Sequence and Spatial Feature Fusion for Kidney and Tumor Segmentation From CT Volumes

Han Wenzhong[1] and Kang Li[1]

[1] Shenzhen University, Shenzhen 518060, China
hanwinz@163.com

Abstract. Morphological heterogeneity makes the precise segmentation of both the tumor and the kidney becomes a problem for diagnosis and quantitative analysis of the next treatment. In this work, we propose an encoder-decoder architecture which consist of recent neural network for downsampling and 3D convolutional neural network for upsampling. Recent neural network can integrate contextual information between slices and 3D convolutional neural network can make use of spatial information of one case.

Keywords: Recent neural network, Convolutional neural network, Contextual information, spatial information.

1 Introduction

Accurate segmentation of tumor tissue and kidney in CT images can help to make accurate pathological staging, provide a basis for quantitative analysis of volume calculations in the lesion area, and then provide assistance for radiation dose calculation and treatment planning.

In the past decades, a lot of algorithms, including thresholding, region growing, deformable model based methods and machine learning based methods have been proposed to segment lesion in CT image. In recent years, convolutional neural networks (CNNs) have advanced the state of art in image segmentation tremendously as this technique allows to learn rich feature representations over multiple scales. However, the integration of these representations for obtaining full-resolution segmentations is not straightforward and it is an active field of research. To solve this problem, Fausto Milletari *et al.* [1] presented an approach which is based on a fully convolutional architecture consisting of an encoding and a decoding part. While encoding is based on a standard ResNet architecture, decoding is implemented using convolutional LSTMs. [2]The core idea of this approach is to use a memory mechanism, implemented via convolutional LSTMs, for fusing features extracted from different layers of the encoder. However, this approach may lose some spatial information.

In this work, we put conLSTM layers in the encode stage to catch features between slices and put the conv3D layers in the decode stage to catch spatial information which keep some global features.

2 Method

2.1 Data preprocess

For image preprocessing, we truncated the image intensity values of all scans to the range of [-200,300] HU to remove the irrelevant details. And we normalize every images to makes network training easier to converge. In the first stage, we trained a network from resampled cases to attain coarse kidney segmentation . For tumor segmentation in the second stage, the network is trained on the images with the original resolution. This is because in some training cases tumors are notably small, thus we use images with the original resolution to avoid possible artifacts from image resampling. In test stage, we also employ the images with original resolution for accurate kidney and tumor segmentation.

2.2 Encoder-decoder architecture

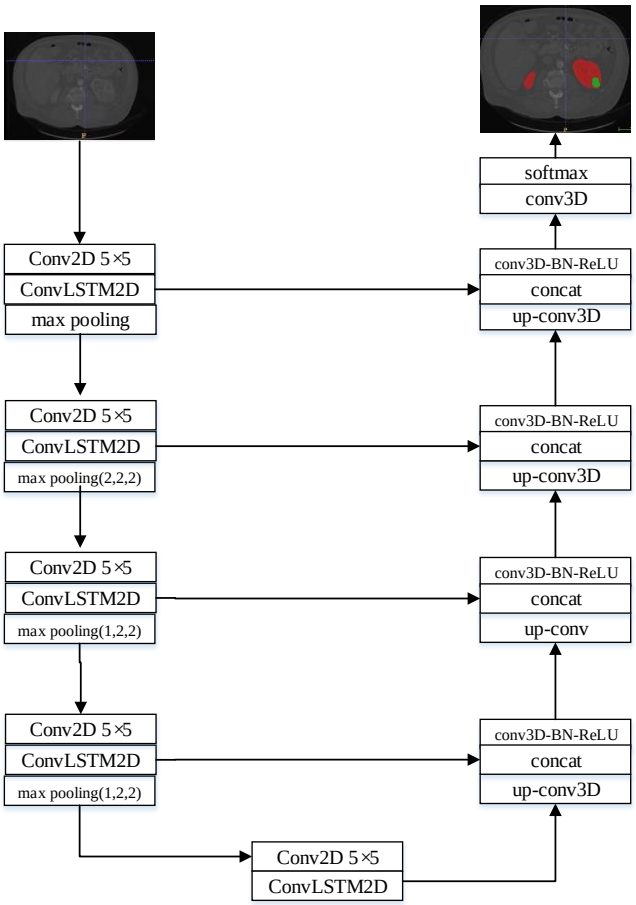

Fig. 1. Network designed in this work. In the encode stage, we use conv2D with 5×5 kernel while in the decode stage, 3×3 kernel is used.

