Multi-organ segmentation based on 2.5D semi-supervised

Hao Chen[1[0000−0002−6849−9413]], Wen Zhang[1[0000−0001−8679−5462]], Xiaochao Yan[1[0000−0002−4358−6588]], Yanbin Chen[1[0000−0003−0231−1390]], Xin Chen[1[0000−0003−3039−4939]], Mengjun Wu[1[0000−0002−6176−4560]], Lin Pan[1[0000−0002−6176−4560]], and Shaohua Zheng[1(✉)[0000−0002−9133−705X]]

Intelligent Image Processing and Analysis Laboratory, Fuzhou University, Fuzhou FJ 350108, CHN sunphen@fzu.edu.cn

Abstract. Automatic segmentation of multiple organs is a challenging topic. Existing approaches either focus only on the 2D background or explore only the 3D background. It can lead to insufficient contextual exploration in organ segmentation. In recent years, many methods for automatic segmentation based on fully supervised deep learning have been proposed. However, it is very expensive and time-consuming for experienced medical practitioners to annotate a large number of pixels. In this paper, we propose a new two-dimensional multi slices semi supervised method to perform the task of abdominal organ segmentation. The network utilizes the information along the Z-axis direction in CT images and preserves and exploits the useful temporal information in adjacent slices. Also, we combine Cross-Entropy loss and Dice loss as loss functions to improve the performance of our method. In using unlabeled images, we use the EMA (Exponential Moving Average) algorithm to optimize the teacher model weights based on the teacher-student model in the process of training the student model with labeled data, allowing the teacher model to set pseudo-labels for unlabeled data and then learned by the students, using a consistency loss function to promote refinement of model performance for the purpose of using unlabeled images. The mean DSC for all cases we obtained on the validation set was 0.5684, the mean NSD was 0.5971, and the total run time was 783.14s.

Keywords: Semi-supervised · Deep learning · Organ segmentation.

1 Introduction

Automatic segmentation of multiple organs is a challenging topic. In recent years, many fully supervised deep learning automatic segmentation methods have been proposed that rely on a large number of pixel-level annotations, but the annotation of multiple organs is very expensive and time-consuming. In addition, existing 2D methods cannot fully utilize spatial information, and 3D methods consume a lot of computational resources.This paper proposes a novel 2.5D multi-slice semi-supervised approach to achieve abdominal organ segmentation.

There are also two difficult problems with medical segmentation: (1) Manual annotation of organs from CT scans is time-consuming and laborious. (2) Medical data involves patient privacy issues. Therefore, we usually do not have access to a large number of cases, let alone labeled cases. As a potential alternative, semi-supervised learning can explore useful information from unlabeled cases.

Integrating learning and other algorithms that consume huge CPU and GPU computing power can obtain better metrics and segmentation results. However, it will eventually lead to a very bloated model. In contrast, lightweight models have lower requirements for device performance and are easier to deploy in practical applications. The main theme of the FLARE2022 Challenge is to propose a solution with high efficiency and high accuracy as the benchmark while occupying low GPU and CPU resources. It is important to have both lightweight models and low resource usage, as well as high accuracy and efficiency. This is exactly the difficulty of the FLARE2022 challenge. Representative of the current state-of-the-art approaches is nnU-Net [4], which provides a fully automated end-to-end segmentation method and has topped several competitions. Although its model has good performance on segmentation, it also has the drawback of not being lightweight enough and high requirements of GPU memory.

In this paper, our contributions are as follows:

1. We propose a 2.5D semi supervised multi organ segmentation framework, which takes adjacent slices as input [6], improves the utilization of 3D information in CT without increasing too many computing resources, and uses the teacher-student semi supervised strategy to utilize unlabeled CT information.

2. Using 2D U-Net [4] as the main network framework and incorporating the CBAM attention module (Convolutional Block Attention Module) [11] to improve the mining of data information.

3. EMA [5] is used to optimize the teacher model to make its model performance more robust and to add pseudo-labels to unlabeled images.

2 Method

Figure 1 illustrates the general workflow of our semi-supervised segmentation network. First, we train the labeled data to obtain the student model while smoothing the model weights using the EMA (exponential moving average) method to obtain the teacher model. Then, the teacher model is used to predict the unlabeled data to obtain pseudo-labels. Finally, the pseudo-label is used to train the student model on the unlabeled data to obtain the final student model. In this stage, the teacher-student model uses the U-Net-CBAM network architecture, and one case of every 35 unlabeled data is selected for training.

