LIGHTWEIGHT V-NET FOR LIVER SEGMENTATION

Tao Lei[1], Wenzheng Zhou[1], Yuxiao Zhang[1], Risheng Wang[1], Hongying Meng[2], Asoke K. Nandi[2]

[1]School of Electronic Informataion and Artifical Intelligence, Shaanxi University of Science and Technology, Xi'an 710021, P. R. China

[2]Department of Electronic and Computer Engineering, Brunel University London, Uxbridge, Middlesex, UB8 3PH, United Kingdom

ABSTRACT

The V-Net based 3D fully convolutional neural networks have been widely used in liver volumetric data segmentation. However, due to the large number of parameters of these networks, 3D FCNs suffer from high computational cost and GPU memory usage. To address these issues, we design a lightweight V-Net (LV-Net) for liver segmentation in this paper. The proposed network makes two contributions. The first is that we design an inverted residual bottleneck block (IRB block) and a 3D average pooling block and apply them to the proposed LV-Net. Compared with vanilla convolution, depth-wise convolution and point-wise convolution employed by the IRB block can not only reduce the number of parameters significantly, but also extract features sufficiently well by decoupling cross-channel corrections and spatial correlations. The second is that the LV-Net employs 3D deep supervision to improve the final loss function in training phase, which makes the proposed LV-Net acquire a more powerful discrimination capability between liver areas and non-liver areas. The proposed LV-Net is evaluated on public LiTS dataset, and experiments demonstrate that the proposed LV-Net is superior to popular 2D and 3D networks in terms of segmentation performance, parameter quantity and computational cost.

Index Terms— deep learning, image segmentation, 3D fully convolutional neural network, network compression

1. INTRODUCTION

As automatic image segmentation algorithms can help doctors to improve liver disease diagnosis and develop a better treatment plan, liver segmentation remains one of the hotspots in the field of medical image analysis. Before the appearance of deep learning [1], three kinds of popular image segmentation algorithm are often used for liver segmentation: grayscale value-based algorithms [2][3][4], statistical shape model-based algorithms [5][6][7], and texture feature-based algorithms [8][9]. However, liver has a complex and variant shape, and it has a similar grayscale value with neighboring organs in CT images; thus it is very difficult to extract features of liver using those algorithms. With the development of deep learning techniques [10], end-to-end liver segmentation attracts the attention of researchers. As deep learning can learn high-layer semantic features of liver from abdominal images, CNNs based on deep learning can provide excellent segmentation results. Currently, there are two types of popular deep convolutional neural network used for liver segmentation, the first is 2D networks such as U-Net [11] and CE-Net [12], the second is 3D networks such as 3D U-Net [13] and V-Net [14].

Although deep learning can achieve better liver segmentation in the way of end-to-end, it causes some new problems that limit the clinical deployment of deep learning. As liver slices constitute a volumetric data, it is difficult to utilize 3D spatial information of liver slices when a 2D CNN is used for liver segmentation. Compared with 2D CNNs, the 3D CNNs can utilize the spatial information among neighboring liver slices effectively; therefore, they achieve better segmentation results. Unfortunately, these 3D CNNs require a large number of network parameters and high computational cost. Researchers usually must use image patch or image zooming to train these 3D CNNs, which is a tradeoff between segmentation performance and hardware resource requirements. Therefore, how to remove redundant parameters and reduce the computational cost effectively of 3D CNNs are important when we extend 3D CNNs to practical clinical application.

To address these issues mentioned above, we propose a lightweight V-Net (LV-Net) for liver segmentation. The proposed network is more practical since it requires less memory usage while maintaining liver segmentation accuracy. Experiments demonstrate that the proposed LV-Net is superior to popular CNNs since it provides better segmentation results with less memory usage.

2. THE PROPOSED NETWORK

V-Net is a very popular 3D fully convolutional neural network in medical image segmentation. It has a symmetric structure, and it is composed of an encoder and a decoder. The encoder is used to extract useful features from input data and the decoder is used to reconstruct the features to obtain

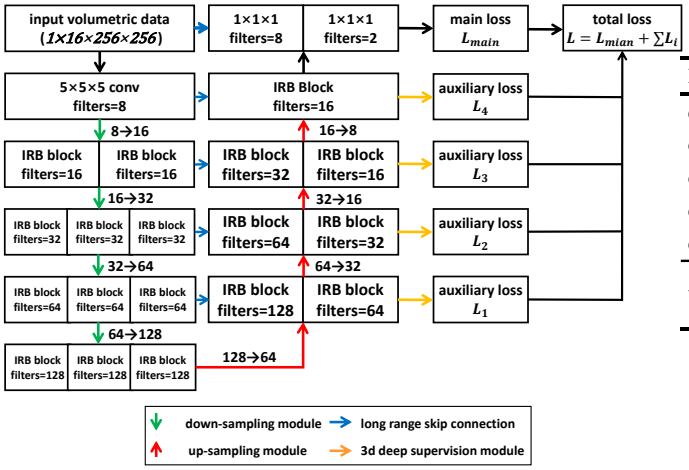

Fig. 1: The detailed structure of LV-Net.

