An embarrassing attempt to optimize nnUNet

Ziqi Zhou
Guangdong Key Laboratory of Intelligent Information Processing, Shenzhen University
Yuehai Campus: 3688 Nanhai Avenue, Nanshan District, Shenzhen, Guangdong, China
zhouziqi2019@email.szu.edu.cn

Li Kang
Guangdong Key Laboratory of Intelligent Information Processing, Shenzhen University
Yuehai Campus: 3688 Nanhai Avenue, Nanshan District, Shenzhen, Guangdong, China
kangli@szu.edu.cn

Abstract

Abdominal organ segmentation plays an extremely important role in clinical practice, and to some extent it seems to be a solved problem, as the most advanced methods have achieved interobserver performance in several benchmark datasets. To improve the accuracy of segmentation, the adaptive segmentation framework nnUNet is often used for implementation. However, due to the computational burden, the inference speed of nnUNet is not fast, and it needs to be improved. Therefore, we first modified the original network structure to make inference speed faster: In the deep supervision stage of the network, only the last three highest layers are selected as the output, and corresponding weights are assigned to them; Secondly, The number of test time augmentation is reduced to two to minimize the time for inference; Finally, since the dataset of the competition is a multi-center dataset, a combination of dice and topk loss is used to focus on supervising the samples with the largest differences. From the test results, proposed model achieves a compromise between accuracy and speed.

Key words:3D;U-Net;organ segmentation;abdominal

1. Introduction

Rapid segmentation of abdominal organs is essential for many clinical applications such as computer-aided diagnosis and computer-aided surgery. However, due to the heterogeneous data, the complex background, different organ sizes and blurred boundary of organs, the task of abdominal organ segmentation becomes very challenging. In order to solve this problem, we introduce a new framework nnU-Net for multiple organ segmentation in the abdominal region through the classic medical image segmentation network U-Net. nnU-Net is a deep learning based segmentation method that automatically configures itself, including preprocessing, network architecture, training and post-processing for any new task in the biomedical field. In image segmentation based on deep learning, good training effect is not only the selection of appropriate network framework, but also the selection of data processing and loss function. In order to solve the lightweight problem in the segmentation process, we have reduced the number of sample flips during test time augmentation, and reduce the number of levels of deep supervision to save GPU memory usage. In order to improve the segmentation accuracy of long-tail samples and speed up convergence, we proposed to use the combination of Top K loss function and Dice loss to segment the final output results, so as to get the final segmentation results.

2. Method

Figure 1 illustrates the applied 3D nnU-Net, where a U-Net architecture is adopted.

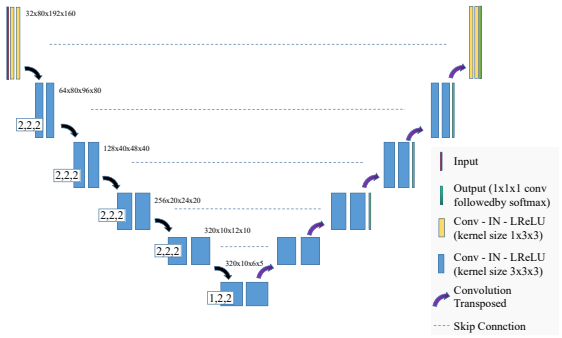

Figure 1. Network architecture

The proposed method is basically consistent with nnUNet in the data preprocessing part. In the training phase,

the improvements made to the network are mainly in loss and deep supervision. We use the combination of Top K cross entropy loss function and Dice loss to calculate error between weighted average of the last three layers of output and ground truth.

Figure 2 illustrates the applied overall framework, where the segmentation of abdominal organs is adopted.

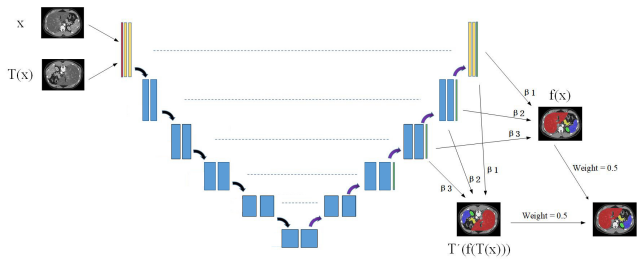

Figure 2. Overall framework of the network

2.1. Preprocessing

The proposed method includes the following preprocessing steps:

- Cropping strategy:
  Combination of random cropping and ROI cropping. Due to the variability of the input size and the imbalance of the category, we follow the setting of nnUNet and maintain approximately one-third of the positive and negative sample ratio. The probability of one-third will choose the patch centered on the foreground pixel as the input. Otherwise, the patch is randomly cropped from the sample. The size of the patch is [80,192,160].
- Resampling method for anisotropic data:
  None.
- Intensity normalization method:
  First, the dataset is clipped to the [0.5, 99.5] percentiles of the intensity values of the training dataset. Then a z-score normalization is applied based on the mean and standard deviation of the intensity values.

