TWO-STAGE NETWORK FOR OAR SEGMENTATION

Pan Chen, Chenghai Xu, Xiaoying Li, Yingying Ma, Fenglong Sun
Digital China Health Technologies Co., Ltd.

ABSTRACT

For cancer, radiotherapy is a standard treatment and the first step is to identify the target volumes to be targeted and the healthy organs at risk (OAR) to be protected. In this paper, we propose a deep learning framework for the segmentation of OAR in CT images of the thorax, specifically the heart, the esophagus, the trachea and the aorta. The key idea is using a two-stage strategy. First, we use a 3d U-shape convolution network to get the localization of four organs. Then using next 3d U-shape convolution network we obtain the precise segmentation results. Also, we try some tricks to improve the performance.

Index Terms— OAR segmentation, deep learning, two-stage

1. INTRODUCTION

Delineating organs in risk is a very important routine work in radiation therapy. However, current manual delineation is time consuming and relies on the knowledge and experience of a doctor. In this paper, inspired by [1], we introduce a two-stage OAR segmentation framework based on DCNN, consisting of 1) a localization model to detect the interest of region containing the four organs: esophagus, heart, trachea, aorta; 2) a segmentation model to focus on this interest of region and obtain the segmentation result. Since the CT image is three dimensional volume data and these organs are intrinsically 3d objects, DCNN filters learning directly on the overall 3d CT volume enables capturing the complete spatial context of organs. But due to the computational intensity and limit amount of GPU memory, the input image can't be very large. Our method has two advantages: on the one hand, we don't need high resolution image for localization task; on the other hand, the segmentation model can only concentrate on the important local region. Both reduce the sizes of input images and make the algorithm efficient.

2. METHOD

As mentioned above, we first determine the region where OAR are located. Then we focus on this region to get fine segmentation.

2.1. Localization

2.1.1. Data preprocessing

With CT intensity values being not standardized, normalization is critical to allow for data from different processing types (with or without IV contrast). we normalize each image independently by subtracting the mean and dividing by the standard deviation.

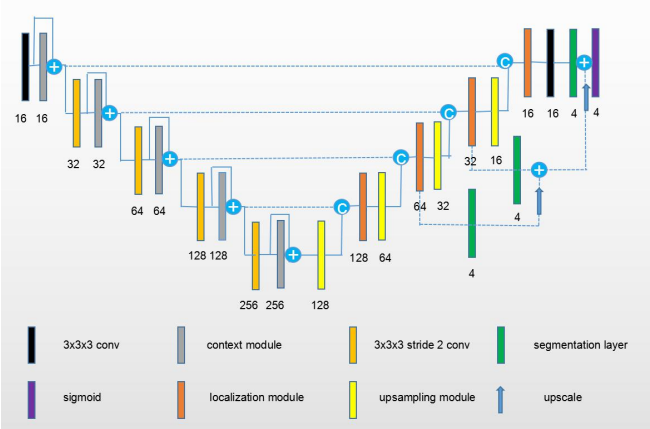

Figure 1. Network architecture. The number below the box refers to the number of output channels

2.1.2. Network architecture

Our network is inspired by the U-Net architecture [2] and we refer to [3]. See Figure 1. Our architecture comprises of a context pathway and a localization pathway. In the context pathway, residual blocks that we call context modules are used. Every context module consists of two 3x3x3 convolutional layers and a dropout layer (p=0.3) is in between. Context modules are connected by stride 2 3x3x3 convolutions. As going deeper, the context modules give more abstract representations of lower image resolution. In the localization pathway, we have three localization modules. Every localization module consists of a 3x3x3 convolution, which halves the number of channels, followed by a 1x1x1 convolution. The upsampling module consists of an upsampling (size 2, stride 2) and a 3x3x3 convolution which halves the number of channels. Through upsampling and concatenating, a merged feature map is send to the localization module. We employ deep supervision in the localization pathway by integrating segmentation layers (3x3x3 convolution) at different levels of the network and combining them via element-wise summation to form the final network output. Throughout the network we use leaky ReLU non-linearities for all feature maps. We furthermore replace the traditional batch

normalization before activation functions with instance normalization in the case of small batch size.

