# Efficient Context-Aware Network for Abdominal Multi-organ Segmentation

Fan Zhang

*Department of Radiation Algorithm, Shanghai Xingmai Information Technology CO. LTD, Shanghai 200000, China.*
*zhangfan2@fosun.com*

Yu Wang

*Department of Radiation Algorithm, Shanghai Xingmai Information Technology CO. LTD, Shanghai 200000, China.*
*wangyu@fosun.com*

## Abstract

The contextual information, presented in abdominal CT scan, is relative consistent. In order to make full use of the overall 3D context, we develop a whole-volume-based coarse-to-fine framework for efficient abdominal multi-organ segmentation. 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. We propose strip pooling module to capture anisotropic and long-range contextual information, which exists in abdominal scene. Qualitative evaluation on the FLARE2021 validation cases, this method significantly improved pathological segmentation.

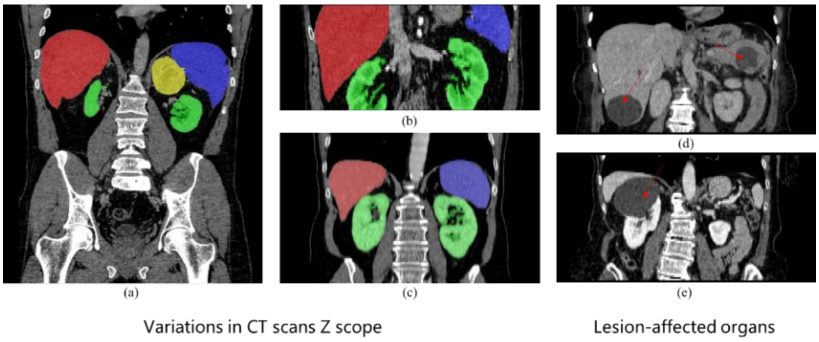
## 1. Introduction

In this paper, we focus on multi-organ segmentation from abdominal CT scans. As shown in Fig. 1, the main difficulties stem from three aspects:

1) the variations in CT scans scope, shape and size of different organ.

2) the abnormalities, like lesion-affected organ, may lead to segmentation failure.

3) the limited GPU memory size and high computation cost.



Variations in CT scans Z scope

Lesion-affected organs

Fig. 1 An illustration of abdominal CT images vary in scan scope (a-c), and lesion-affected organ (d-e) on FLARE2021 dataset.

A common solution [1] 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 [2] 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. To capture multi-organ position-related information, we exploit strip pooling [3] for collecting anisotropic and long-range context. This strip pool offers two advantages. First, compared to self-attention or non-local module, strip pool consumes less memory and matrix computation. Second, it deploys long but narrow pooling kernels along one spatial dimension to simultaneously aggregate both global and local context.

The main contributions of this work are as follows:

1) We propose a whole-volume-based coarse-to-fine framework to make full use of the overall 3D context.
2) We design anisotropic convolution block with low computation cost. We propose strip pooling module to capture anisotropic and long-range contextual information.
3) The effectiveness and efficiency of the proposed whole-volume-based coarse-to-fine framework are demonstrated on FLARE2021 challenge dataset, where we achieve the state-of-the-art with relative low time cost and less memory usage.

## 2. Method

As mentioned in Fig. 2, this whole-volume-based coarse-to-fine framework is composed of coarse and fine segmentation. A detail description of the method is as follows.

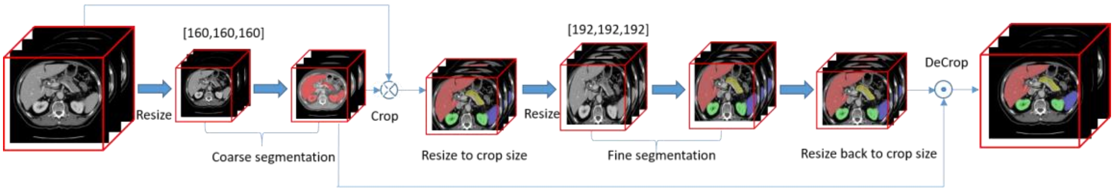


Fig. 2. A schematic diagram of whole-volume-based coarse-to-fine segmentation framework.

### 2.1 Preprocessing

- Reorientation image to target direction.
- Resampling image to fixed size.

  Coarse input: [160, 160, 160]

  Fine input: [192, 192, 192]
- Intensity normalization: First, the image is clipped to the range [-325, 325]. Then a z-score normalization is applied based on the mean and standard deviation of the intensity values.