As shown in Fig. 1, we propose an encoder-decoder architecture like Unet.[3] We employ convolutional LSTMs in the encode stage, which have the capability of selectively updating their internal states at each step depending on the result of a convolution. The upsampling layer is implemented by the bilinear interpolation, followed by the concatenation with low-level features catched by conLSTM. Then a 3×3×3 convolutional layer is used. Before each convolution layer, the batch normalization and the Rectified Linear Unit (ReLU) are employed in the architecture. It is worth noting that the larger kernels(5 × 5) is used to gain larger receptive field for downsampling related with low-level features. But the larger kernels(3×3) is used in the decode stage to reduce the amount of parameters in the network. The detailed structure of our network is shown in Table I.

Table 1. architecture of this work.

encode layers	output	decode layers	output
input	16×224×224×1	upsampling3d_1 (1,2,2)	8×28×28×256
conv2d_1 (5,5)	16×224×224×32	conv3d_1 (2,2,2)	8×28×28×256
convLSTM2d_1	16×224×224×32	concatenate_1	8×28×28×512
maxpooling3d_1 (2,2,2)	8×112×112×32	conv3d(3,3,3)-bn-relu_1	8×28×28×256
conv2d_2 (5,5)	8×112×112×64	upsampling3d_2 (1,2,2)	8×56×56×256
convLSTM2d_2(3,3)	8×112×112×64	conv3d_2 (2,2,2)	8×56×56×128
maxpooling3d_2 (1,2,2)	8×56×56×64	Concatenate_2	8×56×56×256
conv2d_3 (5,5)	8×56×56×128	conv3d(3,3,3)-bn-relu_2	8×56×56×128
convLSTM2d_3(3,3)	8×56×56×128	upsampling3d_3 (1,2,2)	8×112×112×128
maxpooling3d_3 (1,2,2)	8×28×28×128	conv3d_3 (2,2,2)	8×112×112×64
conv2d_4 (5,5)	8×28×28×256	concatenate _3	8×112×112×128
convLSTM2d_4(3,3)	8×28×28×256	conv3d(3,3,3)-bn-relu_3	8×112×112×64
maxpooling3d_4 (1,2,2)	8×14×14×256	upsampling3d_4 (1,2,2)	16×224×224×64
conv2d_5 (5,5)	8×14×14×512	conv3d_4 (2,2,2)	16×224×224×32
convLSTM2d_5(3,3)	8×14×14×512	concatenate _4	16×224×224×64
		conv3d(3,3,3)-bn-relu_4	16×224×224×32
		conv3d_5 (1,1,1)-softmax	16×224×224×3

2.3 Loss function and training details

To train the networks, we employed cross-entropy function as the loss function, which is described as:

$$c = -\frac{1}{n}\sum_{x}\left[y \ln a + \left(1 - y\right) \ln \left(1 - a\right)\right] \qquad (1)$$

where a denotes the probability of voxel x belongs one class (background, kidney or lesion), and y indicates the ground truth label for voxel x .

we used Adam Optimizer and set the learning rate as 0.00005. The batch_size was setted as 4.The input_size was setted as 16×224×224.

References

1. Milletari F, Rieke N, Baust M, et al. CFCM: Segmentation via Coarse to Fine Context Memory[J]. 2018.
2. Shi X, Chen Z, Wang H, et al. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting[J]. 2015.
3. Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation[C]//International Conference on Medical image computing and computer-assisted intervention. Springer, Cham, 2015: 234-241.