2.1.3. Training procedure

Our network architecture is trained with 128x128x128 voxels obtained by zooming the original CT images and batch size 1, due to the memory limitation of GPU. The employed optimizer is *Adam* with an initial learning rate $lr=5x10^{-4}$, the following learning rate schedule: after every epoch, the learning rate is reduced to be 98.5% of the original and a L_2 weight decay of 10^{-5} .

In order to deal with the class imbalance in the data, we use a multiclass Dice loss function, i.e. the average of Dice loss functions of the four classes. During training, flipping at three axes on the fly is applied to prevent overfitting. Note that we segment the image to obtain the localization.

2.1.4. Post-processing

At inference phase, the prediction is upsampled by a nearest neighbor interpolation to match the shape of the original input scan. After that, for each organ, we remove the connect regions whose volumes are smaller than 5% of the volume of the maximal connect region.

2.2. Segmentation

2.2.1. Training data

In essence, we use the same network architecture and same training procedure to obtain the segmentation model except for using different training data. Noting that the training data of the localization model is zoomed from the original CT image, so its resolution is lower, which is enough for localization, but not good for segmentation.

For saving the resolution as much as possible, we clip the original data by extending the bounding box which exactly contains the groundtruth with 10 pixels in all directions. If one side of this clipped cube is smaller than 128, we replace it by 128. That is, using a bounding box of shape 128x128x128 to be the lower bound. As additional data argument, bounding boxes of shape 200x200x150, and 256x256x200 are also used. Then resize this clipped cube to 128x128x128 as our input of segmentation module.

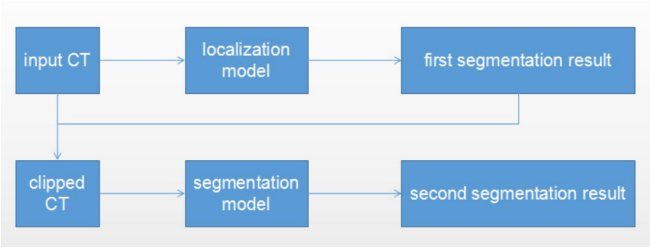

Figure 2. two-stage segmentation

2.2.2. Inference procedure

At inference phase of our two-stage segmentation, we first use the localization model to get a rough localization of four organs, and basing on this information to clip the original CT image and resize the clipped cube to 128x128x128. Then send this resized cube to the segmentation model to obtain the fine segmentation result, which becomes the final result after post-processing and returning to the original image. See Figure 2.

3. RESULTS

We trained and evaluated our network on the SegTHOR 2019 training dataset (40 CTs) and test dataset (20 CTs)[4]. No external data was used and the network was trained from scratch. To improve performance further, we added a multi-test step on the esophagus, and consider more detailed post-processing on the segmentation of heart and esophagus. It takes about 1 minute for our algorithm to obtain the segmentation result from one raw CT image. Table 1 gives the performance of our algorithm on the test dataset.

	Dice				Hausdorff			
	esophagus	heart	trachea	aorta	esophagus	heart	trachea	aorta
score	0.8166	0.9329	0.8910	0.9232	0.4914	0.2417	0.2746	0.3081

Table 1.

4. REFERENCES

[1] Roger Trullo, et al. "Fully automated esophagus segmentation with a hierarchical deep learning approach." 2017 IEEE International Conference on Signal and Image Processing Applications (ICSIPA) IEEE, 2017.

[2] Ronneberger, O., Fischer, P., & Brox, T., "U-net: Convolutional networks for biomedical image segmentation," In International Conference on Medical image computing and computer-assisted intervention, Springer, Cham, pp. 234-241, October 2015.

[3] Isensee, Fabian , et al. "Brain Tumor Segmentation and Radiomics Survival Prediction: Contribution to the BRATS 2017 Challenge." International MICCAI Brainlesion WorkshopSpringer, Cham, 2017.

[4] Roger Trullo, Caroline 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 International Symposium on Biomedical Imaging (ISBI), pp. 1003-06, 2017.