Wide-stride V-net for Fast and Low GPU memory Abdominal Organ Segmentation

Jiapeng Zhang, Wei Liang, Ying Guo
University of Shanghai for Science and technology
Shanghai, China
201440057@st.usst.edu.cn

Abstract

U-net has been proved as the most successful segmentation architecture for medical image processing in recent years. On this basis, nnU-Net replaces the complex process of manual pipeline optimization with a systematic approach based on explicit and interpretable heuristic rules, and is proved as the SOTA method of multi-organ segmentation. However, there might be segmentation tasks for which nnU-Net's automatic adaptation is suboptimal. While target organs are affected by lesions or the performance of target organs varies between patients, if the training data is limited, the network still has the risk of overfitting. Besides, training and predicting an nnU-Net based model also requires high hardware support. This work aims to solve the problem of over-fitting of general nnU-Net and to reduce the number of model parameters. In this work, the stride of $3 \times 3 \times 2$ is used to reduce the resolution of the feature map at the end of each stage of V-Net. On the one hand, the depth of the network is reduced, which helps to alleviate the problem of overfitting under limited data; on the other hand, a larger downsampling rate allows larger input sizes while ensuring the lowest level encoder can get a larger receptive field. The parameter of our model is only 25% of the baseline. Besides, existing network frameworks (such as PyTorch) usually use full precision (Float64) for prediction. However, for intensive prediction tasks such as 3D image segmentation, the use of full-precision model parameters will greatly increase the hardware burden in the deduction process. In this work, the half-precision (Float32) is used in the prediction stage, which reduces the GPU load by about 36% without losing the prediction accuracy.

1. Introduction

Abdominal organ segmentation plays an important role in clinical practice, the state-of-the-art methods have achieved inter-observer performance in several benchmark datasets. However, most of the existing abdominal datasets only contain single-center, single-phase, single-vendor, or single-disease cases, and it is unclear whether the excellent performance can be generalized on more diverse datasets. Some SOTA methods have good general applicability, but they still perform worse than human doctors for some specific problems (for example, organs with lesions or organs with excessive differences between patients, etc.). Moreover, many SOTA methods use model ensembles to boost performance, but these solutions usually have a large model size and cost extensive computational resources, which are impractical to be deployed in clinical practice.

The encoder-decoder style architecture with skip connection was first introduced by the U-net [1]. The vast majority of successful algorithms for image segmentation in the medical domain such as residual U-net [2] and Dense U-net [3] are based on this U-shape structure. However, recent research shows that even plain U-net can achieve excellent results. The nnU-net ('no-new U-net') [4] proposed by Lsensee et al. achieves the state-of-the-art performance on six well-established segmentation challenges. Therefore, we infer that the adjustment of specific parameters for specific problems in many medical image processing tasks may be more effective than the adjustment of complex network structures.

According to the Fast and Low GPU Memory Abdominal Organ Segmentation challenge which required develop segmentation methods that can segment the liver, kidney, spleen, and pancreas simultaneously, we attempted to design our method based on the original V-Net[2]. To reduce the depth of the network while improving the global receptive field of the convolutional neural network, a larger stride is used in each downsampling stage. A larger stride also allows the model to use a larger input size, so that a more free patch-based strategy can be utilized.

2. Method

Figure 1 illustrates the applied Wide-stride V-net, where a V-Net [2] architecture is adopted. the stride of 3 × 3 × 2 is used to reduce or raise the resolution of the feature map at the down block and up block. Due to the wider stride is used, we name the architecture as Wide-stride V-Net. To make sure the the convolutional neural network(cnn) can obtain more comprehensive receptive field on the bottom layer, and to enable the patch to contain as many foreground targets as possible, the size of the feature map at each stage of the network has been fully redesigned. According to our statistics on the training data set, for an image downsampled to 256×256, the 216×189 area in the center of the image can cover all the foreground areas. Therefore, 216×189 is set as the input size of the image on the transverse plane.