Table 1. Comparison of the efficiencies of different networks.

Models	IRB block	Vanilla 3D Convolution
expansion point-wise convolution	$(1 \times 1 \times 1 \times C) \times C \times \epsilon$	
depth-wise convolution	$(k \times k \times k \times 1) \times C \times \epsilon$	$(k \times k \times k) \times C \times C$
contraction point-wise convolution	$(1 \times 1 \times 1 \times C \times \epsilon) \times C$	
total	$k^3C\epsilon + 2C^2\epsilon$ 64,768 (12.65 %)	k^3C^2 512,000 (100 %)

the final segmentation result. V-Net employs long range skip connections between encoder and decoder at symmetric layers to fuse low-layer features and high-layer features together, which improves the final predictions. Both encoder and decoder depend on vanilla 3D convolution. The design of V-net causes large memory usage and high computational cost.

To address those drawbacks of V-Net, we propose a lightweight V-Net for liver segmentation. The proposed LV-Net shown in Fig. 1 has two advantages: (1) the LV-Net only requires low memory usage since it removes a large number of redundant parameters existing in the V-Net; and (2) the LV-Net provides better segmentation results than popular 2D and 3D networks due to the employment of deep supervision.

2.1. Network compression

According to Fig. 1, we use inverted residual bottleneck (IRB) block [15] instead of vanilla convolution to construct encoders and decoders of LV-Net. The IRB block is composed of depth-wise convolution and point-wise convolution. Fig. 2a shows the detailed architectures of the IRB block.

In the V-Net, $5 \times 5 \times 5$ convolutional kernels are used to extract spatial-dimension features and channel-dimension features. But in a IRB block, the input feature maps are firstly expanded on channels via the operation of $1 \times 1 \times 1$ point-wise convolution; secondly, the operation of $5 \times 5 \times 5$ depth-wise convolution is used to extract spatial-dimension features; thirdly, the $1 \times 1 \times 1$ point-wise convolution is used to squeeze feature channels; finally, the composition of residual feature maps and the input feature maps is considered as the final output feature maps. Compared with the bottleneck block of ResNet [16], both the entrance and exit of the IRB block are narrow, but the middle part of the IRB block is wide. There are two advantages of the design of IRB block: (1) the IRB block can extract more features from input feature maps at the input stage; and (2) the IRB block can remove redundant features by squeezing the channel dimension of output feature maps at output stage. Besides, the width of the middle part of IRB block is decided by expansion rate ϵ. The parameter ϵ is an important hyper-parameter that can adjust the model capacity of the network. Tuning the ϵ reasonably can avoid over-fitting phenomenon and make LV-Net fit other segmentation tasks flexibly. Due to the limitation of GPU memory, in this paper, we fix ϵ to 4 to get the best performance.

As the vanilla convolution achieves the united mapping of feature maps on spatial correlations and cross-channel correlation, spatial features and channel features are often coupled together [17] at the output feature maps of vanilla convolution, which limits the feature extraction of subsequent convolutional layers. The IRB block is a variant of depth-wise separable convolution, and it has stronger capability of feature extraction since the IRB block can overcome the coupling between spatial features and channel features via depth-wise convolution and point-wise convolution. Besides, as the point-wise convolution and depth-wise convolution employ $1 \times 1 \times 1$ kernels, $k \times k \times k$ kernels, respectively, the IRB block reduces of the number of network parameters. For Fig. 2b, Table 1 shows the comparison of number of network parameters between one IRB block and one vanilla convolution layer, where $k \times k \times k$ is the size of a kernel, $k = 5$, C is the number of channels, $C = 64$, and $\epsilon = 4$. It is clear that the IRB block requires fewer parameters (12.65%) than a convolution layer of V-Net.

