A multi-view multi-stage and multi-window framework for pulmonary artery segmentation from CT scans

ZeYu Liu¹, Yi Wang², Jing Wen³, Yong Zhang⁴, Hao Yin⁵, Chao Guo⁶, and ZhongYu Wang⁷

¹ Chongqing University, Chongqing city, China cvlearn0163.com
² Chongqing University, Chongqing city, China yiwang@cqu.edu.cn
³ Chongqing University, Chongqing city, China wj@cqu.edu.cn
⁴ Chongqing University, Chongqing city, China zangyong7630@163.com
⁵ Chongqing University, Chongqing city, China 202114131132@cqu.edu.cn
⁶ Chongqing University, Chongqing city, China mzlxavier1230@gmail.com
⁷ Ziwei king star Digital Technology Co., Ltd, Hefei city China wzhongyu@ziweidixing.com

Abstract. This is the technical report of the 9th place in the final result of PARSE2022 Challenge. We solve the segmentation problem of the pulmonary artery by using a two-stage method based on a 3D CNN network. The coarse model is used to locate the ROI, and the fine model is used to refine the segmentation result. In addition, in order to improve the segmentation performance, we adopt multi-view and multi-window level method, at the same time we employ a fine-tune strategy to mitigate the impact of inconsistent labeling.

Keywords: medical segmentation · pulmonary artery · coarse to fine

1 Introduction

The segmentation of the pulmonary artery is of great significance, however performing a one-stage segmentation makes the results less accurate. For this we implemented a two-stage method for pulmonary artery segmentation. The advantage of the two-stage method is high accuracy. Manual annotation is often performed at a certain plane. In order to allow the network to obtain the comprehensive information of the three planes of the patch, we use a multi-view training and inference method. In order to highlight the tissue of the segmentation target, we adopt a special multi-window method to distinguish trunk and branches. Compared with 2D CNN, 3D CNN can obtain more one-dimensional information, so we design a novel 3D CNN network, and the details is in our upcoming published paper. To address the inconsistency of manual annotations, we employ a fine-tuning strategy to modify the inference results at the testing phase.

We summarized our work as below: (1) A multi-stage, multi-view, multi-window framework for pulmonary artery segmentation. (2) A novel 3D CNN network. (3) A Fine-tune strategy to mitigate the impact of inconsistent labeling.
2 Method

2.1 Preprocessing

We adopted the following data preprocessing method. As for coarse model, we used the method like below:

- scale intensity
- resize

As for fine model, we use the preprocessing method like below:

- scale intensity
- rand crop

We notice that the main pulmonary artery and the branches are in different intensity distribution intervals. To distinguish between these two different types of tissues, we use two window levels and do not truncate the intensity in order to preserve the information. Therefore, both the coarse model and the fine model have two input channels, as shown in Fig 2.1, one channel has intensity value distribution [-900, 0], and the other is [0, 300].

![Fig. 2.1. Different window levels](image)

2.2 Proposed method

As for our proposed method, the first stage is used for locating the ROI, and the second stage is used for refining segmentation result. The process is shown in Fig 2.2. When training coarse model, the original image is scaled by trilinear interpolation, and the label is scaled by nearest neighbor interpolation. Fine is a common strategy in medical image segmentation, which has been studied for a long time.\cite{3,4,5,6}

When training coarse model, the original image is scaled by trilinear interpolation, and the label is scaled by nearest neighbor interpolation. When training
the coarse model, the original image is scaled by trilinear interpolation, and the label is scaled by nearest neighbor interpolation, the purpose of this is to accurately locate the pulmonary artery region. The training data of the second stage is randomly cropped according to the area determined by the label, and samples are separated into positive and negative based on the patch’s center voxel. We randomly choose one of the coronal, sagittal, and horizontal planes as the orientation of the patch during each iteration of the fine model training.

![Diagram](image)

**Fig. 2.2.** The proposed solution flow, one stage is used to locate the pulmonary artery region, and the second stage is used for fine segmentation

### 2.3 Network

In the field of image segmentation, U-Net[2] is one of the most popular methods, as shown in Fig 2.3, for this challenge. We also based on the idea of its classical structure to design our network. The left side of the network is an encoder and the right side is a decoder. Feature fusion is achieved by skip-connection between encoder and decoder. Because of the limited space and article type, the specific details of the network and other work will be in our upcoming published paper, so it is inconvenient to discuss details in this technical report. In this challenge, whether it is the first stage or the second stage, we both use our network to achieve our goal.

