Hybrid Cascaded Neural Network for Liver Lesion Segmentation

Raunak Dey, Yi Hong

Computer Science Department, University of Georgia

Abstract. Automatic liver lesion segmentation is a challenging task while having a significant impact on assisting medical professionals in the designing of effective treatment and planning proper care. In this paper we propose a cascaded system that combines both 2D and 3D convolutional neural networks to effectively segment hepatic lesions. Our 2D network operates on a slice by slice basis to segment the liver and larger tumors, while we use a 3D network to detect small lesions that are often missed in a 2D segmentation design. We employ this algorithm on the LiTS challenge obtaining a Dice score per case of 68.1%, which performs the best among all non pre-trained models and the second best among published methods. We also perform two-fold cross-validation to reveal the over- and under-segmentation issues in the LiTS annotations.

Keywords: Liver lesion segmentation · Hybrid neural network · Small lesion segmentation.

1 Introduction

Liver lesions are groups of abnormal cells in the liver and some of them lead to the cancer. Liver cancer is one of the leading causes of cancer deaths worldwide and more than 700,000 deaths are reported each year according to the American Cancer Society. For the liver cancer screening, the Computer Tomography (CT) is the most commonly used imaging tool and the technique of the automatic liver lesion segmentation from a CT scan has great impacts on cancer diagnosis, surgery planning, and treatment evaluations.

Due to the heterogeneous and diffusive appearance of hepatic lesions, the liver lesion segmentation is a challenging task. Researchers have proposed many segmentation algorithms based on the classical segmentation techniques, e.g., thresholding [15], region growing [16], active contour [14], and ensemble [8]. Recently, deep neural networks have been widely used in the liver lesion segmentation and have shown improved performance in segmenting and detecting liver lesions. These algorithms either use 2D convolutional neural networks [4][11], 3D networks [9], or a combination of both [5][3]. One of the issues facing by the existing methods is the segmentation of the small lesions, which are often missed in their segmentation predictions.

In this paper we introduce a hybrid cascaded segmentation network to segment the liver lesions, especially the small ones. The network utilizes both 2D
Fig. 1. Overview of our proposed hybrid cascaded network for the liver lesion segmentation. We adopt 2D CompNets (see Fig. 2 for architecture details) to segment the liver region and large lesions and apply a 3D CompNet (Fig. 4) to segment the small lesions. The red dashed lines indicate the test phase of the segmentation.

and 3D convolutional networks to effectively find both large and small lesions in the liver region. We use a 2D convolutional neural network (CNN) to obtain the liver mask from the input CT volume of the abdomen region, which locates the liver region for further processing. Another 2D CNN is then employed to extract large lesions from 2D slices, while a dedicated 3D CNN is proposed to segment out small lesions from 3D volumes.

We evaluate our method on the Liver Tumor Segmentation (LiTS) dataset [1] and obtain a Dice score per subject of 68.1% without any pre-training or post-processing. This result ranks first among the currently published non pre-trained networks. Moreover, with the proposed special treatment of the small lesion segmentation using a 3D CNN, we improve the network performance of segmenting the large lesions only by 7.1% in terms of the Dice score per subject. In addition, using a two-fold cross-validation on the LiTS training set, we observe that our network can segment unannotated regions which share the similar intensity as that of the tumor region annotated on the same slice in the LiTS dataset.

2 Method

2.1 Overview

In this section we describe our hybrid cascaded network to handle the liver lesion segmentation problem. We first segment the liver from the original CT scan slice-by-slice using a 2D neural network. The generated liver mask allows us to focus on the lesion segmentation located inside of the liver. At the stage of lesion segmentation, we treat large and small lesions separately with a 2D network branch to segment large lesions and a 3D network branch to segment small lesions. This hybrid design was motivated by the goal of balancing the computational efficiency without sacrificing the accuracy. Compared to the 3D
segmentation network, a 2D network is more efficient in terms of both computation time and memory cost; however, on a 2D slice small lesions are likely to be confused with some normal tissues in the liver region. Therefore, we propose to use an adaptive 3D network to handle the small lesion segmentation, which greatly reduces such false positives. Figure 1 depicts the overview of the proposed hybrid cascaded network. Its backbone network is the CompNet [6], which was demonstrated to be more robust in the segmentation on pathological images than the U-Net [17].

