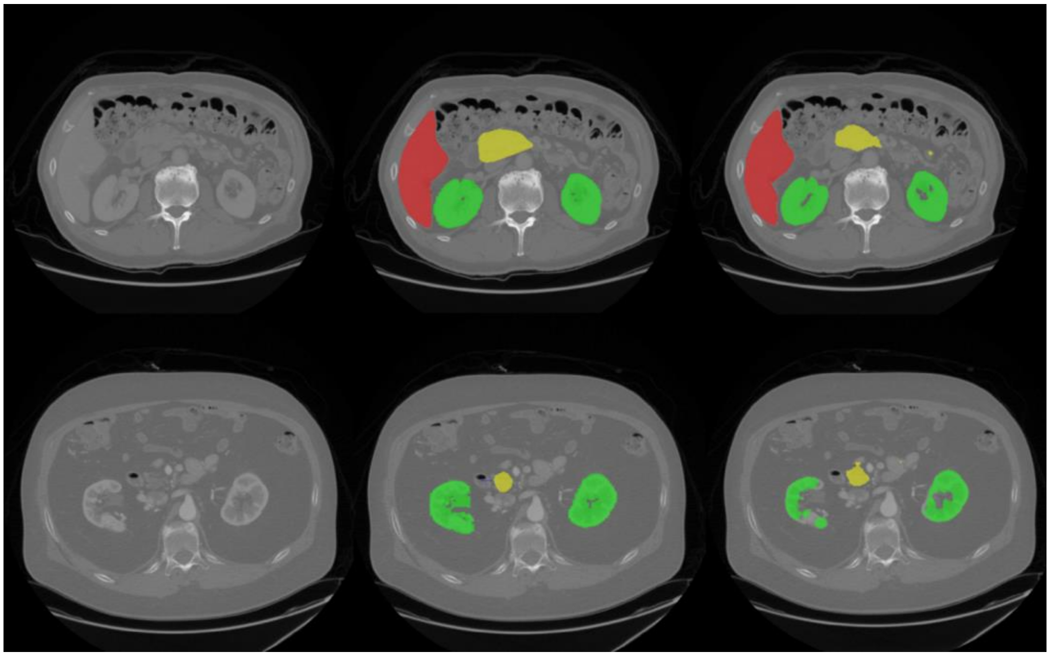

Figure 4. Segmentation results of some training sets. First column is the image, second column is the ground truth, and third column is the predicted results by our method. The visualization of this figure uses ITK-SNAP software [21].