2.1 Preprocessing

In terms of preprocessing, we performed four main steps as follows.

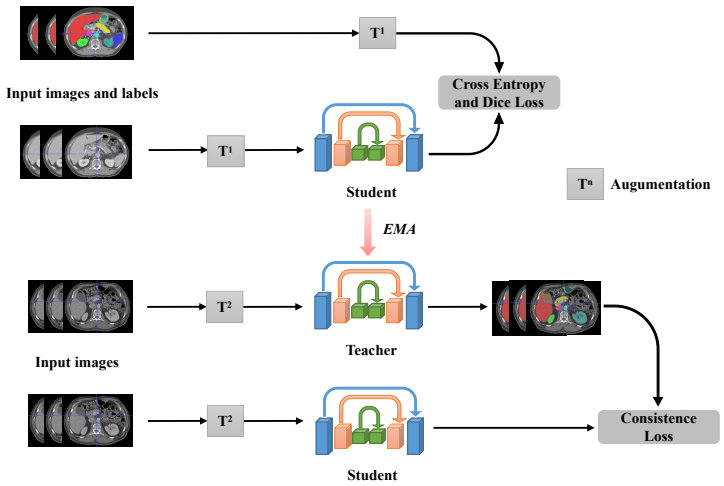


Fig. 1. The workflow of our method. The labeled dataset is fed into the student network after data augmentation, and the student network is passed through the EMA to obtain a more robust teacher model. The unlabeled dataset is pseudo-labeled by the teacher network after data augmentation and is fed into the student network with the corresponding unlabeled dataset.

Threshold truncation: According to our observation of several CT files, when the threshold value is taken around [-250,300], the contrast between organ tissues is more obvious, so we do threshold truncation based on this.

Cropping strategy: In the training phase, do crop for images and labels according to the slices containing labels. We discard the slices without labels. In the z-axis, we take the number of slices down to the second power of 2 to speed up the subsequent data reading.

Resampling method for anisotropic data: We adjust the slice size during data loading to reduce GPU memory usage. After testing with different sizes, the 2D slice size of 128×128 has an impact on model performance, while 256×256 takes up more memory without much performance improvement. So we resample the 2D slice from the original 512×512 to a size of 192×192. Meanwhile, in the scaling process, We do a trilinear sampling of the image to ensure that the image texture features, etc. change as little as possible, and do a nearest neighbor sampling of the labels to ensure that the label values remain unchanged.

Intensity normalization method: To make the data more easily compute and improve model's performance, we normalize and standardize the threshold value. At this point, our preprocessing is over.

2.2 Proposed Method

We apply the attention mechanism to the U-Net segmentation network, which can better pay attention to salient regions and suppress irrelevant background regions. The attention module can be well embedded in the CNN framework and can improve the model performance without increasing too much computational effort when compare to other attention modules. In terms of details, connecting 3 slices so that they occupy channels as 3-dimensional inputs makes full use of the 3D information without increasing the GPU memory footprint as much as possible. In addition, the values obtained by EMA using moving average are smoother and smoother on the Loss image, with less jitter, and do not make the sliding average fluctuate greatly due to an abnormal value taken at one time, so the robustness is better and the results obtained can be optimized.

Figure 2 illustrates the Network architecture of our method.