### 2.2 Liver Segmentation

A liver CT scan typically has hundreds of slices and each slice has a resolution of $512 \times 512$ pixels. Directly working on such 3D images would cause time and memory issues using a segmentation neural network under an encoder-and-decoder design. Fortunately, the segmentation of the liver region inside the abdominal cavity is relatively easier than the segmentation of the liver lesions. As demonstrated by recent works [4][12][18] and our experiments, liver segmentation relied only on 2D slices can achieve more than 95% overall dice accuracy. Therefore, we use a 2D network to segment out the liver from a 3D CT scan.

We follow the design of the basic CompNet proposed in [6] and adapt it to our liver segmentation task, resulting in the 2D network architecture shown in Fig. 2. This network takes 2D slices as input and outputs 2D liver masks that are stacked into a 3D mask. The encoder branches of the liver segmentation, its complementary segmentation, and the reconstruction sections consist of multiple blocks of two convolutional layers followed by the batch normalization layers. All convolutional layers use a filter size of $3 \times 3$ and the filter numbers in the blocks are as follows 32, 64, 128, 256, and the transition block of 512. Following each block we have a pooling layer. The decoder branches of the network mirror...
the encoder ones with the pooling layers being replaced by 2D transposed convolutional layers. We multiply the input 3D scan with the obtained liver mask to retain only the liver region for the following lesion segmentation.

### 2.3 Two-Step Lesion Segmentation

After having the liver segmented, we might directly apply a 3D segmentation network to extract the liver lesions. However, a 3D Network operating only in the liver region is still computationally expensive and may suffer from the problem of lacking 3D training scans. In order to mitigate these issues, we propose to use a mixed network to handle the segmentation of large and small lesions separately. In particular, we use a 2D network to scan slice-by-slice and predict larger tumors if present. However, this method is not sufficient for predicting small lesions, as the false positive examples shown in Fig. 3. This happens due to the appearance of the small lesion matching that of other tissues or vessels on 2D liver slices. A 3D network learning from cropped volumes with small lesions can reduce such false positives, because it leverages the observation that the location of the tissue or vessel appears to travel considerably in-between slices whereas the liver lesion constantly appears to be roughly stationary across slices. Regarding the threshold of the lesion size used to separate the large and small lesions, it may vary for different applications and datasets. Here, we set the threshold as a resolution of $32 \times 32 \times 32$ by experimentally testing on the LiTS dataset, which greatly balances the prediction accuracy and the computational efficiency.

**Large Lesion Segmentation.** We use another 2D CompNet like the one used for the liver segmentation (Fig. 2) to extract lesions larger than $32 \times 32 \times 32$ from the input slices with the liver only. Since this network takes care of the prediction of large lesions, we clean the training masks by using the component labeling to remove all small lesion masks, whose horizontal and vertical dimensions are both less than or equal to 32 on a 2D slice, for every slice. In the test phase we also remove the detected lesions less than $32 \times 32$ using the component labeling.

**Small Lesion Segmentation.** We use a 3D CompNet as shown in Fig. 4 to segment the small lesions. This 3D network takes a volume of $32 \times 32 \times 32$ as input. The encoder branches of the lesion segmentation, its complementary segmentation, and the reconstruction consist of three 3D convolutional blocks. Each block is composed of two 64, 128, and 256 feature maps, respectively, each followed by a 3D pooling layer with a pool size $(2, 2, 2)$. The transition block has 512 feature maps. The decoder branches mirror the encoder branches with the pooling layers replaced by the 3D transposed convolutional layers.
The U-Net-like network has the small lesion segmentation (top left), its complementary segmentation (bottom left), and the image reconstruction (right). Each block has two convolutional layers with the numbers of feature maps showing in the block box.

To generate the training samples for this 3D network, we use the component labeling to locate the center of a lesion and estimate its dimensions on a 2D slice. We sample a 3D volume for a lesion whose both horizontal and vertical dimensions on the 2D slice are smaller than or equal to 32. Since the small lesions often occupy only a few slices, we choose a cube with a size of $32 \times 32 \times 32$ to cover the lesion. In particular, around the lesion center located on a slice, we select 15 slices above and 16 slices below to create a 3D volumes for training.

During the testing phase, we use a 3D sliding cube over the liver volume to predict the small lesions using the above trained 3D network. To choose an appropriate stride size for the sliding cube, we test on 32, 16, 8, and 4 voxels, respectively. It turns out that the stride 32 and 16 are too large to capture tiny lesions. The stride 8 is an appropriate one, since a further reduction of the stride size does not reveal any further changes but increasing the prediction time. To fuse the overlap predictions, we take the average of all predictions at that voxel and set the value greater than 0.5 to 1 as the final prediction.