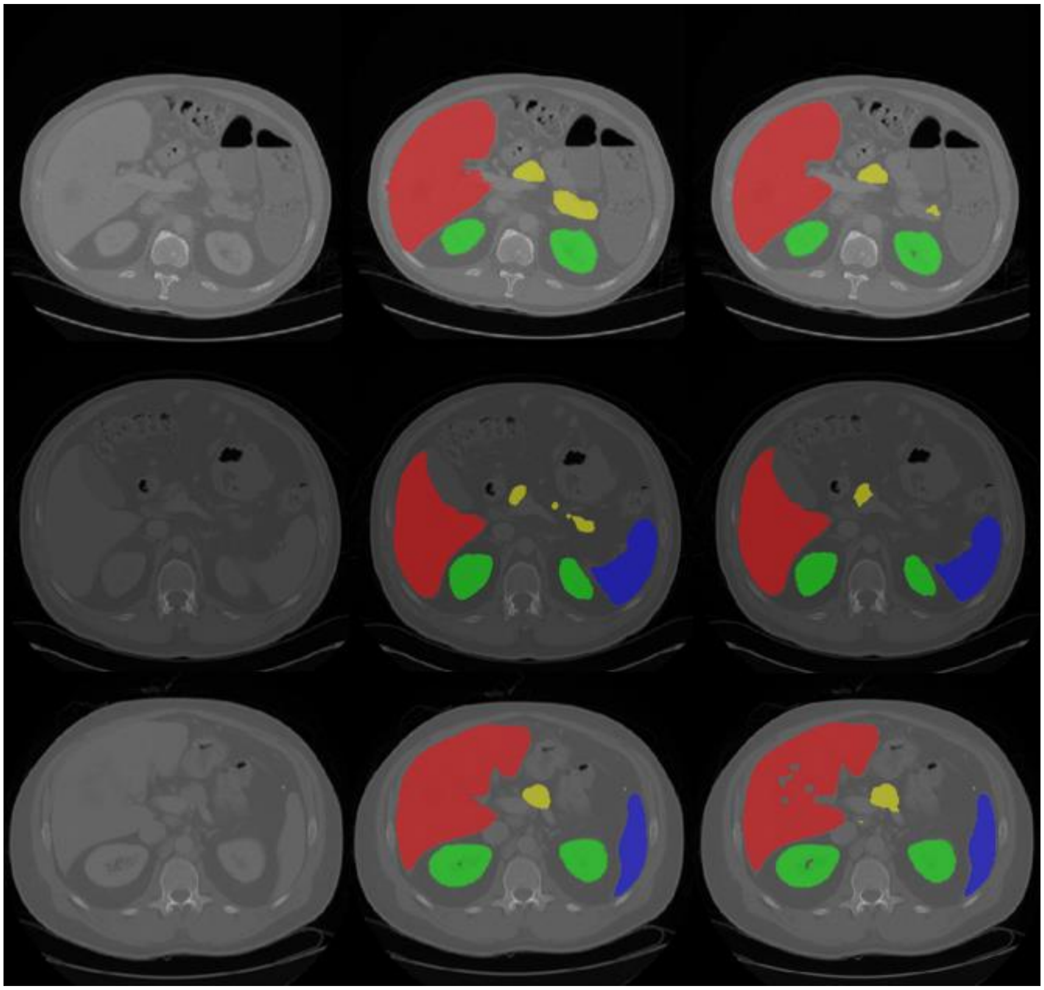

Figure 5. Segmentation results of some validation sets. First column is the image, second column is the ground truth, and third column is the predicted results by our method. The visualization of this figure uses ITK-SNAP software [21].

