SEGMENTATION OF THORACIC ORGANS AT RISK IN CT IMAGES COMBINING COARSE AND FINE NETWORK

Li Zhang, Lishen Wang, Yijie Huang, Huai Chen

Shanghai Jiao Tong University

ABSTRACT

Segmentation of multiple thoracic organs from three dimensional Computed Tomography (CT) images is challenging due to its huge computation and indistinguishable features. In this paper, we propose a novel multitask framework where coarse segmentation network and fine segmentation network share the same encoder part. We firstly use coarse segmentation network to obtain the regions of interest(ROIs) localization, and then crop multi-level ROIs from the encoder part to form decoder for detail-preserving segmentation. We apply our proposed method on test data set and achieve good results.

***Index Terms*—** Multiple organs, Coarse segmentation, Fine segmentation

1. INTRODUCTION

As we all know, radiation therapy is preferred when treating lung and esophageal cancer. Physicians would delineate the target tumor manually, which is usually located between normal organs, called Organs at Risk (OAR). The common way for avoiding errors is to segment these organs on Computed Tomography (CT) images firstly, so this task is of great importance to surgeries.

However, the segmentation of multiple organs is challenging: (1) The data to be processed is of large amount and the computation is huge since medical images are three-dimensional; (2) Some organs are neighbored closely and their contours in CT images have low contrast; (3) Some organs' shapes and locations differ greatly between patients.

In this paper, our goal is to automatically segment four kinds of thoracic organs at risk in CT images: heart, aorta, trachea, esophagus. We design a kind of deep neural network to segment these organs at the same time, which achieving a relatively good result in ISBI 2019, Challenge 4: SegTHOR: Segmentation of THoracic Organs at Risk in CT images[1].

2. OUR METHOD

2.1. Data preprocessing

The CT scans have the same resolution 512×512, but their in-plane resolution varies per pixel and z-resolution is also nonuniform. So the first step is to let these samples have the same spacing. Considering the distribution of these samples, we set the uniform spacing to [1.0, 1.0, 3.0], which corresponds to x axis, y axis, z axis.

However, due to the limitation of GPU memory, it's impossible for us to import the data directly into the deep learning model. So the next step is to clip the CT images roughly by using ostu threshold. We calculate the threshold value by applying ostu method to the middle scan of every sample. Then we use this threshold value to carry on binary segmentation to remove background voxels, which greatly reduces the amount of the data.

2.2. Segmentation model

Although we preprocess the CT images, the large amount of data still troubles us. In order to obtain more accurate results, we adopt two-stage methods. Firstly we use coarse segmentation to get the regions of interest(ROIs), then we apply fine segmentation in these regions to get final results. Our experiments have demonstrated that our method is robust and effective.

Our overall network consists of two parts: encoder part and decoder part. Since encoder part is to extract features from CT images, coarse segmentation and fine segmentation have the same encoder part but have different decoder part.

2.2.1. Coarse segmentation

Supposed the input resolution is [X,Y,Z], which corresponds to x axis, y axis, z axis of CT images. As is shown in Fig. 1, the outcome feature maps from the encoder part have 256 channels and their resolution is [X/8,Y/8,Z/2]. In the encoder part, there are one separate convolution operation and four ResBlocks[2] which consist of two or three convolution operations. Considering some organs have large span, we introduce dilation convolution operations in the latter two ResBlocks to enlarge the receptive field. We set dilation d_1=3 in shallower layer while setting dilation $d_2 = 6$ in deeper layer.

In decoder part, for the purpose of coarse segmentation, we execute convolution operation and simgoid function for the outcome feature maps form latter three ResBlocks respectively[3]called F_1,F_2, F_3. Then we calculate the aver-