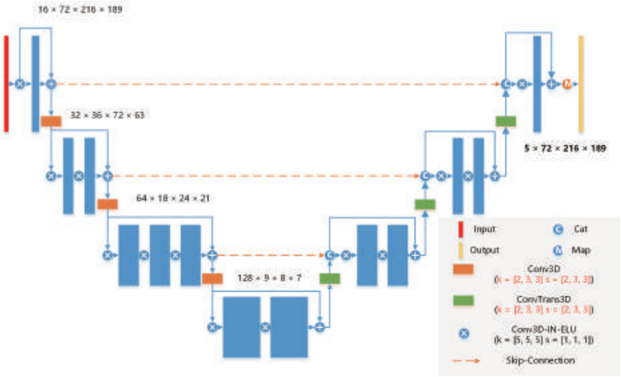

Figure 1. Network architecture

2.1. Preprocessing

The proposed method includes the following preprocessing steps:

- Cropping strategy: None.
- Resampling method for anisotropic data:
  To obtain a larger receptive field during the training process, we tend to use a relatively complete patch for training. In this way the model can capture better relative relationship between the various organs. Constrained by hardware conditions, the original image is downsampled to 256 × 256 on the transverse plane, and the axis spacing is unified to 2.5. Both in-plane and out-of-plane with third-order spline interpolation.
- Intensity normalization method:
  Considering that the evaluation indicators of the pancreas usually lower than the other classes, the dataset is clipped to the [0.5, 99.5] percentiles of the intensity values of the pancreas in the training dataset first. 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: detail of each layer, hyper-parameters, such as strides, weights size, etc. If a standard network is used, indicate the modification:

  A Wide-stride V-net is used as shown in Figure 1, all the hyper-parameters are set as the defaulted ones.
- Loss function: we use the summation between Dice loss and cross entropy loss because it has been proved to be robust [5] in medical image segmentation tasks.
- Number of model parameters: 11,592,467 (can be computed via such as torchsummary library for Pytorch)
- Number of flops: 803,651,596,160 (can be computed via such as fvcore library for Pytorch)

2.3. Post-processing

No post-processing step is used.

3. Dataset and Evaluation Metrics

3.1. Dataset

- A short description of the dataset used:
  The dataset used of FLARE2021 is adapted from MSD [6] (Liver [7], Spleen, Pancreas), NIH Pancreas [8, 9, 10], KiTS [11, 12], and Nanjing University under the license permission. For more detail information of the dataset, please refer to the challenge website and [10].
- 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.
- Note:
  In the competition phase, the model proposed in this article is trained on 289 cases in the provided training set, and validated by other 72 cases.

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 16.04.6 LTS
CPU	Intel(R) Xeon(R) CPU E5-2640 V3 @2.60GHz
RAM	8×4GB; 2.4MT/s
GPU	Nvidia Geforce RTX 2080 ×4
CUDA version	11.1
Programming language	Python3.7
Deep learning framework	Pytorch (Torch 1.8.1, torchvision 0.9.0)
Specification of dependencies	V-net

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	Randomly crop an area with a length of 72 on the axis, and ensure that each patch contains at least one foreground class.
Batch size	4
Patch size	72×216×189
Total epochs	1000
Optimizer	Adam
Initial learning rate	0.0001
Learning rate decay schedule	MultiStepLR: milestones=[300, 600, 900], gamma=0.5.
Stopping criteria, and optimal model selection criteria	Stopping criterion is reaching the maximum number of epoch (1000).
Training time	106.8 hours
CO_2eq[1]	

4. Implementation Details

4.1. Environments and requirements

The environments and requirements of the baseline method is shown in Table 2.

4.2. Training protocols

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.

- Patch aggregation method:
  According to the statistics of the masks in the training set, the patch size used in this article can cover all the foreground organs. Therefore, in the prediction stage, First, multiple patches are obtained by the center cropping method, and then the multiple predicted patches are directly spliced together on the axis to obtain the final forecast result.