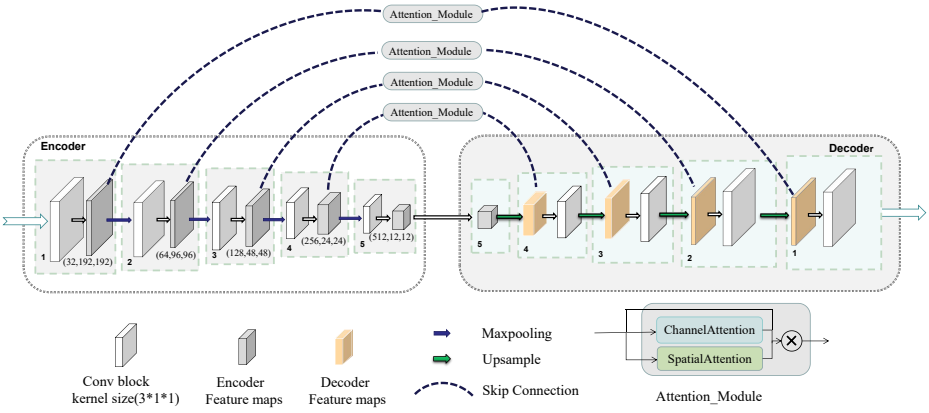


Fig. 2. Network architecture.U-Net-CBAM is based on U-Net with the addition of the CBAM attention module in the jump connection can better extract the image features and details.

Strategies to use the unlabelled cases: We use the teacher's EMA model to predict unlabeled data to get pseudo-labels and use consistency loss to allow student models to learn the content of unlabeled data.

Network architecture details description

Our proposed U-Net-CBAM network consists of a combination of U-Net network and CBAM module, which consists of spatial attention module and channel attention module. The network input first goes through 5 convolution modules and 4 max pooling layers to complete the downsampling process, and then goes through 5 convolution modules and four upsampling to get the output. The output of each layer of the downsampling path is connected by the features of the skip connection and the upsampling path, respectively. The skip connection per-

forms channel-wise and spatial-wise feature correction on the features through the CBAM module.

Loss function [7]

1) Supervised loss function: It consists of multivariate Cross-Entropy Loss and DiceLoss [9] with weights, and the percentages of both are α and $1-\alpha$, respectively. DiceLoss comes with weights to facilitate the adjustment of the percentages between different categories.

$$Loss = \alpha * CELoss(g,p) + (1-\alpha) * DiceLoss(g,p) \tag{1}$$

where g is ground truth, p is prediction and we use α to balance the celoss and DiceLoss.

$$DiceLoss = 1 - \sum_{i=0}^{k-1} \beta_i * Dice(g_i, p_i) \tag{2}$$

where g_i is ground truth, p_i is prediction, β_i represents weight of every class.

$$Dice(G,P) = \frac{2|G \cap P|}{|G| + |P|} \tag{3}$$

where G and P is what we want to compare by using dice.

$$CEloss(g,p) = -\sum_{i=1}^{C} g_i \log(p_i) \tag{4}$$

where C denotes the number of classes, g_i is ground truth, p_i is prediction.

2) Unsupervised loss function: we use consistency loss [5], which directly measures the consistency between prediction and Pseudo Label.

$$ConsistenceLoss = \|M_s(A_1(x)) - M_t(A_2(x))\|_2^2 \tag{5}$$

where $A_1()$ and $A_2()$ are different functions of transformations, $M_s()$ and $M_t()$ denotes output of model of Student and model of Teacher.

Strategies to reduce time and resource consumption

Firstly, we use U-Net network architecture and combine it with CBAM modules. Compared with other attention modules, it can reduce parameters and computing power. Secondly, we use multiple slices to avoid complete loss of 3D information and 2D methods to avoid loss of 2D information. Besides, do resize operation on the image to reduce the video memory consumption without affecting the model performance as much as possible by appropriately reducing the resolution. Last but not least, normalization and standardization of the image can improve the inference speed and resource consumption of the computer.

2.3 Post-processing

Operation 1: We perform a median filtering operation with a convolution kernel of (5×5×5) on the image to optimize the edges of the segmentation result.

Operation 2: Since we did the image reduction operation in pre-processing, i.e., from the original size to z×192×192 according to the trilinear sampling method, we enlarge the image back to the original size in post-processing according to the nearest neighbor interpolation operation.

When we operate 1, some of the results in the segmentation result disappear, even reducing the size of the convolution kernel causes this, which makes us abandon operation 1.

3 Experiments

3.1 Dataset and evaluation measures

The FLARE2022 dataset is curated from more than 20 medical groups under the license permission, including MSD [10], KiTS [2,3], AbdomenCT-1K [8], and TCIA [1]. The training set includes 50 labelled CT scans with pancreas disease and 2000 unlabelled CT scans with liver, kidney, spleen, or pancreas diseases. The validation set includes 50 CT scans with liver, kidney, spleen, or pancreas diseases. The testing set includes 200 CT scans where 100 cases have liver, kidney, spleen, or pancreas diseases and the other 100 cases have uterine corpus endometrial, urothelial bladder, stomach, sarcomas, or ovarian diseases. All the CT scans only have image information and the center information is not available.