To compress the size of V-Net further, we use the composition of point-wise convolution and average pooling instead of vanilla convolution in the stage of down-sampling, and use point-wise convolution and trilinear interpolation instead of deconvolution in the stage of up-sampling. As there is no trainable parameter in pooling layers and trilinear interpolation layers, the proposed down/up-sampling module can reduce the number of network parameters. Fig. 2b shows the detailed architecture of our down-sampling module, where the parameter number of LV-Net is $2C^2$ while V-Net is $16C^2$ at the stage of down-sampling, LV-Net is $C^2/2$ while V-Net is $4C^2$ at the stage of up-sampling.

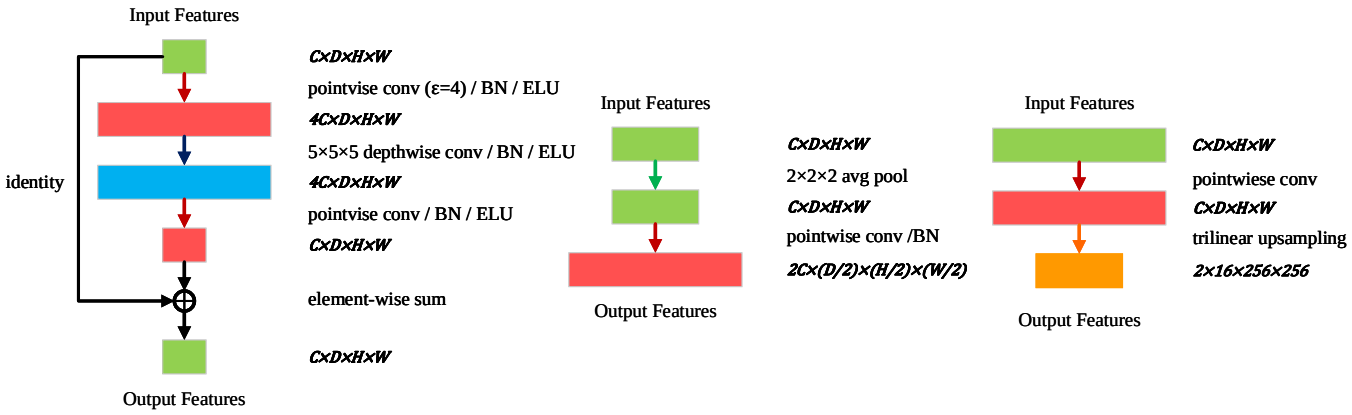

Fig. 2: The modules in LV-Net. (a) IRB block. (b) Downsampling module. (c) 3D supervision module.

2.2. 3D deep supervision

In the training phase of deep neural networks, gradient vanishing is a notorious problem due to the difficulty of transmitting gradient value to shallow layers, which makes it more difficult to train deep networks [18]. Especially, this will be worse when we use a 3D convolutional neural network including a large number of parameters to train a small dataset [19]. To address the issue, we present a novel strategy that integrates deep supervision mechanism into decoder as shown in Fig. 1. We can see that a branch network is considered as a constraint after each decoder stage, the branch is able to inject gradient values from different losses, which can avoid the problem of gradient vanishing. Fig. 2c shows the detailed architecture of 3D deep supervision, where the point-wise convolution is firstly used for the input feature maps, then the trilinear interpolation is used for up-sampling, and finally softmax layer is used for computing the probability map of the segmentation result. Here, we use a cross-entropy loss function to estimate the difference between the final feature maps and labels.

We define the loss function from the end of decoder as the main loss function denoted by L_{main}

$$L_{main}(X, W) = \sum_{x_i \in X} -\log p(t_i|x_i; W), \qquad (1)$$

where X denotes training samples, W denotes the parameters of backbone network, t_i is the label of x_i, $x_i \in X$. Besides, there are also four auxiliary loss functions denoted by L_1, L_2, L_3, L_4, respectively,

$$L_i(X; W_i, \hat{w}_i) = \sum_{x_i \in X} -\log p(t_i|x_i; W_i; \hat{w}_i), \qquad (2)$$

where W_i denotes parameters of backbone networks, $\hat{w}_i$ denotes the parameters of point-wise convolution in deep supervision block. According to Eq. 1 and Eq. 2, we present the final loss function of LV-Net

$$L = L_{main} + \sum_{i=1}^{4} \eta_i L_i + \lambda(\|W\|^2 + \sum_{i=1}^{4} \|\hat{w}_i\|^2), \qquad (3)$$

where η_i represents the weight of the i-th auxiliary loss function. Here the third term of Eq. 3 is the weight decay, where λ denotes the decay coefficient.

As different convolutional layers of decoder have different contributions on the final loss function, the balancing weight η_i is variant for L_i. Generally, the deeper a network is, the wider the perceptive field is, and the representation capability of feature is stronger, which means the output of decoder at deep layers is more important than the output of shallow layers. According to this principle, we set $\eta_1 = 0.2$, $\eta_2 = 0.4$, $\eta_3 = 0.6$, and $\eta_4 = 0.8$, respectively.