2.2. Proposed Method

- Network architecture details:
  The network architecture is based on 3D low resolution nnU-Net, as shown in the Figure 1, but slightly modified. We only give output weights to the last three layers of the output layer, and the corresponding output weights are $\beta_1 = 0.7$, $\beta_2 = 0.2$ and $\beta_3 = 0.1$ respectively; On the other hand, to minimize the inference time, the data is only augmented once in test time augmentation, instead of inferencing 8 times like the original framework.
- Loss function: we use the summation between Dice loss and Top K loss. The voxels with larger losses are supervised, so that the simplification of the structure will not cause too much negative impact on the indicators. The overall loss of the model can be written as:

$$Loss = \sum_{i=1}^{3} \beta_i * (1 - \frac{2\sum_{c=1}^{C}\sum_{j}^{N} p_{cj} g_{cj}}{\sum_{c=1}^{C}\sum_{j}^{N} p_{cj}^2 + \sum_{c=1}^{C}\sum_{j}^{N} g_{cj}^2} + \frac{\sum_{c=1}^{C}\sum_{j}^{N} \mathbb{1}(l_{cj} \geq t) l_{cj}}{C\sum_{j}^{N} \mathbb{1}(l_{cj} \geq t)}), l_{cj} = -g_{cj}\log(p_{cj}) \quad (1)$$

  where p, g are the predictions and ground truth, β means the weight of deep supervision, C means the number of class, N means the number of voxels, and t is the threshold of Top K loss, which is set to the decile of all voxel losses in descending order.
- Number of model parameters: 30777024
- Number of flops: 1177522176000

2.3. Post-processing

A connected component analysis of all ground truth labels is applied on training data.

3. Dataset and Evaluation Metrics

3.1. Dataset

- A short description of the dataset used:
  The dataset used of FLARE2021 is adapted from MSD [1] (Liver [2], Spleen, Pancreas), NIH Pancreas [3, 4, 5], KiTS [6, 7], and Nanjing University under the license permission. For more detail information of the dataset, please refer to the challenge website and [8].
- Details of training / validation / testing splits:
  The total number of cases is 511. An approximate 70%/10%/20% train/validation/testing split is employed resulting in 361 training cases, 50 validation cases, and 100 testing cases. The detail information is presented in Table 1.

3.2. Evaluation Metrics

- Dice Similarity Coefficient (DSC)
- Normalized Surface Distance (NSD)
- Running time
- Maximum used GPU memory (when the inference is stable)

Table 1. Data splits of FLARE2021.

Data Split	Center	Phase	# Num.
Training (361 cases)	The National Institutes of Health Clinical Center	portal venous phase	80
	Memorial Sloan Kettering Cancer Center	portal venous phase	281
Validation (50 cases)	Memorial Sloan Kettering Cancer Center	portal venous phase	5
	University of Minnesota	late arterial phase	25
	7 Medical Centers	various phases	20
Testing (100 cases)	Memorial Sloan Kettering Cancer Center	portal venous phase	5
	University of Minnesota	late arterial phase	25
	7 Medical Centers	various phases	20
	Nanjing University	various phases	50

Table 2. Environments and requirements.

Windows/Ubuntu version	Ubuntu 18.04.5 LTS
CPU	Intel(R) Core(TM) i9-10900K CPU @ 3.70GHz
RAM	8×4GB; 2.67MT/s
GPU	Nvidia RTX3090
CUDA version	11.2
Programming language	Python3.7
Deep learning framework	Pytorch (Torch 1.8.0)
Specification of dependencies	nnUNet
(Optional) code is publicly available at	Github

Table 3. Training protocols.

Data augmentation methods	Rotations, scaling, Gaussian noise, Gaussian blur, brightness, contrast, simulation of low resolution, gamma correction and mirroring.
Initialization of the network	"he" normal initialization
Patch sampling strategy	More than a third of the samples in a batch contain at least one randomly chosen foreground class which is the same as nn-Unet [9].
Batch size	2
Patch size	80×192×160
Total epochs	300
Optimizer	Stochastic gradient descent with nesterov momentum ($\mu = 0.99$)
Initial learning rate	0.01
Learning rate decay schedule	poly learning rate policy: $(1 - epoch/1000)^{0.9}$
Stopping criteria, and optimal model selection criteria	Stopping criterion is reaching the maximum number of epoch (300).
Training time	15 hours
CO_2eq[1]	

4. Implementation Details

4.1. Environments and requirements

A server equipped with a GTX3090 is used to deploy the training. The environments and requirements of our method is shown in Table 2.