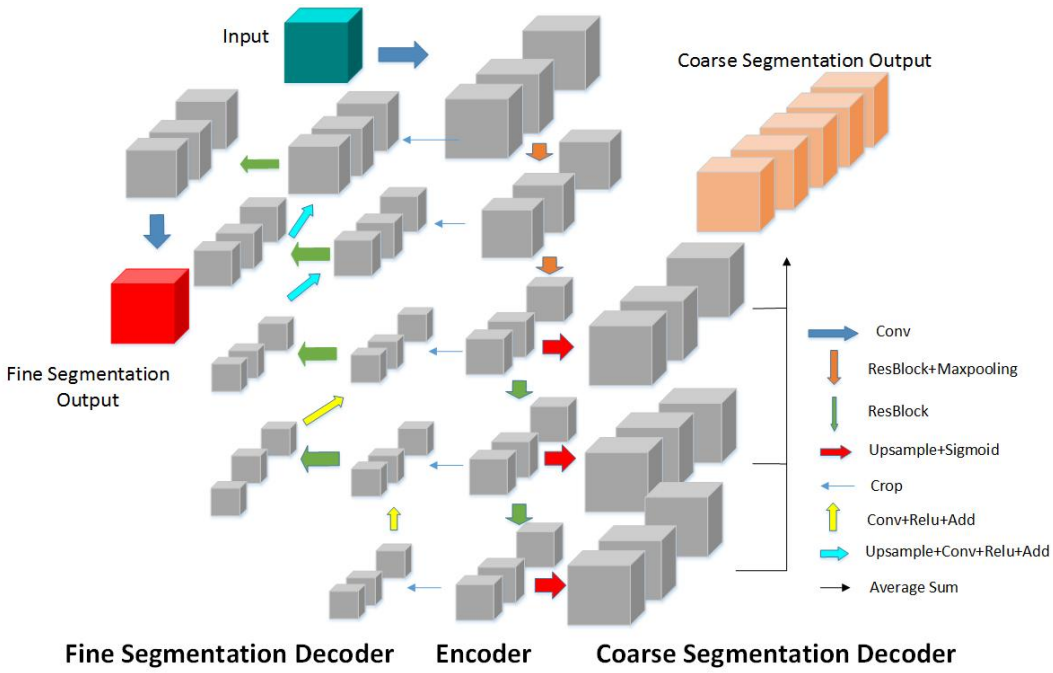

Fig. 1. The illustration of our overall framework.

age sum of these three new feature maps: F_{ave}. F_{ave} have five channels, which denote the background and four organs respectively.

So, supposed the responding ground truth is F_{ori}, we can construct the corresponding loss function as followed:

$$Loss = 1 - 2 \times \frac{F_{ave} \bigcap F_{ori}}{F_{ave} \bigcup F_{ori}}$$

As for every kind of organ, we use this loss function to train. In the end, corresponding four channels display the coarse segmentation of target organs.

2.2.2. *Fine segmentation*

Based on coarse segmentation output, we can get the rough location of every kind of organ. So we don't need to use full information of encoder part to decoder to obtain the fine segmentation. Firstly, we can get the three-dimensional bounding box of target organs in original shape. Then we can adjust the bounding box size of each layer in the encoder part according to the resolution. We crop the voxels of corresponding bounding boxes to obtain ROIs and carry out fine segmentation decoder.

As is shown in the left part of Fig. 1, after convolution and Relu function, the deeper features are added to the upper features to execute ResBlock operation, which allows the network to propagate context information to higher resolution layers[4]. So it is reasonable that we can capture features of various sizes and obtain fine segmentation. It is worth noting that we use different decoder paths for every kind of organ, since: (1) The parameters and computation have already decreased greatly by only processing ROIs; (2) Each kind of organ has distinctive features.

The construction of the loss function is the same as the above. In order to be more effective when learning, we adopt hard voxel mining method, that is to exert higher weights to these error-prone voxels. It is not difficult to find that we can obtain better results.

3. EXPERIMENTS

3.1. Implementation details

We distribute our model on two NVIDIA TITAN X and our implementation is based on Pytorch. We adopt $3 \times 3 \times 3$ kernel size in every convolution operation and use dilation $d_1 = 3$,$d_2 = 6$ in the latter two layers in encoder part. When learning, we firstly train the coarse segmentation network including encoder part and its decoder part individually for 40 epochs. Then we train coarse segmentation network and fine segmentation network together for 50 epochs.

3.2. Results on test data

We apply our proposed method on test data set and the results can be seen in Table. 1.

Table 1. Our method on test data.

Organ	Dice	Hausdorff
Esophagus	0.7732	1.6774
Heart	0.9384	0.2089
Trachea	0.8939	0.2741
Aorta	0.9232	0.3081

4. REFERENCES

[1] Roger Trullo, C. Petitjean, Su Ruan, Bernard Dubray, Dong Nie, and Dinggang Shen, "Segmentation of organs at risk in thoracic CT images using a sharpmask architecture and conditional random fields," in *IEEE 14th International Symposium on Biomedical Imaging (ISBI)*, 2017, pp. 1003–1006.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[3] Jonathan Long, Evan Shelhamer, and Trevor Darrell, "Fully convolutional networks for semantic segmentation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[4] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, Eds., Cham, 2015, pp. 234–241, Springer International Publishing.