Two-Stage Method for Kidney and Tumor Segmentation Based On Cascade Res-VNet

Xiaoyu Hao[1]

University of Science and Technology of China, Hefei 230026, Anhui, China
hxy2018@mail.ustc.edu.cn

Abstract. In this work, We proposed a two-stage method for segmentation of kidneys and kidney tumors in CT images based on cascade Res-VNet. In the first stage, we consider the kidney and tumor as a whole region and use a cascade Res-VNet to do the segmentation. In the second stage, we firstly extract patches of ROIs based on the results of the first stage and use some tricks to improve precision. Then, we use another cascade Res-VNet to segment the tumor. Ouput from each stage are combined together as the final results.

Keywords: Kidney · Tumor · Cascade · Res-VNet · Segmentation.

1 Materials

1.1 Data

The experimental data is collected from KiTS19[2] challenge dataset. The training and test datasets include 210 and 90 cases, respectively. The CT scan has an in-plane size of 512 x 512 pixels and a spatial resolution between 0.43 mm and 1.04 mm. The number of slices varies from 29 to 1059 with a slice thickness between 0.5 mm and 5 mm. All the datasets

Data preprocessing: In order to highlight the ROI, we chose the kidney window for global intensity value normalization. The window level and window width we chosen are 30 and 310, respectively. The value greater than 185 is set to 185 and the value less than -125 is set to -125. Moreover, we normalized each 3D CT volume by subtracting the mean and dividing the standard deviation, then we use linearly method to convert the data intensity values into range [-1,1].

2 Method

2.1 Multi-Stage Task

In order to get more accurate segmentation results, we divided the automatic segmentation of kidneys and tumors into two stages. In the first stage, considering the tumors always grow inside or around the kidneys, we regarded the kidney

and tumor as a whole region to segment. Then, we performed automatic segmentation of tumors based on the whole region obtained from the first stage and we called this procedure the second stage. We trained the Res-VNet independently for the two stages, and the total procedure is shown in Fig. 1.

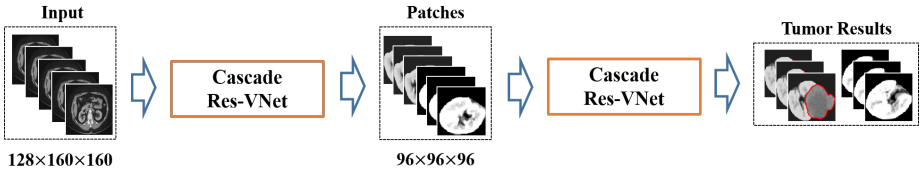

Fig. 1. Pipeline of data preprocessing for the first stage

2.2 Segmentation

We let I be an (CT) 3D volume with corresponding ground-truth K and T donating kidney mask and tumor mask, respectively. Moreover, we let Out_T be the segmentation result of tumor and Out_K be the segmentation result of kidney, Then, we assume H is the union of K and T:

$$H_i = K_i \cup T_i. \tag{1}$$

where i ($i = 1, 2, 3, \dots$) donating the index of cases.

Whole Region of Kidney and Tumor: In the first stage, for the purpose of removing interference of other tissues or organs, we cropped each slice and its corresponding mask by using a same bounding box. The bounding box is the body location of the most middle slice for each 3D volume and it was obtained by using morphological methods. Due to the different dimensions of 3D images, we used the zoom method to unify the size of I and H to 128x160x160. To ensure the accuracy of the learning process, we just let $F_{out} = H$ in training phase, where F_{out} is the output of the first stage. The pipeline of ROIs extraction was shown in Fig. 2.

Tumor: In the second Stage, we only segment the region of tumor. We let I_P and T_P be the input image and annotation of the second stage. I_P and T_P for each case is calculated by the following steps:

First, we selected the largest two remaining connected domains and calculated their bounding boxes which are defined as B_1 and B_2. We used these two bounding boxes to locate the kidneys. Sometimes, only one connected domain will be reserved because there is only one kidney can be seen in the CT image and in this case, we will let $B_2 = B_1$.