The evaluation measures consist of two accuracy measures: Dice Similarity Coefficient (DSC) and Normalized Surface Dice (NSD), and three running efficiency measures: running time, area under GPU memory-time curve, and area under CPU utilization-time curve. We will improve the running efficiency measures while meeting the running efficiency measures.

3.2 Implementation details

Environment settings We performed the training and inference process in the environment of Table 1, using the lower machine configuration shown in the table in order to always be reminded of the special requirements of the team challenge for resource usage. The development environments and requirements are presented in Table 1.

Training protocols In training the teacher and student models, we optimized the models using the Adam optimizer with an initial learning rate of 0.001 and a learning rate reduction strategy using CosineAnnealingLR. The training protocols are presented in Table 2 and Table 3.

Table 1. Development environments and requirements.

Windows/Ubuntu version	Ubuntu 18.04.6 LTS
CPU	Intel(R) Core(TM) i5-7500 CPU@3.40GHz
RAM	4×4GB; 2400MT/s
GPU (number and type)	1 RTX 2080Ti (8G)
CUDA version	11.4
Programming language	Python 3.8
Deep learning framework	Pytorch (Torch 1.7.0, torchvision 0.8.2)
Specific dependencies	medicaltorch, pandas, scipy, collections
(Optional) Link to code	

Table 2. Training protocols.

Network initialization	Teacher Net
Batch size	16
Patch size	3×192×192
Total epochs	150
Optimizer	Adam
Initial learning rate (lr)	0.001
Lr decay schedule	CosineAnnealingLR
Training time	3.25 hours
Number of model parameters	74.1M[1]
Number of flops	8.22G[2]
CO_2eq	1 Kg[3]
Loss function	Cross Entropy and Dice Loss[4]

4 Results and discussion

4.1 Quantitative results on validation set

On the provided validation datasets, we perform ablation experiments and the results are shown in Table 4. The purpose is to compare the model performance on the validation set using only labeled data and using both labeled and unlabeled data. When using only labeled datasets, the Dice score reaches 0.5454, while when using both labeled and unlabeled datasets, the Dice score reaches 0.5684.

After using unlabeled data, the segmentation results have been improved, and the previously under-segmented organs can be initially segmented. The improvement in the Dice score illustrates that using pseudo-labels to exploit unlabeled data can improve model performance.

The provided validation set with ground truth are used to evaluate the performance of the proposed method in this paper, two judging metrics, DSC and NSD, are applied, and the results are shown in Table 4. The average value of DSC is 0.5684 and the average value of NSD is 0.5971.

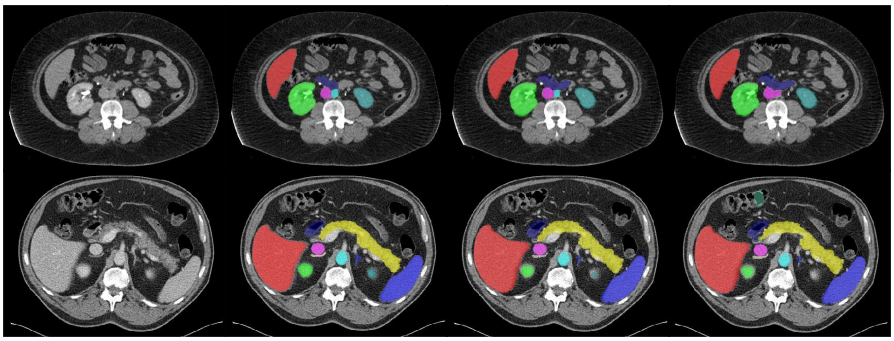
Table 3. Training protocols for the refine model.

Network initialization	Student Net
Batch size	8
Patch size	3×192×192
Total epochs	100
Optimizer	Adam
Initial learning rate (lr)	0.001
Lr decay schedule	CosineAnnealingLR
Training time	16 hours
Number of model parameters	74.1M[5]
Number of flops	8.65G[6]
CO_2eq	1 Kg[7]
Loss function	Consistence Loss[8]

We can find that the model performs better in the segmentation of three organs, the Liver, Spleen, and Aorta, but is slightly weak in the other organs.

