Application of Efficient Context-Aware Network for Pulmonary Airway

Yuntao Zhu[1 [0000−0003−2816−2709]] and Liwen Zou[1]

Abstract. This is a technical report about application of efficientSegNet network.Main component derived from origin paper. We use the **efficientSegNet** network, which is composed of basic encoder, slim decoder and efficient context block. For the decoder module, anisotropic convolution with a k*k*1 intra-slice convolution and a 1*1*k inter-slice convolution, is designed to reduce the computation burden. For the context block, we propose strip pooling module to capture anisotropic and long-range contextual information. This method place on the 2021-MICCAI-FLARE challenge. Codes and models are available at https://github.com/Shanghai-Aitrox-Technology/EfficientSegmentation

Keywords: EfficientSegNet · Deep Learning · Pulmonary Artery.

1 Introduction

In this paper, we focus on Pulmonary Airway segmentation from abdominal CT scans by efficientSegNet[4]. The main difficulties stem from these aspects: 1) The variations in field-of-views, shape and size of different vessel. 2) The limited GPU memory size and high computation cost.

A common solution [2] is to develop a sliding-window method, which can balance the GPU memory usage. Usually, this method need to sample sub-volumes overlap with each other to improve the segmentation accuracy, while leading to more computation cost. Meanwhile, sub-volumes sampled from entire CT volume inevitably lose some 3D context, which is important for distinguishing multi-organ with respect to background.

We develop a whole-volume-based coarse-to-fine framework [5] to effectively and efficiently tackle these challenges. The coarse model aims to obtain the rough location of target organ from the whole CT volume. Then, the fine model refines the segmentation based on the coarse result. This coarse-to-fine pipeline can cover anatomical variations for different cases. To capture the spatial relationships between multi-organ, we exploit strip pooling [1] for collecting anisotropic and long-range context. This strip pool offers two advantages. Firstly, compared to self-attention or non-local module, strip pool consumes less memory and matrix computation. Secondly, it deploys long but narrow pooling kernels along one spatial dimension to simultaneously aggregate both global and local context.

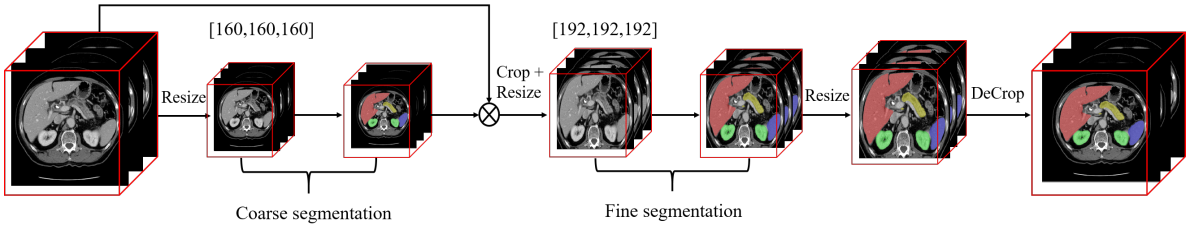

Fig. 1. A schematic diagram of whole-volume-based coarse-to-fine segmentation framework.

2 Method

As mentioned in Figure 1, this whole-volume-based coarse-to-fine framework is composed of coarse and fine segmentation with a basic U-Net and a carefully designed **efficientSegNet**, respectively. A detail description of the method is as follows.

2.1 Preprocessing

The baseline method includes the following preprocessing steps:

- Reorienting images to the left-posterior-inferior (LPI) view by flipping and reordering.
- Resampling image to fixed size. The sizes of coarse and fine input are [160, 160, 160] and [192, 192, 192], respectively.
- Intensity normalization: First, the image is clipped to the range [-600, 400]. Then a z-score normalization is applied based on the mean and standard deviation of the intensity values.

2.2 Proposed Method

The proposed **efficientSegNet** consists of three major parts: basic encoder, slim decoder, and efficient context block, as shown in Figure 2.

As depicted in Figure 3 and Figure 4, the encoder module is composed of two residual convolution blocks, and the decoder module with one residual convolution block. As to decoder module, we separate a standard 3D convolution with kernel size $3\times3\times3$ into a $3\times3\times1$ intra-slice convolution and a $1\times1\times3$ inter-slice convolution. The residual convolution block is implemented as follows: conv-instnorm-ReLU-conv-instnorm-ReLU (where the addition of the residual takes place before the last ReLU activation).