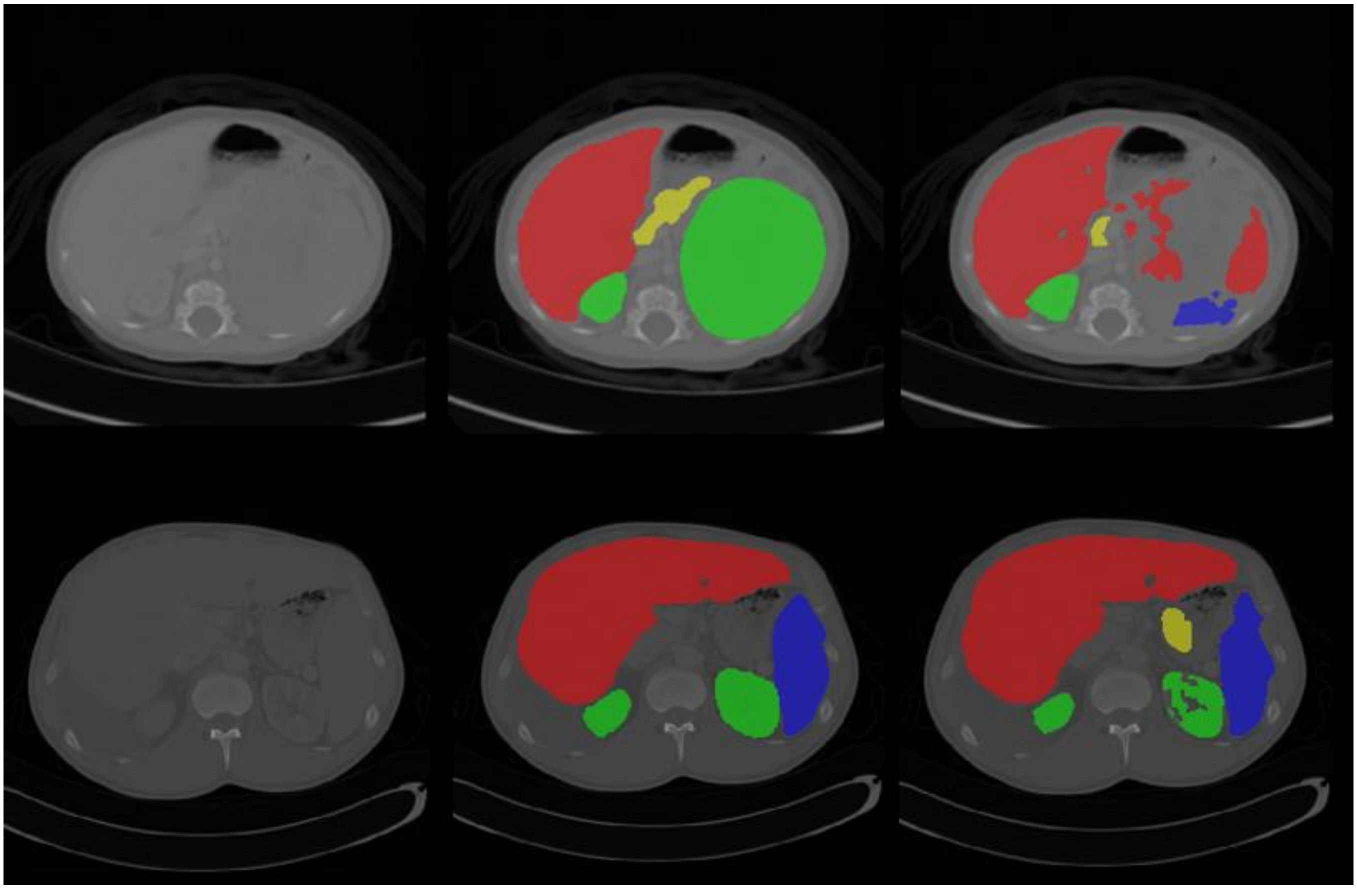

Figure 6. Some failure cases. First column is the image, second column is the ground truth, and third column is the predicted results by our method. The visualization of this figure uses ITK-SNAP software [21].

6. Discussion and Conclusion

Since this is a multi-site, multi-organ data set, there is a large heterogeneity, so the segmentation results in different cases show very big differences. We show some excellent segmentation effects in Figure 5. It can be seen that the use cases in Figure 5 usually have relatively clear outlines, and the boundaries of various organs have obvious intensity changes, so their task difficulty is relatively small, and the neural network is easier to segment good results.

At the same time, we also got a lot of low-quality segmentation results, as shown in Figure 6. First of all, these two examples have one thing in common, their intensity values are not obvious, even the naked eye is difficult to clearly distinguish the contours of each organ. Secondly, for the first use case, the size of the kidney on the right is very large, it may be that some kind of disease has occurred, so the neural network mistakenly segmented this area into other organs. In addition, the size of each organ of the multi-organ segmentation task is quite different. For example, the liver is often much larger than the pancreas. For small organs, the misclassification of a few pixel values will also lead to a sharp drop in DSC indicators.

Our method has achieved good segmentation results in some use cases, but there are also some failed cases, which shows that the generalization ability of our method needs to be improved, but our method is very lightweight and in the stable reasoning stage, it only occupies less than 3G of video memory. Compared with the method of high resource consumption, our method has a better clinical application prospect.

Acknowledgment

The authors of this paper declare that the segmentation method they implemented for participation in the FLARE challenge has not used any pre-trained models nor additional datasets other than those provided by the organizers.

References

[1] 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*, 2015, pp. 234–241. 1

[2] A. L. Simpson, M. Antonelli, S. Bakas, M. Bilello, K. Farahani, B. Van Ginneken, A. Kopp-Schneider, B. A. Landman,G. Litjens, B. Menze *et al.*, "A large annotated medical image dataset for the development and evaluation of segmentation algorithms," *arXiv preprint arXiv:1902.09063*, 2019. 2

[3] P. Bilic, P. F. Christ, E. Vorontsov, G. Chlebus, H. Chen,Q. Dou, C.-W. Fu, X. Han, P.-A. Heng, J. Hesser *et al.*, "The liver tumor segmentation benchmark (lits)," *arXiv preprint arXiv:1901.04056*, 2019. 2

[4] H. Roth, A. Farag, E. Turkbey, L. Lu, J. Liu, and R. Summers, "Data from pancreas-ct. the cancer imaging archive (2016)." 2