4.2 Qualitative results on validation set

The segmentation excellence results are shown in the Figure 3 and the results of the poor segmentation are shown in the Figure 4. We demonstrate ablation experiments in a visual manner and we can find that semi supervised method does make segmentation better. They are all from the provided validation set with ground truth.



(a) Image (b) Ground truth (c) Semi-Supervised (d) Supervised

Fig. 3. Excellent segmentation results

Possible reasons for the failure of cases or organ segmentation are listed as follows:

Table 4. Overview of DSC and NSD metrics on validation set.

Name	Mean(DSC)	Mean(DSC)	Mean(NSD)	Mean(NSD)
Metohd	w/o ssl	w ssl	w/o ssl	w ssl
Liver	0.8079	0.8496	0.7320	0.7718
RK	0.4590	0.4816	0.3773	0.4022
Spleen	0.7169	0.7073	0.6400	0.6475
Pancreas	0.5177	0.5463	0.6109	0.6447
Aorta	0.8525	0.8703	0.8578	0.8811
IVC	0.6561	0.6671	0.6219	0.6424
RAG	0.4057	0.3968	0.5415	0.5458
LAG	0.3045	0.3750	0.4236	0.5130
Gallbladder	0.2460	0.3540	0.1918	0.3026
Esophagus	0.5981	0.5638	0.7202	0.6917
Stomach	0.5650	0.5890	0.5580	0.5736
Duodenum	0.3685	0.3957	0.5475	0.5911
LK	0.5916	0.5931	0.5307	0.5544
Average	0.5454	0.5684	0.5656	0.5971

1) We used too few adjacent slices for training.

2) Some of the organ segments were lost in the preprocessing because some information was lost in the scaling of the images.

3) The pseudo-labels from the teacher model may contain error information, which the student model may have learned in training.

4.3 Segmentation efficiency results on validation set

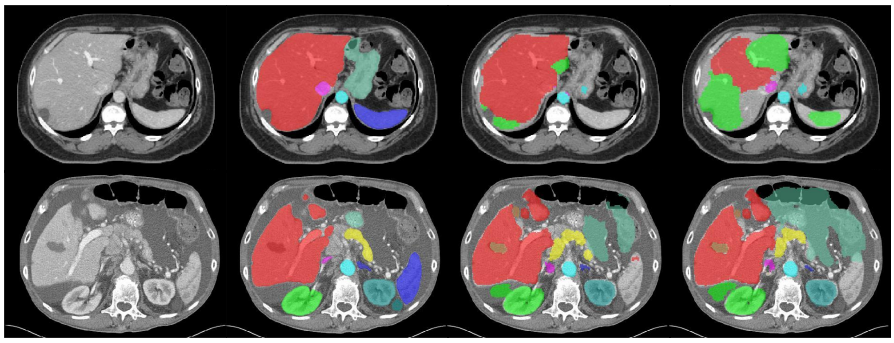
The segmentation efficiency results of the validation set are shown in Table 5, which mainly includes various metrics such as CPU, GPU, and runtime. We can see that the model perfectly meets the minimum requirements of the Challenge in terms of resource usage and consumption, i.e., the GPU-Memory usage is <2048MB, and we only occupy 1455MB. The model has a good level in Time, AUC-GPU-Time, and AUC-CPU-Time metrics, and the average values are 15.66s, 17140.12, and 348.15 respectively.

4.4 Results on final testing set

This is a placeholder. We will send you testing results after the challenge.

4.5 Limitation and future work

The limitation of our approach is that we cannot guarantee a good segmentation result in the supervised training phase, and therefore the trained model



(a) Image (b) Ground truth (c) Semi-Supervised (d) Supervised

Fig. 4. Poorer segmentation results

Table 5. Overview of segmentation efficiency on validation set.

	Time/s	GPU-Memory/MB	AUC-GPU-Time	AUC-CPU-Time
Lowest	8.53	1455	94932	1661.84
Average	15.66	1455	17140.12	348.15
Highest	74.56	1455	7644	181.96

cannot pseudo-label the unlabeled data well. Future work can be directed to improve the effect of setting pseudo-labels by optimizing the model in the supervised training phase.