### 2.2 Proposed Method

The proposed efficient-segNet consists of three major parts: the feature encoder module, the context extractor module, and the feature decoder module, as shown in Fig. 3.

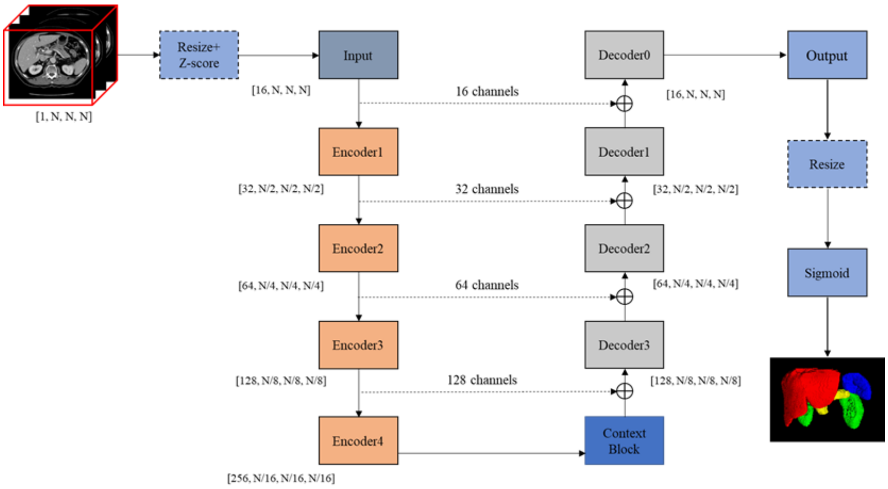


Fig. 3. Illustration of the proposed efficient-segNet.

As depicted in Fig.4, The encoder module is composed of two residual convolution block, and the decoder module with one residual convolution block. As to decoder module, we separate a standard 3D convolution with kernel size

3×3×3 into a 3×3×1 intra-slice convolution and a 1×1×3 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).

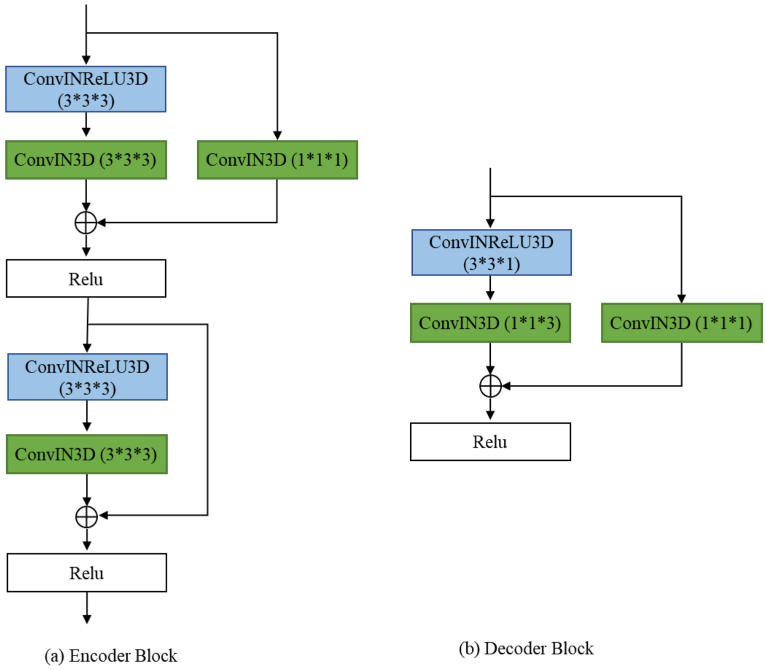


Fig. 4. Illustration of the encoder and decoder block.

We adopt 3D-based mixed pooling module(Fig. 5) to extract contextual feature, which is composed of the standard spatial pooling operations and the anisotropic strip pooling. The standard spatial pooling operation employs two average pooling with the stride of 2×2×2 and 4×4×4. The anisotropic strip pooling with three different-direction receptive fields: 1×N×N, N×1×N and N×N×1, where N is the size of feature map in last encoder module.