[5] H. R. Roth, L. Lu, A. Farag, H.-C. Shin, J. Liu, E. B. Turkbey, and R. M. Summers, "Deeporgan: Multi-level deep convolutional networks for automated pancreas segmentation," in *International conference on medical image computing and computer-assisted intervention*. Springer, 2015, pp. 556–564. 2

[6] K. Clark, B. Vendt, K. Smith, J. Freymann, J. Kirby, P. Koppel, S. Moore, S. Phillips, D. Maffitt, M. Pringle *et al.*, "The cancer imaging archive (tcia): maintaining and operating a public information repository," *Journal of digital imaging*, vol. 26, no. 6, pp. 1045–1057, 2013. 2

[7] N. Heller, F. Isensee, K. H. Maier-Hein, X. Hou, C. Xie,F. Li, Y. Nan, G. Mu, Z. Lin, M. Han *et al.*, "The state of the art in kidney and kidney tumor segmentation in contrastenhanced ct imaging: Results of the kits19 challenge," *Medical Image Analysis*, vol. 67, p. 101821, 2021. 2

[8] N. Heller, S. McSweeney, M. T. Peterson, S. Peterson,J. Rickman, B. Stai, R. Tejpaul, M. Oestreich, P. Blake,J. Rosenberg et al., "An international challenge to use artificial intelligence to define the state-of-the-art in kidney and kidney tumor segmentation in ct imaging." American Society of Clinical Oncology, vol. 38, no. 6, pp. 626–626, 2020. 2

[9] J. Ma, Y. Zhang, S. Gu, C. Zhu, C. Ge, Y. Zhang, X. An,C. Wang, Q. Wang, X. Liu, S. Cao, Q. Zhang, S. Liu,Y. Wang, Y. Li, J. He, and X. Yang, "Abdomenct-1k: Is abdominal organ segmentation a solved problem?" IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021. 2, 3, 4

[10] Xingjian, S. H. I., et al. "Convolutional LSTM network: A machine learning approach for precipitation nowcasting." Advances in neural information processing systems. 2015.

[11] Song, Hongmei, et al. "Pyramid dilated deeper convlstm for video salient object detection." Proceedings of the European conference on computer vision (ECCV). 2018.

[12] Çiçek, Özgün, et al. "3D U-Net: learning dense volumetric segmentation from sparse annotation." International conference on medical image computing and computer-assisted intervention. Springer, Cham, 2016.

[13] Qin, Wenjian, et al. "Superpixel-based and boundary-sensitive convolutional neural network for automated liver segmentation." Physics in Medicine & Biology 63.9 (2018): 095017.

[14] Tang, Wei, et al. "A two-stage approach for automatic liver segmentation with Faster R-CNN and DeepLab." Neural Computing and Applications (2020): 1-10.

[15] Le, Doan Cong, et al. "Semi-automatic liver segmentation based on probabilistic models and anatomical constraints." Scientific Reports 11.1 (2021): 1-19.

[16] da Cruz, Luana Batista, et al. "Kidney segmentation from computed tomography images using deep neural network." Computers in Biology and Medicine 123 (2020): 103906.

[17] Ala'a, R., et al. "Kidney segmentation in MR images using active contour model driven by fractional-based energy minimization." Signal, Image and Video Processing 14.7 (2020): 1361-1368.

[18] Moon, Hyeonsoo, et al. "Acceleration of spleen segmentation with end-to-end deep learning method and automated pipeline." Computers in biology and medicine 107 (2019): 109-117.

[19] Cai, Jinzheng, et al. "Improving deep pancreas segmentation in CT and MRI images via recurrent neural contextual learning and direct loss function." arXiv preprint arXiv:1707.04912 (2017).

[20] Zhou, Yuyin, et al. "A fixed-point model for pancreas segmentation in abdominal CT scans." International conference on medical image computing and computer-assisted intervention. Springer, Cham, 2017.

[21] Yushkevich, Paul A., et al. "User-guided 3D active contour segmentation of anatomical structures: significantly improved efficiency and reliability." Neuroimage 31.3 (2006): 1116-1128.