5 Conclusion

In the work of abdominal multi-organ segmentation, we propose a 2.5D-based semi-supervised segmentation method to achieve effective use of unlabeled data and reduce the occupation of computing resources. The main findings and results are as follows.

Finding:

1) Our method sets 150 epochs during the training process and actually converges to a smaller value (0.03) with 100 epochs.

2) In some cases, it is able to segment each organ well. However, in some cases, the corresponding organs can not be segmented, and after observation, we find that there is more noise in the CT images in these cases.

3) The performance of the model is improved after using pseudo-labels to exploit unlabelled data.

Results:

1) We believe that only a part of the data distribution of some cases is learned during training and the generalization performance of the model is not very good.

2) The quality of the CT image files themselves is also important, i.e. the equipment used to take the CT images should also be checked.

3) Our teacher-student model strategy is feasible, and it improves the performance of our model to some extent.

Acknowledgements The authors of this paper declare that the segmentation method they implemented for participation in the FLARE 2022 challenge has not used any pre-trained models nor additional datasets other than those provided by the organizers. The proposed solution is fully automatic without any manual intervention.

References

1. Clark, K., Vendt, B., Smith, K., Freymann, J., Kirby, J., Koppel, P., Moore, S., Phillips, S., Maffitt, D., Pringle, M., et al.: The cancer imaging archive (tcia): maintaining and operating a public information repository. Journal of Digital Imaging **26**(6), 1045–1057 (2013) 6
2. Heller, N., Isensee, F., Maier-Hein, K.H., Hou, X., Xie, C., Li, F., Nan, Y., Mu, G., Lin, Z., Han, M., et al.: The state of the art in kidney and kidney tumor segmentation in contrast-enhanced ct imaging: Results of the kits19 challenge. Medical Image Analysis **67**, 101821 (2021) 6
3. Heller, N., McSweeney, S., Peterson, M.T., Peterson, S., Rickman, J., Stai, B., Tejpaul, R., Oestreich, M., Blake, P., Rosenberg, J., et al.: An international challenge to use artificial intelligence to define the state-of-the-art in kidney and kidney tumor segmentation in ct imaging. American Society of Clinical Oncology **38**(6), 626–626 (2020) 6
4. 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) 2
5. Li, X., Yu, L., Chen, H., Fu, C.W., Xing, L., Heng, P.A.: Transformation-consistent self-ensembling model for semisupervised medical image segmentation. IEEE Transactions on Neural Networks and Learning Systems **32**(2), 523–534 (2020) 2, 5
6. Lv, P., Wang, J., Wang, H.: 2.5 d lightweight riu-net for automatic liver and tumor segmentation from ct. Biomedical Signal Processing and Control **75**, 103567 (2022) 2
7. Ma, J., Chen, J., Ng, M., Huang, R., Li, Y., Li, C., Yang, X., Martel, A.L.: Loss odyssey in medical image segmentation. Medical Image Analysis **71**, 102035 (2021) 5
8. Ma, J., Zhang, Y., Gu, S., Zhu, C., Ge, C., Zhang, Y., An, X., Wang, C., Wang, Q., Liu, X., Cao, S., Zhang, Q., Liu, S., Wang, Y., Li, Y., He, J., Yang, X.: Abdomenct-1k: Is abdominal organ segmentation a solved problem? IEEE Transactions on Pattern Analysis and Machine Intelligence (2021). https://doi.org/10.1109/TPAMI.2021.3100536 6
9. Milletari, F., Navab, N., Ahmadi, S.A.: V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: 2016 fourth international conference on 3D vision (3DV). pp. 565–571. IEEE (2016) 5

10. Simpson, A.L., Antonelli, M., Bakas, S., Bilello, M., Farahani, K., Van Ginneken, B., Kopp-Schneider, A., Landman, B.A., Litjens, G., Menze, B., et al.: A large annotated medical image dataset for the development and evaluation of segmentation algorithms. arXiv preprint arXiv:1902.09063 (2019) 6
11. Woo, S., Park, J., Lee, J.Y., Kweon, I.S.: Cbam: Convolutional block attention module. In: Proceedings of the European conference on computer vision (ECCV). pp. 3–19 (2018) 2