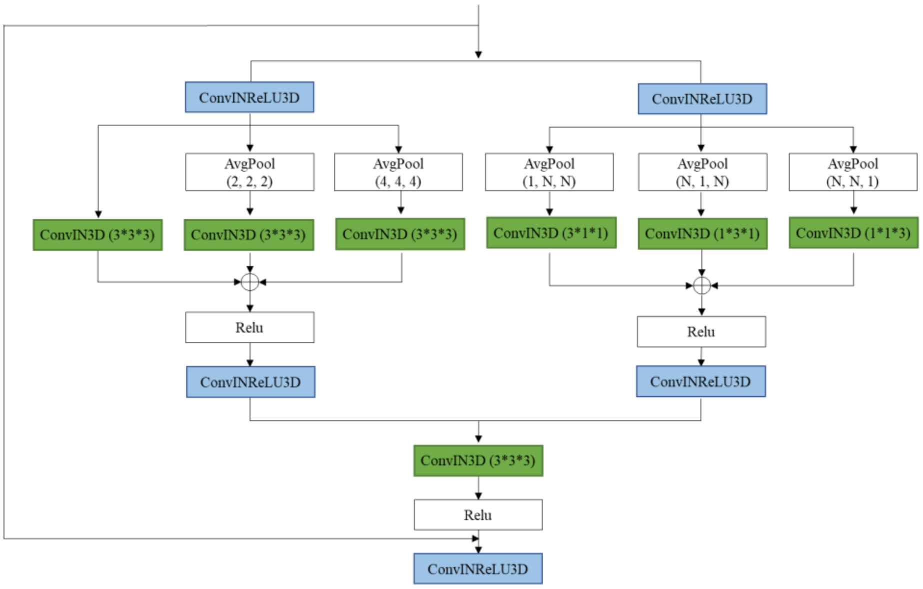


Fig. 5. Illustration of the context block.

We aggregate low and high level feature with addition rather than concatenation, because the former consumes less GPU memory.

### 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

The dataset used of FLARE2021 was composed of 361 cases, which was randomly divided into training (80%) and validation (20%) set.

### 3.2 Evaluation Metrics

The evaluation metrics are composed of dice similarity coefficient (DSC), normalized surface distance (NSD), running time, and maximum used GPU memory (when the inference is stable).

# 4. Implementation Details

## 4.1. Environments and requirements

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

Table 1. Environments and requirements.

| | |
|---|---|
| **Ubuntu version** | 16.04.10 |
| **CPU** | Intel(R) Xeon(R) Gold 5218 CPU @ 2.30GHz (×4) |
| **RAM** | 502G |
| **GPU** | Nvidia A100 (× 4) |
| **CUDA version** | 11.0 |
| **Programming language** | Python 3.6 |
| **Deep learning framework** | Pytorch (torch 1.5.0，torchvision 0.8.0) |

## 4.2. Training protocols

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

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×) |
| **Batch size** | 16 |
| **Patch size** | Coarse: 160, 160, 160<br>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** | 0.001 |
| **Learning rate decay schedule** | epoch <= epochs * 0.66:   init_lr<br>epochs * 0.66 < epoch <= epochs * 0.86:   init_lr * 0.1<br>epochs * 0.86 < epoch:   init_lr * 0.05 |
| **Training mode** | mixed precision (FP16) |
| **Training time** | 6 hours |

## 4.3. Testing protocols

The same pre-process and post-process methods were applied as training steps. In order to reduce the time cost of pre-process and post-process, resample and intensity normalization were completed in GPU. We implemented the connected component analysis in C++ library, namely cc3d. We implemented the inference in FP16 mode. Dynamic empty cache was used to reduce GPU memory.

## 5. Results

The average running time is 6.5 s per case in inference phase, and maximum used GPU memory is 2224 MB. Table 3 illustrates the results on validation set.

Table 3. Quantitative results of validation set in terms of DSC and NSD.

| Organ | DSC | NSD |
|---|---|---|
| Liver | 0.9818 | 0.9155 |
| Kidney | 0.9562 | 0.8983 |
| Spleen | 0.9830 | 0.9723 |
| Pancreas | 0.8339 | 0.6234 |
| Average | 0.9387 | 0.8524 |

## Acknowledgement

We sincerely appreciate the organizers with the donation of FLARE2021 dataset. We declare that pre-trained models and additional datasets are not used in this paper.

## References

[1] Isensee, F., Petersen et al.: "nnu-net: Breaking the spell on successful medical image segmentation". arXiv preprint arXiv:1904.08128 (2019)

[2] Zhu, Z. et al. "A 3D Coarse-to-Fine Framework for Volumetric Medical Image Segmentation." 2018 International Conference on 3D Vision (3DV) (2018).

[3] Hou, Q. et al. "Strip Pooling: Rethinking Spatial Pooling for Scene Parsing." 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020).