Based on the analysis above, the introduction of deep supervision has two advantages: (1) it improves the training efficiency of a network, i.e., it can speed up the convergence of network; and (2) it can help the network to learn more complex and useful features leading to high segmentation performance.

3. EXPERIMENTS

To evaluate the performance of proposed LV-Net on liver segmentation tasks, we consider MICCAI 2017 Liver Tumor Segmentation Challenge (LiTS) dataset [20] as experimental data. The LiTS includes 131 labeled 3D CT scans, where the resolution in-plane ranges from 0.55mm to 1.0mm and slice spacing ranges from 0.45mm to 6.0mm. In our experiments, we randomly choose 90 and 10 volume data to construct training set and validation set, respectively. The other 31 volume data are considered as test set. Experiments are performed on a workstation with Intel Core i9 9900X @ 3.5GHz, 128GB RAM, double NVIDIA GeForce RTX 2080Ti GPU, Windows 10 Pro, and PyTorch 1.2.

3.1. Dataset pre-processing

In this experiment, image preprocessing includes three stages: truncating the range of image intensity values, scaling the slice, and normalizing the grayscale value of slice. The first stage is used to enhance liver area and remove irrelevant details, which can achieve better feature learning and thus improve segmentation effect. Here we set the range of [-200, 200] HU. The second stage is used to reduce the memory requirement of hardware environment. Here we choose sequential 16 slices and resize each slice from 512×512 to 256×256. The last stage is used for the normalization of input samples, which is a key factor that affects the final segmentation performance. Here, we use mean value and variance to normalize input data.

3.2. Training

We set the values of hyper-parameters to train LV-Net. The batch size is set to 4. The initial learning rate is 0.001, and it multiplies 0.9 at the end of each epoch. We use cross entropy loss and adaptive moment estimation (ADAM) [21] to optimize the network, and weight decay is set to 1e-5. The total loss is the weighted sum of main loss and auxiliary losses. In addition, ELU [22] is considered as the activate function, which can not only boost up the traning speed, but also bring better generalization performance than ReLU. Batch normalization is performed after each convolutional layer. The validation set mentioned above is used to check whether the model is overfitting or not at the end of each epoch. Once the model achieves the best performance on validation set which often happens after about 20 epochs' training, the training stops and the model parameters are saved for further evaluation.

3.3. Evaluation and results

We use five metrics to evaluate comprehensively the segmentation quality of each network in this experiment. They are: DICE per case (DICE), volume overleap error (VOE), relative volume difference (RVD), average symmetric surface distance (ASSD [mm]), and maximum symmetric surface distance (MSSD [mm]). Note that a perfect segmentation means that the value of DICE score is 1, while the value of each of VOE, ASSD, and MSSD score is 0.

Table 2 presents the segmentation performance on the test set using U-Net [11], CE-Net [12], 3D U-Net [13], V-Net [14], and proposed LV-Net. It is clear that our LV-Net achieves an average DICE of 0.954, an average VOE of 0.086, an average RVD of 0.016, an average ASSD of 1.871 mm, an average MSSD of 29.496 mm. Except for the value of RVD that is slightly lower than the value provided by 3D U-Net in the first place, the remaining metrics of LV-Net are higher than comparative networks. The LV-Net shows the best segmentation performance on the LiTS dataset.

Table 2. Quantitative evaluation results of different networks on the liver segmentation testing set.

Models	DICE	VOE	RVD	ASSD (mm)	MSSD (mm)
U-Net [11]	0.9399	0.1114	0.0322	5.7985	123.5763
CE-Net [12]	0.9404	0.1103	0.0619	4.1162	115.4076
3D U-Net [13]	0.9400	0.1113	**0.0142**	2.6173	36.4352
V-Net [14]	0.9426	0.1065	0.0192	2.4887	38.2826
LV-Net	**0.9543**	**0.0856**	0.0156	**1.8705**	**29.4960**

Table 3. Comparison of the efficiencies of different networks.