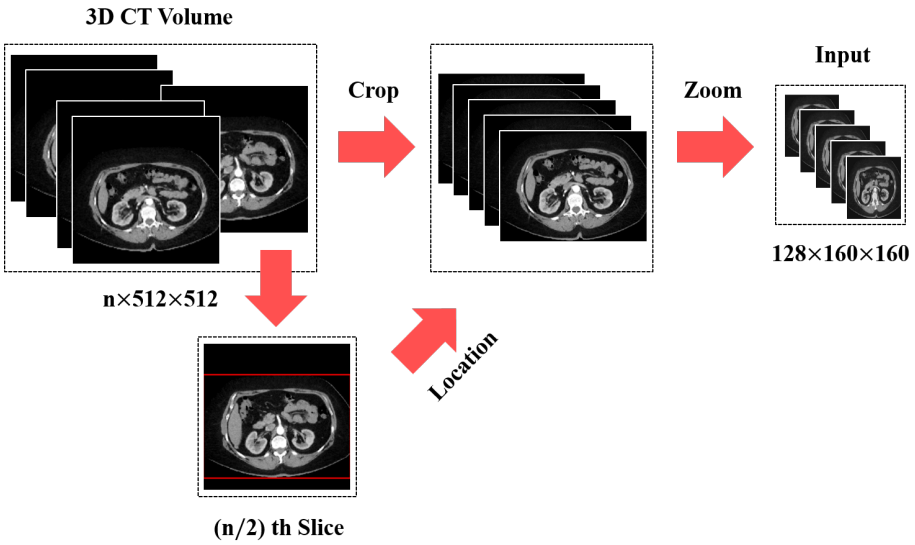

Fig. 2. Pipeline of ROIs extraction for the first stage

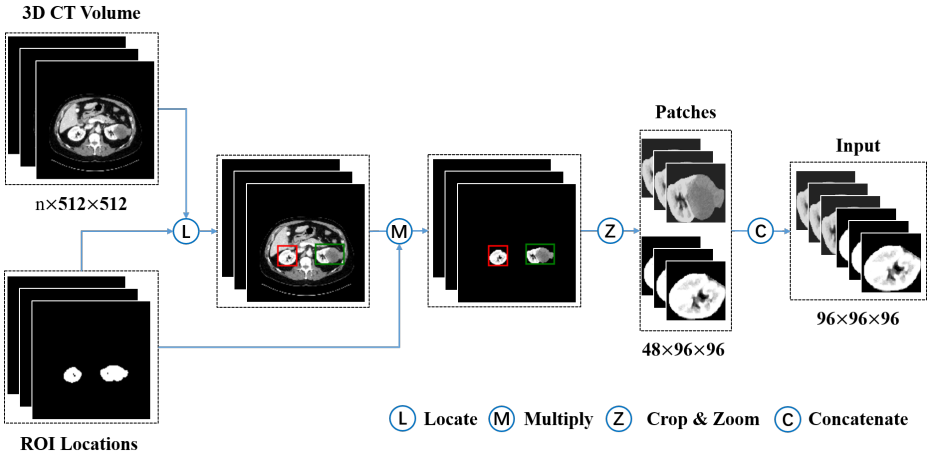

Fig. 3. Pipeline of ROIs extraction for the second stage

Second, we zoomed F_{out} to the origin size and calculated all connected domains of F_{out}, then we deleted the connected domains whose volume were less than the threshold. After some experiments, we set the threshold to 100. Let I_T be the result of multiplying I and F_{out}:

$$I_T = I * F_{out} \tag{2}$$

This method is applied to clear interference from other tissues and organs.

Third, I_T and T were cropped according to the coordinates and size of the bounding boxes (B_1 and B_2). The patches of I_T are defined as P_1, P_2, and patches croped from T are T_1, T_2. Because of the different size of connected domains, we need to zoom those patches into same size. After analyzing all the training data, we choose to zoom the patches to 48x96x96.

At last, we concatenated P_1, P_2 as I_P and concatenated T_1, T_2 as T_P in the first axis (the order is random) to obtain ROIs and their corresponding annotations. These ROIs were used to segment tumors in the second stage. And we let S_{out} donate the output of the second stage. Fig. 3. shows the pipeline of ROIs extraction.

Results Merge: We used two trained networks to get F_{out} and S_{out} sequentially. After that, we used the intersection of F_{out} and S_{out} as the final result of tumor:

$$Out_T = F_{out} \cap S_{out} \tag{3}$$

Out_K was obtained by calculating the difference set between F_{out} and Out_T:

$$Out_K = F_{out} - Out_T \tag{4}$$

2.3 Network Architecture:

Res-VNet: Our network was inspired by VNet[5] architecture which is widely used in medical image segmentation. VNet is composed of two paths, the contraction path on the left is used to extract high-level context information. The right expanding path uses skip connections to fuse high-level context features with fine-grained local features for precise contour localization.

The expressive power of the network is enhanced as the depth of the network increases. In order to get more accurate segmentation, we cascaded two networks together in sequence. To a certain extent, this approach increases the depth of the network. In order to avoid degradation of network and maintain the expressive power of the network, we added residual structures[2] to our architecture. Due to the use of residual block, the architecture is known as Res-VNet.

Moreover, we replaced the max pooling layers in the network by convolution layers whose kernel size is 2x2x2 and stride is 2 to avoid the loss of information of small ROI. To capture multi-scale context information, we replaced the convolutions in the last two blocks of the contraction path with dilated convolutions. Fig. 4. shows the architecture of Res-VNet.