3 Experiments

Dataset and Preprocessing. To test our proposed network, we use the public Liver Tumor Segmentation (LiTS) dataset, which consists of 130 abdomen CT scans for training and 70 for testing. To train the 2D liver segmentation network, we fully use all training scans with a total of 58,638 2D slices. In the network training of the large liver lesion segmentation, we focus on the slices with the liver present, resulting in 19,163 2D slices in total. The 3D network is trained on 11,503 3D small lesion samples with size of $32 \times 32 \times 32$.

We preprocess the liver CT scans using a histogram based thresholding method. We select the rightmost peak of the intensity histogram distribution
of a CT scan for normalization and use the histogram equalization algorithm to generate the enhanced images as shown in Fig. 5. The implementation of the preprocessing and the liver lesion segmentation network is available online.

**Experimental Settings.** We use Keras with the TensorFlow backend to implement our proposed network. The 2D CompNet for the liver segmentation is trained for 40 epochs using the Adam optimizer with a learning rate of 5e-5. Next, we train the 2D and 3D CompNets for the large and small lesion segmentation in the same manner, i.e., we first train the networks using the Adam optimizer with a learning rate of 5e-5, the same as that used in [4], and having an early stopping scheme with the tolerance being set to 5; then we train the networks with a learning rate of 1e-6 using an early stopping with a tolerance of 10 trials. Both steps have 150 maximum number of epochs for training. In addition, we use an L2 regularization with a parameter of 2e-4 and a dropout with a rate of 0.3 after all pooling and upsampling layers to mitigate overfitting.

**Experimental Results.** Since we do not have the ground truth for the LiTS test set, we first perform the two-fold cross-validation on the training set to quantitatively and qualitatively evaluate the performance of our proposed method. We obtain 67.3% Dice per case and some visual samples are presented in Fig. 6.

By evaluating on the test set of the LiTS challenge, we list in Table 1 our result with comparison to those of the currently published approaches. According to the Dice score per case, the most important metric for measuring an algorithm’s performance on the LiTS challenge, our approach is at the second rank.

---

**Fig. 5.** An example of a liver CT scan before (left) and after (right) preprocessing.

**Fig. 6.** Comparison between our predictions (green) and the LiTS annotations (blue) using the two-fold cross-validation on the training set. The red arrows indicate the artifacts which share similar intensity distribution to tumors annotated on the same slice but are missing in the annotations while being predicted by our method.
| Method            | Dice Per Case | Dice Overall | No Pre-training | No Post-processing |
|-------------------|---------------|--------------|-----------------|-------------------|
| H-Dense U-Net [12]| 0.722         | 0.824        | ✓               | ✓                 |
| Multiple U-Nets [1]| 0.680         | 0.796        | ✓               | ✓                 |
| 2.5 D U-Net [7]    | 0.670         | —            | ✓               | ✓                 |
| CDNN [18]          | 0.657         | 0.820        | ✓               | ✓                 |
| FED-Net [3]        | 0.650         | 0.766        | ✓               | ✓                 |
| AH-Net [13]        | 0.634         | 0.834        | ✓               | ✓                 |
| RA U-Net [10]      | 0.595         | 0.795        | ✓               | ✓                 |
| BS-Unet [11]       | 0.552         | 0.729        | ✓               | ✓                 |
| Ours (w/o small lesion) | 0.681 (0.610) | 0.813 (0.776) | ✓               | ✓                 |

Table 1. Comparison among published approaches and ours on the LiTS challenge.

Fig. 7. Examples of our predictions (indicated by the green lines) on the LiTS test set.

following after the H-Dense U-Net [12], which however needs pre-training. In addition, our small lesion segmentation could be an add-on component to their method for a further improvement in segmenting small tumors as mentioned in their future work. Our experiment shows that the dice per case can be improved from 0.61 to 0.681 with the consideration of the small lesion segmentation. Worth to mention that, among the methods without pre-training and post-processing, our method has the best performance in terms of the dice per case score. Figure 7 reports some visual results of our predictions on the LiTS test set.

**Observations on LiTS Annotations.** As shown in Fig. 6, the LiTS annotations have both over-segmentation and under-segmentation issues. For the large lesions in both cases shown in Fig. 6, our predictions better fit lesions compared to the ground truth; while for the small lesions, our predictions locate more lesions potentially missing in the ground truth. Similar observations have been reported in [4]. Due to the imperfect ground truth provided by the LiTS challenge, we argue that the metrics computed against the ground truth probably could not be the only way to compare the segmentation results. Visual results could be considered as well and ours indicate the efficiency of our method.