Models	trainable parameters	operations (GFLOPs)	storage usage (MB)
U-Net [11]	13,394,242	123.96	51.15
CE-Net [12]	29,003,668	35.78	110.77
3D U-Net [13]	16,320,322	1,032.80	62.27
V-Net [14]	65,173,903	516.12	248.69
LV-Net	**1,659,282**	**58.07**	**6.56**

We also count the quantities of trainable parameters and computational costs of networks above, as shown in Table 3. Compared with 2D CNNs, 3D CNNs obtains a certain increase in segmentation performance, but they require more memory usage and high computational cost. The proposed LV-Net overcomes the drawbacks of 3D CNNs due to the utilization of depth separable convolution and the design of down/up-sampling modules. Consequently, the LV-Net is significantly ahead of other 2D CNNs and 3D CNNs on the number of trainable parameters, computational cost, and storage usage. For example, the number of trainable parameters of LV-Net is only 2.55 % of the vanilla V-Net, the computational cost is 11.25 %, and the storage usage is 2.64 %.

4. CONCLUSION

In this work, we mainly studied liver segmentation based on 3D deep convolutional neural networks. We presented a lightweight V-Net based 3D network by employing depth separable convolution and 3D deep supervision to reduce the memory requirement of 3D network while maintaining the segmentation accuracy for a liver. Experiments demonstrated that the proposed LV-Net can achieve higher segmentation accuracy and is much lighter than popular 2D and 3D networks such as U-Net, CE-Net, 3D U-Net, and V-Net. Consequently, the proposed LV-Net is more suitable for clinical practice, and it can be extended to other medical volumetric segmentation tasks easily.

In the future, we will implement the LV-Net to liver tumor segmentation and liver blood vessel segmentation tasks that are more challenging and valuable in clinical medicine. Additionally, the neural architecture search (NAS) will be studied to search better basic blocks for improving the LV-Net.

5. REFERENCES

[1] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504-507, 2002.

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

[3] T. Lei, X. Jia, Y. Zhang, L. He, H. Meng, and A. K. Nandi, "Significantly fast and robust fuzzy c-means clustering algorithm based on morphological reconstruction and membership filtering," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 5, pp. 3027–3041, 2018.

[4] R. Adams and L. Bischof, "Seeded region growing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 6, pp. 641–647, 1994.

[5] T. Heimann, I. Wolf, and H.-P. Meinzer, "Active shape models for a fully automated 3D segmentation of the liver–an evaluation on clinical data," in *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2006, pp. 41–48.

[6] X. Zhang, J. Tian, K. Deng, Y. Wu, and X. Li, "Automatic liver segmentation using a statistical shape model with optimal surface detection," *IEEE Trans. Biomed. Eng.*, vol. 57, no. 10, pp. 2622–2626, 2010.

[7] S. Tomoshige, E. Oost, A. Shimizu, H. Watanabe, and S. Nawano, "A conditional statistical shape model with integrated error estimation of the conditions; application to liver segmentation in non-contrast CT images," *Med. Image Anal.*, vol. 18, no. 1, pp. 130–143, 2014.

[8] O. Gambino et al., "Automatic volumetric liver segmentation using texture based region growing," in *2010 International Conference on Complex, Intelligent and Software Intensive Systems*, 2010, pp. 146–152.

[9] H. Ji, J. He, X. Yang, R. Deklerck, and J. Cornelis, "ACM-based automatic liver segmentation from 3-D CT images by combining multiple atlases and improved mean-shift techniques," *IEEE J. Biomed. Heal. Informatics*, vol. 17, no. 3, pp. 690–698, 2013.

[10] 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 (CVPR)*, 2015, pp. 3431–3440.

[11] 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 (MICCAI)*, 2015, pp. 234–241.

[12] Z. Gu, J. Cheng, H. Fu, K. Zhou, H. Hao, Y. Zhao, et al, "CE-Net: Context Encoder Network for 2D Medical Image Segmentation," *IEEE Trans. Med. Imaging*, pp. 2281-2292, 2019.

[13] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3D U-Net: learning dense volumetric segmentation from sparse annotation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2016, pp. 424–432.

[14] 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*, 2016, pp. 565-571.

[15] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4510–4520.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[17] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1251–1258.

[18] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, "Deeply-supervised nets," in *Artificial Intelligence and Statistics*, 2015, pp. 562–570.

[19] Q. Dou, L. Yu, H. Chen, Y. Jin, X .Yang, J. Qin, et al., "3D deeply supervised network for automated segmentation of volumetric medical images," *Med. Image Anal.*, vol. 41, pp. 40–54, 2017.

[20] P. Bilic, P. F. Christ, E. Vorontsov, G. Chlebus, H. Chen, Q. Dou, et al., "The Liver Tumor Segmentation Benchmark (LiTS)," arXiv:1901.04056, 2019.

[21] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015.

[22] D.-A. Clevert, T. Unterthiner and S. Hochreiter, "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)," in *International Conference on Learning Representations (ICLR)*, 2016.