- Parameter reduction strategy:
  We found that using full tensor and half tensor hardly have a significant impact on the prediction results, so in the prediction stage we compress the model accuracy to 16-bit floating-point type. This strategy can significantly reduce the memory burden during prediction.

5. Results

5.1. Quantitative results

Table 4, Table 5 and Table 6 illustrate the results on 8 validation cases(4 easy cases and four hard cases). Figure 2, Figure 3 and Figure 4 are the corresponding violin plots of the organ segmentation performance.

As mentioned before, most of the training data used in the model mentioned in this article only include the abdominal cavity area. Since most of the validation set data contain areas other than the abdominal cavity, for the DSC and NSD calculated on the 8 validation cases provided, the obtained values and the dispersed distributions observed from the violin plots indicate unsatisfying segmentation performance for all four organs.

This problem can be reflected in the comparison of the results of easy samples and hard samples. The provided 4 easy samples include 3 cases of abdominal cavity data, while the provided 4 difficult samples are all contain areas other than the abdominal cavity, including one case of from child. Considering that the running time of our methods is much less than the baseline, if the predicted area can be restricted to the abdominal cavity area or the object detection method can be used to restrict the boundary of the organ before segmentation, the method proposed in this paper will be more competitive.

Table 4. Quantitative results on validation set (provided 8 validation cases).

Organ	DSC (%)	NSD (%)
Liver	75.33	61.5
Kidney	60.98	53.88
Spleen	69.19	65.26
Pancreas	52.76	47.58

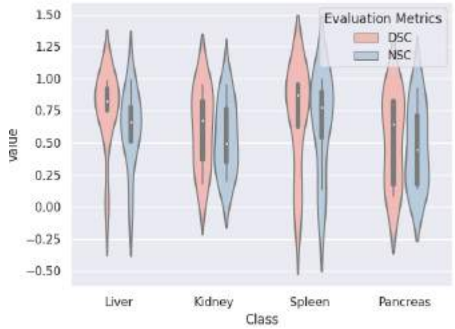

Figure 2. Violin plots of the organ segmentation results (DSC and NSD) on validation set (provided 8 validation cases).

Table 5. Quantitative results on validation set (provided 4 easy cases).

Organ	DSC (%)	NSD (%)
Liver	87.84	76.84
Kidney	80.96	73.01
Spleen	94.06	91.93
Pancreas	82.90	74.68

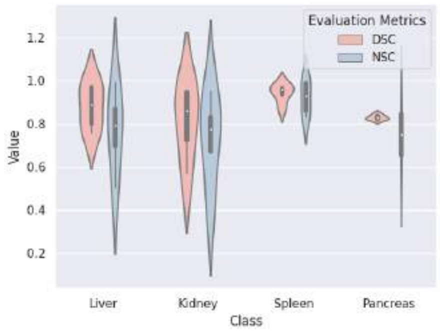

Figure 3. Violin plots of the organ segmentation results (DSC and NSD) on validation set (provided 4 easy cases).

Table 6. Quantitative results on validation set (provided 4 hard cases).

Organ	DSC (%)	NSD (%)
Liver	62.83	46.15
Kidney	41.00	34.76
Spleen	44.32	38.58
Pancreas	22.63	20.48

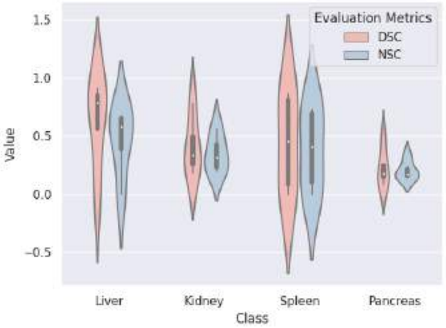

Figure 4. Violin plots of the organ segmentation results (DSC and NSD) on validation set (provided 4 hard validation cases).

Comparison with the result of baseline, the proposed method act even worse in NSD. This phenomenon may caused by our preprocess method and the patch aggregation strategy.