4 Discussion and Conclusion

In this paper we proposed a hybrid 2D and 3D neural network for segmenting the liver lesions. Especially, we designed a dedicated 3D segmentation network for
the small lesions in the liver. This 3D segmentation network could be an add on to a network that suffers from segmenting small objects. Moreover, we observed the imperfect annotations provided by the LiTS data set, which hinders the further improvement of a network’s learning performance and makes the evaluation results questionable for comparison. How to handle and fully leverage such imperfect labels will be our future work. We also plan to extend our framework to other medical applications with the lesion/tumor or small object segmentation.

5 Acknowledgement

This work has been supported by NSF 1755970.

References

1. Bilic, P., Christ, P.F., Vorontsova, E., Chlebus, G., et al.: The liver tumor segmentation benchmark (lits). arXiv:1901.04056 (2019)
2. Bradski, G.: The OpenCV Library. Dr. Dobb’s Journal of Software Tools (2000)
3. Chen, X., Zhang, R., Yan, P.: Feature fusion encoder decoder network for automatic liver lesion segmentation. arXiv:1903.11834 (2019)
4. Chlebus, G., Schenk, A., Moltz, J.H., van Ginneken, B., Hahn, H.K., Meine, H.: Automatic liver tumor segmentation in ct with fully convolutional neural networks and object-based postprocessing. Scientific reports 8(1), 15497 (2018)
5. Christ, P.F., Ettlinger, F., Grün, F., Elshaera, M.E.A., et al.: Automatic liver and tumor segmentation of ct and mri volumes using cascaded fully convolutional neural networks. arXiv:1702.05970 (2017)
6. Dey, R., Hong, Y.: Compnet: Complementary segmentation network for brain mri extraction. In: MICCAI. pp. 628–636. Springer (2018)
7. Han, X.: Automatic liver lesion segmentation using a deep convolutional neural network method. arXiv:1704.07239 (2017)
8. Huang, W., Yang, Y., Lin, Z., Huang, G.B., Zhou, J., Duan, Y., Xiong, W.: Random feature subspace ensemble based extreme learning machine for liver tumor detection and segmentation. In: EMBC. pp. 4675–4678. IEEE (2014)
9. Jiang, H., Shi, T., Bai, Z., Huang, L.: Ahcnet: An application of attention mechanism and hybrid connection for liver tumor segmentation in ct volumes. IEEE Access (2019)
10. Jin, Q., Meng, Z., Sun, C., Wei, L., Su, R.: Ra-unet: A hybrid deep attention-aware network to extract liver and tumor in ct scans. arXiv:1811.01328 (2018)
11. Li, S., Tso, G.K.F.: Bottleneck supervised u-net for pixel-wise liver and tumor segmentation. arXiv:1810.10331 (2018)
12. Li, X., Chen, H., Qi, X., Dou, Q., Fu, C.W., Heng, P.A.: H-denseunet: hybrid densely connected unet for liver and tumor segmentation from ct volumes. IEEE transactions on medical imaging 37(12), 2663–2674 (2018)
13. Liu, S., Xu, D., Zhou, S.K., Paut, O., Grbic, S., Mertelmeier, T., Wicklein, J., Jerebko, A., Cai, W., Comaniciu, D.: 3d anisotropic hybrid network: Transferring convolutional features from 2d images to 3d anisotropic volumes. In: MICCAI. pp. 851–858. Springer (2018)
14. Massoptier, L., Casciaro, S.: A new fully automatic and robust algorithm for fast segmentation of liver tissue and tumors from ct scans. European radiology 18(8), 1658 (2008)

15. Moltz, J.H., Bornemann, L., Dicken, V., Peitgen, H.: Segmentation of liver metastases in ct scans by adaptive thresholding and morphological processing. In: MICCAI workshop. vol. 41, p. 195 (2008)

16. Moltz, J.H., Bornemann, L., Kuhnigk, J.M., Dicken, V., et al.: Advanced segmentation techniques for lung nodules, liver metastases, and enlarged lymph nodes in ct scans. IEEE Journal of selected topics in signal processing 3(1), 122–134 (2009)

17. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: MICCAI. pp. 234–241. Springer (2015)

18. Yuan, Y.: Hierarchical convolutional-deconvolutional neural networks for automatic liver and tumor segmentation. arXiv:1710.04540 (2017)