We adopt 3D-based mixed pyramid pooling (Figure 5) to extract contextual feature, which is composed of the standard spatial pooling and the anisotropic strip pooling. The standard spatial pooling employs two average pooling with the stride of $2\times2\times2$ and $4\times4\times4$. The anisotropic strip pooling with three different-direction receptive fields:

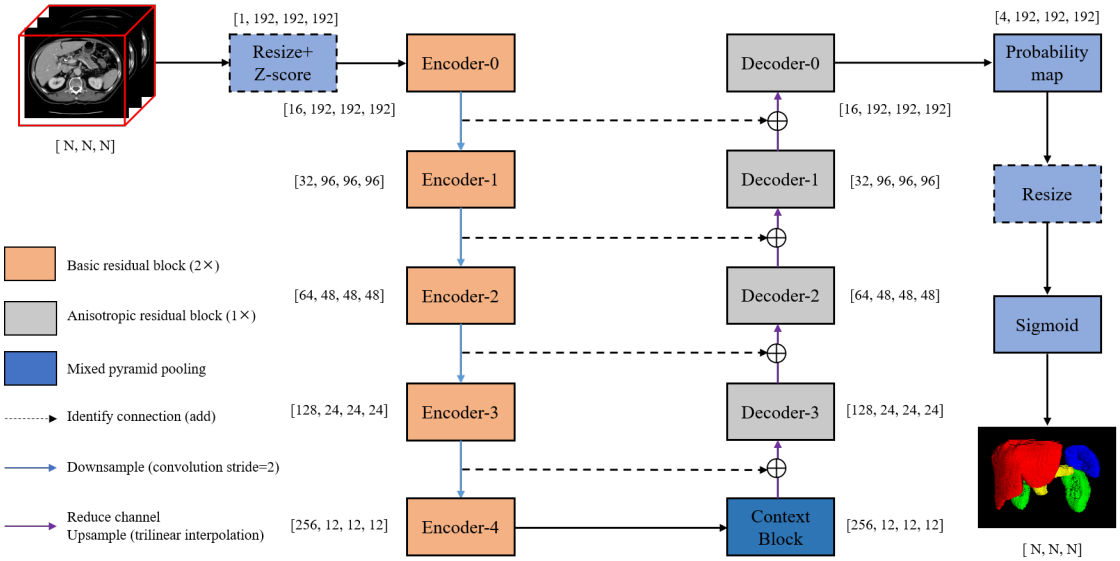

Fig. 2. Illustration of the proposed efficientSegNet.

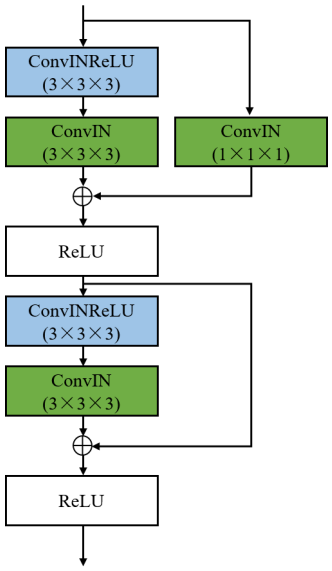

Fig. 3. Illustration of the encoder block.

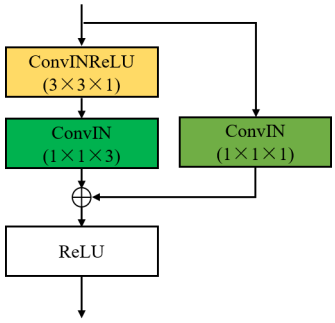

Fig. 4. Illustration of the decoder block.

1×N×N, N×1×N and N×N×1, where N is the size of feature map in last encoder module.

The initial number of feature maps is 8 for coarse model, while 16 for fine model. We aggregate low and high level feature with addition rather than concatenation, because the former consumes less GPU memory. In addition, the number of model parameters is 9 MB, and the number of flops is 333 GB for 192×192×192 input size.

2.3 Post-processing

A connected component analysis of segmentation mask is applied on coarse and fine model output.

3 Dataset and Evaluation Metrics

3.1 Dataset

A short description of the dataset used:

- 500 CT scans (300 for training, 50 for validation, 150 for testing) are collected from multi-sites. The thoracic CT scans are collected from the public LIDC-IDRI dataset [2] and the Shanghai Chest hospital. Each thoracic CT scan is first preprocessed by some strong deep learning models and ensemble strategy to acquire the preliminary segmentation result and then carefullt delineated and double-checked by three radiologists with more than five years of professional experience to acquire the final refined airway tree structure.
- The 300 CT scans for training contain the image.nii.gz along with its binary airway mask (label.nii.gz).
- 50 CT scans for validation contain the image.nii.gz.
- The 150 CT scans for testing are kept private by the organizers.