4.2. Training protocols

Our data preprocessing pipeline is basically the same as nnUNet, but its network structure and trainer have been customized. Thanks to the architecture of nnUNet, we can inherit the nnUNetTrainerV2 base class in a plug-and-play way and easily modify it. The training protocols of the baseline method is shown in Table 3.

4.3. Testing protocols

- Pre-processing steps of the network inputs:
  The same strategy is applied as trainging steps.
- Post-processing steps of the network outputs:
  No post-processing step is used.
- If using patch-based strategy, describing the patch aggregation method:

Table 4. Quantitative results of cross validation in terms of DSC and NSD.

Training		Baseline	Ours
Liver	DSC (%)	94.5±8.09	94.5±5.57
	NSD (%)	79.3±14.9	79.1±13.7
Kidney	DSC (%)	80.4±17.0	78.8±20.9
	NSD (%)	70.9±18.4	73.0±19.3
Spleen	DSC (%)	89.5±18.0	87.2±18.6
	NSD (%)	82.0±19.3	77.3±20.1
Pancreas	DSC (%)	60.1±23.1	67.8±21.1
	NSD (%)	50.6±17.7	54.1±18.5
Running Time		145	70.0
GPU Memory		2298	3117.04

The same patch-based strategy is applied as nnU-Net [9]. Voxels close to the center are weighted higher than those close to the border, when aggregating predictions across patches.

5. Results

5.1. Quantitative results for cross validation.

Table 4 illustrates the comparison between our method and the baseline method. Our method is not much different from the original network in terms of indicators, but has an advantage in running time. Figure 3 is the corresponding violin plots of the organ segmentation performance.

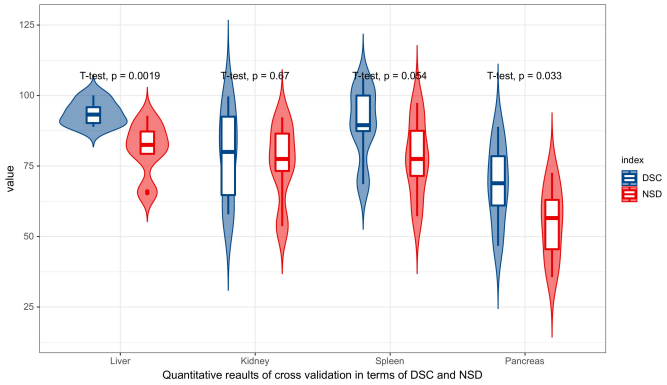

Figure 3. Violin plots of the organ segmentation results (DSC and NSD) of the 5-fold cross validation.

5.2. Qualitative results

Figure 4 presents some challenging examples. It can be seen that our method is good for relatively simple predictions, and it is basically not much different from ground truth. For the difficult predictions of the last four rows, the segmentation performance of the model for liver, kidney and spleen is not bad, but the prediction effect for the pancreas is the worst, and part of the pancreas will be missed or over-segmented.

6. Discussion and Conclusion

Our method has the worst predictions on the pancreas. This may be due to the small number of voxels and the variability of the shape of the pancreas, which makes the pancreas more difficult to predict than other organs; on the other hand, this may be related to TopK loss. If the foreground voxel with the pancreas has a better segmentation effect on the other three organs, it may not be included in the supervision, which further increases the imbalance between the pancreas foreground and background.

We introduce a new framework nnU-Net for multiple organ segmentation in the abdominal region through the classic medical image segmentation network U-Net. In order