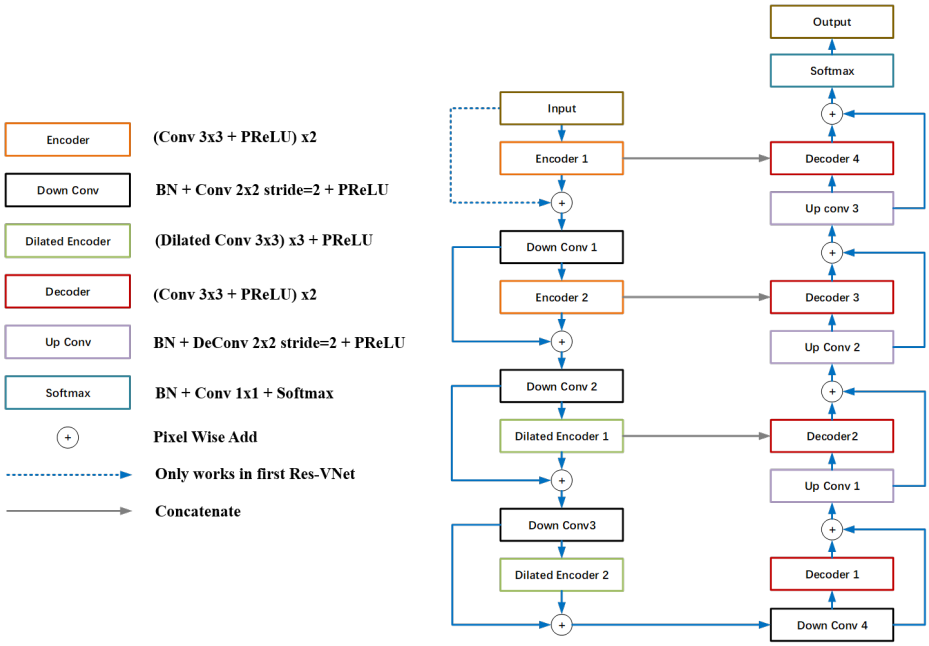

Fig. 4. Res-VNet Architecture

Cascade Network: In order to improve the segmentation accuracy, we cascaded two ResVNet together. During both train and inference procedures, the input images were narrowed down to 1/2 of its original size to reduce the cost of gpu memory. The output will be doubled and concatenated with original image as the input of the second Res-VNet.The architecture of our proposed cascaded Res-VNet is shown in Fig. 5.

2.4 Data Augmentation:

We used data augmentation to reduce overfitting. During training, the input images and labels were randomly rotated from -90 to 90 degrees and flip around a random axis with a probability of 0.5.

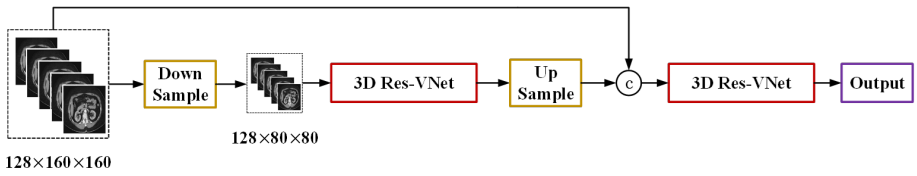

Fig. 5. Cascade Res-VNet Architecture

3 Experiments

3.1 Training Details

The implementation of our networks is based on PyTorch-1.1.0. We used Adam as optimizer and set 1e-3 as the initial learning rate. We totally trained the networks for 500 epochs. When training the network for segmentation of the whole region of kidney and tumor, the learning rate was decayed by multiplying 0.90 every 10 epochs and the batch size was set to 2. For the networks of segmenting tumor, the learning rate will decay by multiplying 0.94 every 5 epochs and the batch size is 4. All the networks were trained by using NVIDIA Tesla V100 GPU.

3.2 Loss Function

We trained the network by minimizing a dice loss function:

$$dice = 1 - \frac{2 * |P| * |G|}{|P|^2 + |G|^2} \tag{5}$$

where P is the results of prediction and G is the ground truth. It is the most common loss function in the field of medical imaging.

As mentioned in 2.3, the data was trained by a cascade network so we independently calculated the loss of each Res-VNet and joint them as follow to be the final loss:

$$\ell = \frac{d_1 + d_2}{2} \tag{6}$$

where d_1 and d_2 seperately donate the dice of each Res-VNet.

References

1. Ronneberger, O., Fischer, P., & Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention (pp. 234-241). Springer, Cham.
2. Heller, N., Sathianathen, N., Kalapara, A., Walczak, E., Moore, K., Kaluzniak, H., ... & Dean, J. (2019). The KiTS19 Challenge Data: 300 Kidney Tumor Cases with Clinical Context, CT Semantic Segmentations, and Surgical Outcomes. arXiv preprint arXiv:1904.00445.
3. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
4. Yang, G., Li, G., Pan, T., Kong, Y., Wu, J., Shu, H., ... & Zhu, X. (2018, August). Automatic Segmentation of Kidney and Renal Tumor in CT Images Based on 3D Fully Convolutional Neural Network with Pyramid Pooling Module. In 2018 24th International Conference on Pattern Recognition (ICPR) (pp. 3790-3795). IEEE.
5. Milletari, F., Navab, N., & Ahmadi, S. A. (2016, October). V-net: Fully convolutional neural networks for volumetric medical image segmentation. In 2016 Fourth International Conference on 3D Vision (3DV) (pp. 565-571). IEEE.