5.2. Qualitative results

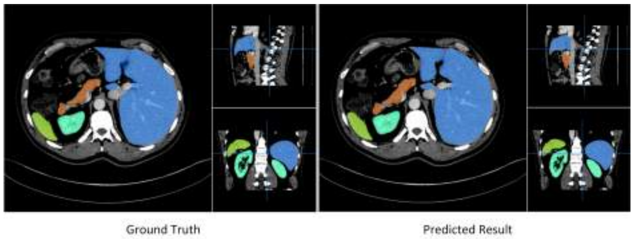

Figure 5. Predicted examples in our validation set(actually 72 cases from the provided training set).

Figure 5 presents some predicted examples in our validation set(actually 72 cases from the provided training set). It can be seen that for organs with clear and normal structures, the method used in this article will not significantly affect the prediction accuracy while reducing the amount of model parameters.

Figure 6 presents some challenging examples. It can be found that our method also can not segment the lesion-affected organs well. The first row of Figure 6 illustrates a fatty liver case where the liver is darker than healthy ones. Second row of Figure 5 shows an example with kidney (green) tumor which causes incorrect segmentation. Since these disease-affected organs rarely appear in the training set, we have to admit that our model does not yet have such specific representations for learning. In other words, although some strategies to prevent model overfitting are used, the risk of overfitting still exists.

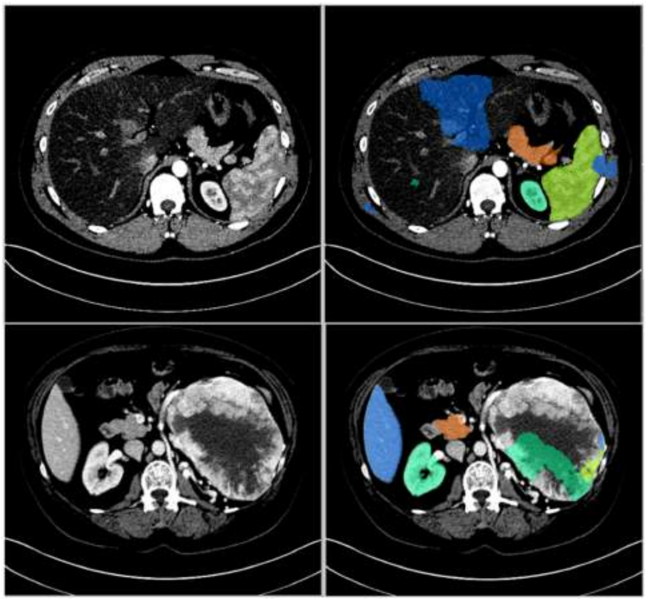

Figure 6. Some predicted result in the real validation set. First column is the image, and second column is the predicted results by our proposed method.

6. Discussion and Conclusion

Since the proposed method has a similar structure to the baseline method, they face the same problems. The proposed method also work well on cases where no diseases exist.

The existence of leison is a critical factor for the segmentation performance. Figure 7 presents the visualization of features in different stage on one case where no diseases exist. Figure 8 presents the visualization of features in different stage on one case where there are tumors on the kidney. It can be seen that the specific channel of the feature map will respond to the high-dimensional features such as shape at the specific area of the image at the corresponding stage. The diversity of lesions may cause the original model's ability to learn features in the specific area of the new sample to become meaningless. This may cause the model to generate inaccurate prediction results.