### 3 Dataset and Evaluation Metrics

The challenge obtained 200 3D volumes from 10 doctors with refined pulmonary artery labeling, 100 for the training dataset, 70 for the closed testing dataset, and 30 for the opened validated dataset. As evaluation metrics, Multi-level Dice, Multi-level HD, and time-cost are used. In this competition, whether it is the first stage or the second stage, we both use this network to achieve our goal.

The Dice coefficient is a common set similarity measure function in semantic segmentation. It is usually used to calculate the similarity between two samples, and its value ranges from [0, 1].

\[
\text{Dice} = \frac{2|X \cap Y|}{|X| + |Y|}
\]  

(3.1)
In semantic segmentation, Dice is more sensitive to the inner filling of the label, while hausdorff distance is more sensitive to the boundary. The formula is defined as below.

\[
H(A, B) = \max(h(A, B), h(B, A))
\]

(3.2)

\[
h(A, B) = \max_{a \in A} \min_{b \in B} \|a - b\|
\]

(3.3)

\[
h(B, A) = \max_{b \in B} \min_{a \in A} \|b - a\|
\]

(3.4)

4 Implementation Details

Here are the details of our implemented environment and training process. In segmentation tasks, class imbalance is a very common phenomenon, like V-Net\cite{1}, we also use Dice loss to solve this problem. Coarse model training parameter is set as Table 4.1, and Fine model training parameter is set as Table 4.2.

5 Results

During this challenge, we make a 5-fold cross-validation experiment on the two-stage network to judge whether the training process can converge when using 3D CNN, eg, 3D U-Net network mentioned in section 2.3. The result is shown at table 5.1. 3D convolution is able to converge and our proposed method outperforms 3D-UNet.

Our submitted docker image inference process can be summarized into three steps: (1) Coarse model locating ROI, (2) Fine mode segmentation on the ROI,
Table 4.1. coarse model training parameter

| Item                  | Value                                    |
|-----------------------|------------------------------------------|
| Initial method        | nn.init.kaiming_normal                   |
| Batch size            | 2                                        |
| Patch size            | 192 192 192                              |
| Total epoches         | 200                                      |
| Optimizer             | Novograd                                 |
| Loss                  | DiceLoss                                 |
| Initial learning rate | 8.00E-03                                 |
| Learning rate decay   | LinearWarmupCosineAnnealingLR            |
| Stopping criteria,    | training 200 epoches, no other model     |
| and optimal model     | selection criteria                        |
| Data Augmentation     |                                          |

Table 4.2. Fine model training parameter

| Item                  | Value                                    |
|-----------------------|------------------------------------------|
| Initial method        | nn.init.kaiming_normal                   |
| Batch size            | 2                                        |
| Patch size            | 192 192 192                              |
| Each voxel case sample patch | 4                                        |
| Total epoches         | 300                                      |
| Optimizer             | Novograd                                 |
| Loss                  | DiceLoss                                 |
| Initial learning rate | 8.00E-03                                 |
| Learning rate decay   | LinearWarmupCosineAnnealingLR            |
| Stopping criteria,    | training 300 epoches, no other model     |
| and optimal model     | selection criteria                        |
| Data Augmentation     |                                          |

Table 4.3. Training Environment

| Item        | Configure                                       |
|-------------|-------------------------------------------------|
| Operating System | windows-10-10.0.19043-sp0                      |
| Memory      | 190GB                                           |
| CPU         | Intel Xeon Gold 6142 M 2.60 GHZ * 16           |
| GPU         | NVIDIA A100-PCIE-80GB *2                        |
(3) 3D-fixpoint method refinement the result. The one-stage restoration method to obtain ROI coordinates is as follows:

\[
\text{OriginalCoordinate} = \frac{\text{ResizedCoordinate} \times \text{ResizedSpacing}}{\text{OriginalSpacing}}
\] (5.1)

Because of the objective inaccurate positioning problem, before restore the coordinate we add a fixed value to it, the value was set as 2. Sliding window inference is performed on the cropped area, and the overlapping area between the sliding windows is set to 0, because the sampling patch method is according to the voxel, setting it to 0 will nearly not produce stitching artifacts. The area delimited by each sliding window will be transposed into the coronal plane, the axial plane, and the horizontal plane for inference. We do not activate the feature map of each plane separately, by contrast, we average them into one feature map before activating.