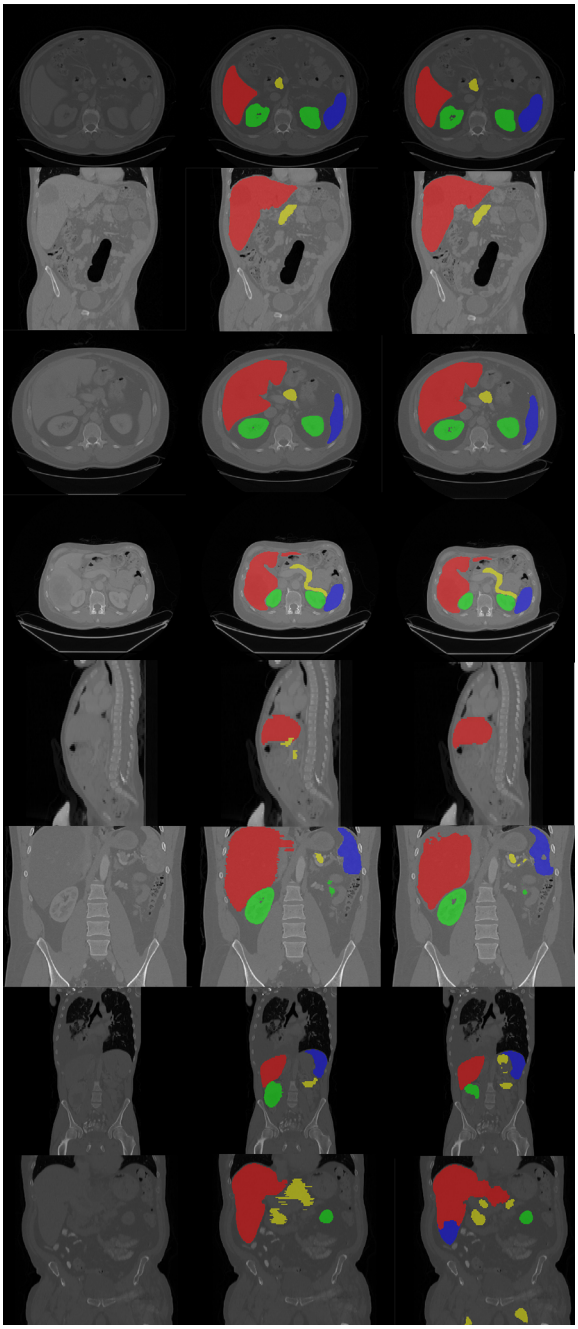

Figure 4. Challenging examples. The first four rows of the image are easy samples, the last four rows are hard samples. The first column of the examples is the voxel image, second column is the ground truth, and third column is the predicted results by our method.

to solve the lightweight problem in the segmentation process,we reduce the number of sample flips during test time augmentation and the number of levels of deep supervision. In order to improve the accuracy of the experimental results, we proposed to use the combination of Top K loss function

and Dice loss to interact with the final output results, so as to get the final segmentation results. From the results, our proposed method shows good performance.

Acknowledgment

The authors of this paper declare that the segmentation method they implemented for participation in the FLARE challenge has not used any pre-trained models nor additional datasets other than those provided by the organizers.

References

[1] A. L. Simpson, M. Antonelli, S. Bakas, M. Bilello, K. Farahani, B. Van Ginneken, A. Kopp-Schneider, B. A. Landman, G. Litjens, B. Menze *et al.*, "A large annotated medical image dataset for the development and evaluation of segmentation algorithms," *arXiv preprint arXiv:1902.09063*, 2019. 2

[2] P. Bilic, P. F. Christ, E. Vorontsov, G. Chlebus, H. Chen, Q. Dou, C.-W. Fu, X. Han, P.-A. Heng, J. Hesser *et al.*, "The liver tumor segmentation benchmark (lits)," *arXiv preprint arXiv:1901.04056*, 2019. 2

[3] H. Roth, A. Farag, E. Turkbey, L. Lu, J. Liu, and R. Summers, "Data from pancreas-ct. the cancer imaging archive (2016)." 2

[4] H. R. Roth, L. Lu, A. Farag, H.-C. Shin, J. Liu, E. B. Turkbey, and R. M. Summers, "Deeporgan: Multi-level deep convolutional networks for automated pancreas segmentation," in *International conference on medical image computing and computer-assisted intervention*. Springer, 2015, pp. 556–564. 2

[5] K. Clark, B. Vendt, K. Smith, J. Freymann, J. Kirby, P. Koppel, S. Moore, S. Phillips, D. Maffitt, M. Pringle *et al.*, "The cancer imaging archive (tcia): maintaining and operating a public information repository," *Journal of digital imaging*, vol. 26, no. 6, pp. 1045–1057, 2013. 2

[6] N. Heller, F. Isensee, K. H. Maier-Hein, X. Hou, C. Xie, F. Li, Y. Nan, G. Mu, Z. Lin, M. Han *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*, vol. 67, p. 101821, 2021. 2

[7] N. Heller, S. McSweeney, M. T. Peterson, S. Peterson, J. Rickman, B. Stai, R. Tejpaul, M. Oestreich, P. Blake, J. Rosenberg *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*, vol. 38, no. 6, pp. 626–626, 2020. 2

[8] J. Ma, Y. Zhang, S. Gu, C. Zhu, C. Ge, Y. Zhang, X. An, C. Wang, Q. Wang, X. Liu, S. Cao, Q. Zhang, S. Liu, Y. Wang, Y. Li, J. He, and X. Yang, "Abdomenct-1k: Is abdominal organ segmentation a solved problem?" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 2

[9] F. Isensee, P. F. Jaeger, S. A. Kohl, J. Petersen, and K. H. Maier-Hein, "nnu-net: a self-configuring method for deep learning-based biomedical image segmentation," *Nature Methods*, vol. 18, no. 2, pp. 203–211, 2021. 3, 4