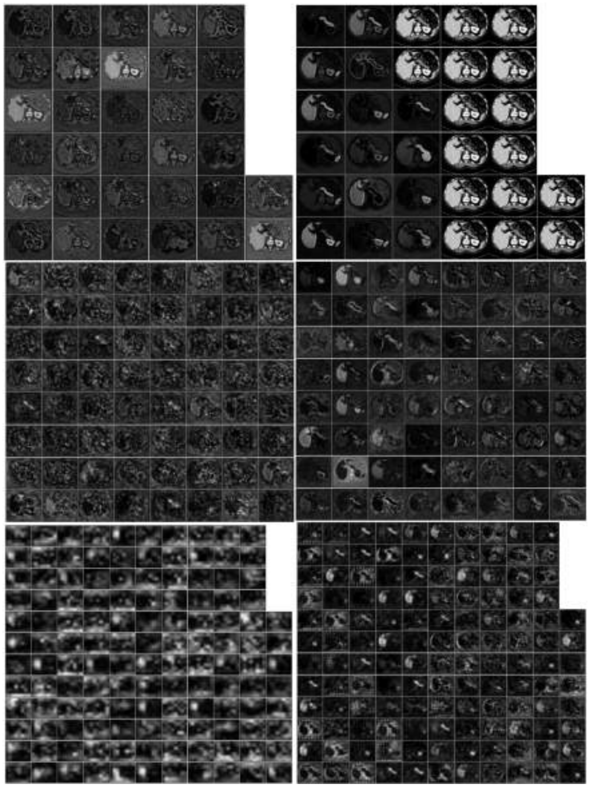

Figure 7. Visualization of features in different stage on one case where no diseases exist.

Since there are few CT images in the original training set contain areas other than the abdominal cavity, the model proposed in this paper can obtain good results in CT images that contain only the abdominal cavity, and the average speed is 10 times that of the baseline. However, part of the data in the validation set contains a large number of regions outside the abdominal cavity. Our model may produce a large number of incorrect segmentation results on such data, which will lead to lower DSC and NSD scores.

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.

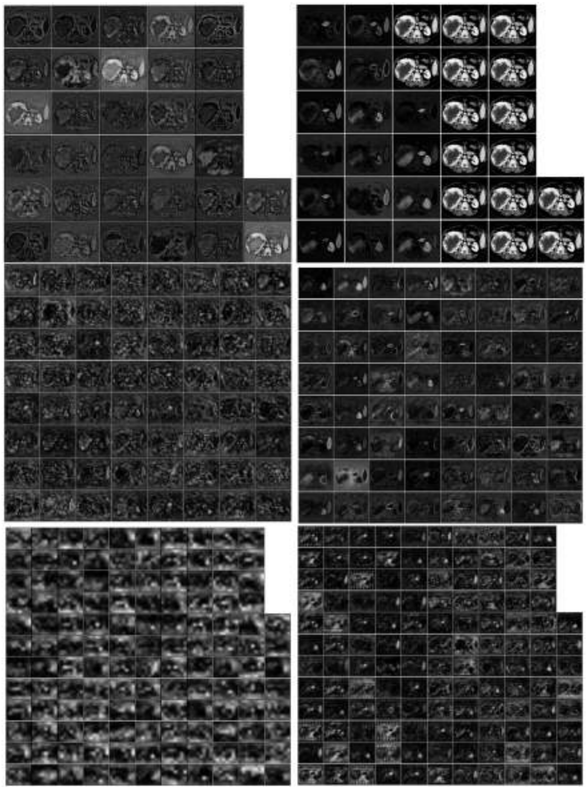

Figure 8. Visualization of features in different stage on one case where there are tumors on the kidney.

References

[1] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241. 1

[2] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *2016 Fourth International Conference on 3D Vision (3DV)*, 2016, pp. 565–571. 1, 2

[3] Z. Wang, N. Zou, D. Shen, and S. Ji, "Non-Local U-Nets for Biomedical Image Segmentation." in *AAAI*, 2020, pp. 6315–6322. 1

[4] 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. 1

[5] J. Ma, J. Chen, M. Ng, R. Huang, Y. Li, C. Li, X. Yang, and A. L. Martel, "Loss odyssey in medical image segmentation," *Medical Image Analysis*, vol. 71, p. 102035, 2021. 2

[6] 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

[7] 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

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

[9] 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

[10] 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

[11] 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

[12] 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