In the validation phase we found that fine-tuning one part of training data to improve performance will not improve the overall performance, the improvement of part data in the validation set will always be accompanied by the decline of another part of the data, we believe that the reason is the inconsistency of manual annotations. In order to solve such contradiction, we add the results of different models’s outputs rather than average them. In the submitted docker image, stage two has two weight files. The first weight file is coming from under the strategy shown in the Table 4.2, and the second one is on the basis of the first weight file, fine-tune the 10 poor cases in the training set.

In order to eliminate outlier false positives caused by the background, the complete segmentation results will be sent to a 3D-fixpoint process for iteration. The difference between the 3D-fixpoint process and the previous fine model segmentation is that before 3D-fixpoint performs the same three-direction sliding window inference, it first extracts the maximum connected component according to the input voxels, and then expand the extracted border area. 3D-fixpoint process will exect 2 times and the expanded voxel num is set 5.

Finally, our two-stage method won the ninth place in the PARSE 2022 challenge in the testing phase. Part of results on the validation set is shown as Fig 5.1.

6 Discussion and Conclusion

For the dataset of this challenge, we propose a multi-view, multi-stage and multi-window framework segmentation method to segment the pulmonary artery, one-stage for locating the pulmonary artery region, and two-stage for fine segmentation. In addition, we also adopt a fine-tune method according to the performance of the validation set to improve our submitted docker image performance. The model of each stage is based on 3D convolution, so we are at a disadvantage place in terms of memory and processing time. However, due to the inconsistency of manual labeling, such efforts are worthwhile, and the final result is clinically better than manual annotation. Our future research work includes hard sample mining, semi-supervised learning, registration, etc.
**Fig. 5.1.** 9 cases of validation set

| Fold  | Model   | Dice | HD  |
|-------|---------|------|-----|
| Fold 1| 3D U-Net| 0.78 | 88.6|
|       | ours    | 0.85 | 34.1|
| Fold 2| 3D U-Net| 0.79 | 105.2|
|       | ours    | 0.87 | 86.6|
| Fold 3| 3D U-Net| 0.80 | 104.0|
|       | ours    | 0.88 | 42.9|
| Fold 4| 3D U-Net| 0.79 | 100.8|
|       | ours    | 0.86 | 47.2|
| Fold 5| 3D U-Net| 0.78 | 97.3|
|       | ours    | 0.87 | 40.2|
| Average| 3D U-Net| 0.79 | 99.2|
|        | ours | 0.87 | 43.8|

**Table 5.1.** Five fold cross-validation on training set
References

1. Milletari, F., Navab, N., Ahmadi, S.A.: V-net: Fully convolutional neural networks for volumetric medical image segmentation. 2016 Fourth International Conference on 3D Vision (3DV) pp. 565–571 (2016)
2. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: MICCAI (2015)
3. Selvarani, P., Vaithiyanathan, V.: Medical image fusion and segmentation using coarse-to-fine level set with brovey transform fusion. Research Journal of Applied Sciences, Engineering and Technology 4, 3623–3627 (2012)
4. Wu, Z., Zhou, Q., Wang, F.: Coarse-to-fine lung nodule segmentation in ct images with image enhancement and dual-branch network. IEEE Access 9, 7255–7262 (2021)
5. Zabihollahy, F., Viswanathan, A.N., Schmidt, E.J., Morcos, M., Lee, J.: Fully automated multi-organ segmentation of female pelvic magnetic resonance images with coarse-to-fine convolutional neural network. Medical physics (2021)
6. Zhu, Z., Xia, Y., Shen, W.C., Fishman, E.K., Yuille, A.L.: A 3d coarse-to-fine framework for volumetric medical image segmentation. 2018 International Conference on 3D Vision (3DV) pp. 682–690 (2018)