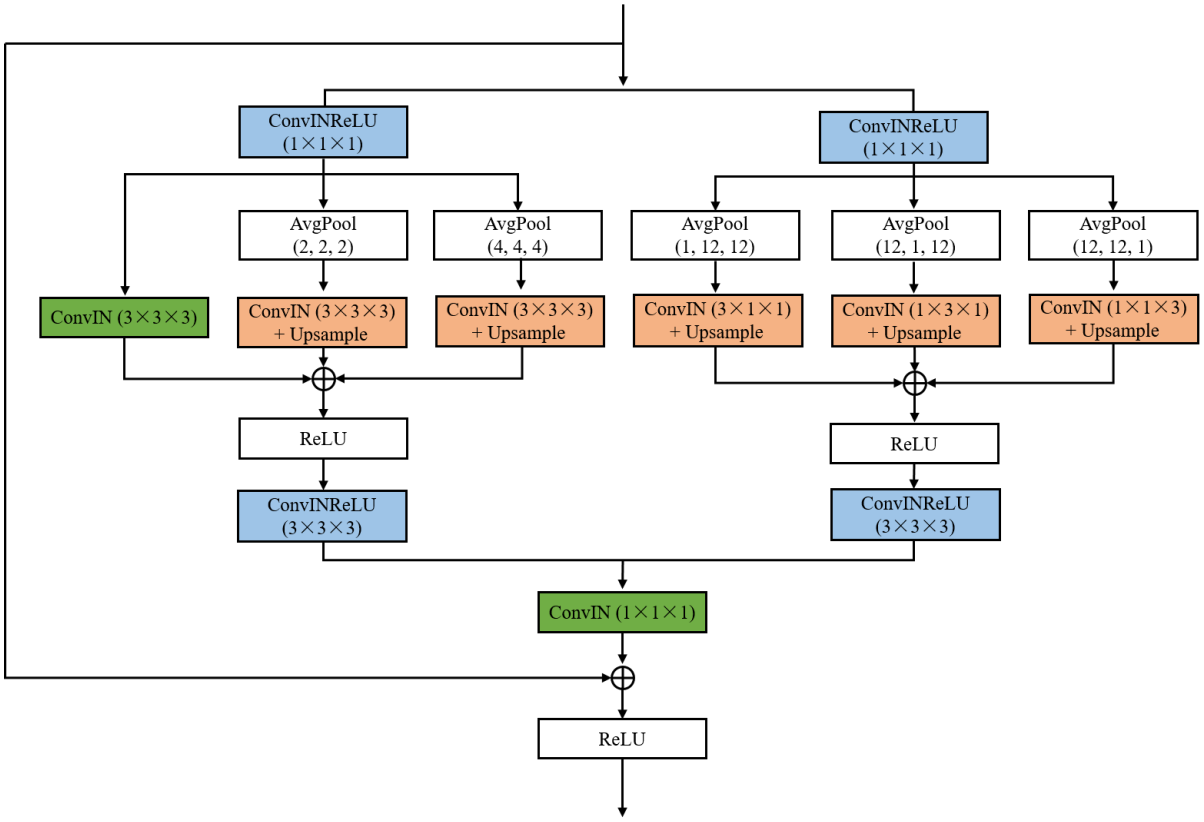

Fig. 5. Illustration of the context block.

Table 1. Environments and requirements.

Ubuntu version	20.04.12
CPU	Intel(R) Xeon(R) Gold 5118 CPU @ 2.30GHz (×4)
RAM	502 GB
GPU	Nvidia GeForce 2080Ti (×8)
CUDA version	11.3
Programming language	Python3.8
Deep learning framework	Pytorch (torch 1.9.0, torchvision 0.2.2)
Code is publicly available at	EfficientSegmentation

Table 2. Training protocols.

Data augmentation methods	Crop and brightness.
Initialization of the network	Kaiming normal initialization
Patch sampling strategy	Augment the sample ratio of pathological image (3 times)
Batch size	16
Patch size	Coarse: 160×160×160 Fine: 192×192×192
Total epochs	200
Optimizer	Adam with betas (0.9, 0.99), L2 penalty: 0.00001
Loss	Dice loss and focal loss (alpha = 0.5, gamma = 2)
Dropout rate	0.2
Initial learning rate	0.01
Learning rate decay schedule	Step decay
Stopping criteria, and optimal model selection criteria	Stopping criterion is reaching the maximum number of epoch (200).
Training mode	Mixed precision
Training time for coarse model	3 hours
Training time for fine model	6 hours

4 Implementation Details

4.1 Environments and requirements

The environments and requirements of the proposed method is shown in Table 1.

4.2 Training protocols

The training protocols of the proposed method is shown in Table 2.

4.3 Testing protocols

The same pre-process and post-process methods are applied as training steps. In order to reduce the time cost of pre-process and post-process, resample and intensity normalization are computed in GPU. We implement the connected component analysis in C++ library, namely cc3d [3]. We implement the inference model in FP16 mode. Dynamic empty cache is used to reduce GPU memory.

5 Discussion and Conclusion

The method can work well on cases where main vessel and most end vessel.However,some under segmentation usually be found on Vascular entrance, either end vessel are too thin

to gain precise endpoint.

Acknowledgment

We sincerely appreciate the organizers with the donation of ATM2022. We declare that pre-trained models and additional datasets are not used in this paper.

References

1. Hou, Q.e.a.: Strip pooling: Rethinking spatial pooling for scene parsing.. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020) 1
2. Isensee, F., Jaeger, P.F., Kohl, S.A., Petersen, J., Maier-Hein, K.H.: nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. Nature Methods **18**(2), 203–211 (2021) 1
3. Silversmith., W.: cc3d: Connected components on multilabel 3d images. (January 2021), https://github.com/seung-lab/connected-components-3d/ 6
4. Zhang, F., Wang, Y., Yang, H.: Efficient Context-Aware Network for Abdominal Multi-organ Segmentation. arXiv:2109.10601 [cs, eess] (Oct 2021) 1
5. Zhu, Z.e.a.: A 3d coarse-to-fine framework for volumetric medical image segmentation. 2018 International Conference on 3D Vision (3DV) (2018) 1