Re-Unet:Multi-Modality Cell Segmentation based on nnU-Net Pipeline

Haotian Lu
School of Electronic Information and Electrical Engineering
Shanghai Jiao Tong University
Shanghai, Dongchuan Rd. 800
flick-lu@sjtu.edu.cn

Jinghao Feng
School of Electronic Information and Electrical Engineering
Shanghai Jiao Tong University
Shanghai, Dongchuan Rd. 800
fjh1345528968@sjtu.edu.cn

Zelin Peng
School of Electronic Information and Electrical Engineering
Shanghai Jiao Tong University
Shanghai, Dongchuan Rd. 800
godlin_bd@126.com

Wei Shen*
School of Electronic Information and Electrical Engineering
Shanghai Jiao Tong University
Shanghai, Dongchuan Rd. 800
wei.shen@sjtu.edu.cn

Abstract

Cell segmentation is an important initial task in medical image analysis, and in recent years, data-driven deep learning methods have made groundbreaking achievements in this field. In this challenge, a multi-modal and partially labeled dataset is provided. In this paper, we propose a multi-modality cell segmentation framework called Re-Unet, which is based on the nnU-Net pipeline and an iterative self-training method. Re-Unet enriches the original data and fully considers the information of cell intervals while making full use of the semi-supervised data. Our proposed method achieves a mean F1 score of 0.6101 on the tuning set and a F1 score of 0.4492 on the testing set.

1 Introduction

Identifying clear cellular trends from microscopy images is a task of great importance as the basis for a variety of biomedical applications[1, 2, 3]. Because of sensitive nature and privacy concerns of Medical Image Analysis, researchers usually get dispersive, sometimes multi-source samples. With the development and wide application of deep learning in vision tasks[4, 5, 6], related models are widely used in microscope image analysis and have achieved outstanding results.

Besides, it's extremely time-consuming for doctors to accurately annotate the images. These lead to a multi-modality and semi-supervised segmentation task. As the samples are independently collected from different centers, we need to address big modality gap. We also want our model be robust to new modalities. These requires high-level comprehension from our model. Besides, less than half of the cases are labeled, so semi-supervised methods must be applied to utilize the unlabeled samples.

Weakly Supervised Cell Segmentation in Multi-modality High-Resolution Microscopy Images was hosted at NeurIPS 2022 to solve this porblem. Provided with 1000 labeled images and 1700+ unlabeled datasets, the competition is currently performing instance segmentation on cell data of different microscope types, staining types, and tissue types.

In this paper, we propose a model named Reunet for multimodal semi-supervised cell-instance segmentation. We first construct a fully-supervised model on labeled samples using nnU-Net [7].

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

To further improve boundary performance, we apply weight map [6]. After fine-tuning, we fuse the model into an iterative self-traning framework [8] to make use of the unlabeled cases. Finally, we convert our semantic segmentation results to instance segmentation result depending on the connectivity between the pixels. The remainder of this paper includes: **2**.Method that introduces our method in detail, **3**.Experiments that shows experimental details, **4**.Results and discussion that shows the model performance and further discussions and **5**.Conclusions.

2 Method

We use nnU-Net [7] as the backbone network for our approach. Firstly, we fine-tune the u-net model on the labeled samples. To improve boundary performance, we incorporate weight map [6] to emphasize the boundary loss. Furthermore, we employ a self-training-based semi-supervised framework [8] to utilize the unlabeled samples.

2.1 Preprocessing

Following preprocessing methods provided officially, we first create interior maps with instance segmentation masks. Specifically, we assign cell interior with 1, background with 0 and boundary with 2. By this means, we formulate our task as a pixel-wise 3-class classification problem. By differentiating boundary and interior pixels, we hope to detect overlapping cells in instance segmentation phase.

Second, we perform channel normalization on sample images to alleviate modality gaps. Given a 3-channel image, for each channel, if not empty, we linearly re-scale pixel intensities into interval $0 \sim 255$. Finally, to make them compatible with nnU-Net, we convert both the sample images and masks into .nifti format.

2.2 Proposed Method

In recent years, convolutional neural networks have significantly improved medical image segmentation. In 2015, U-Net [6] was proposed, and its variants have continued to improve segmentation performance in medical image settings. In 2018, nnU-Net [7], which integrates certain network designing principles, achieved state-of-the-art performance without manual tuning. Due to its self-adapting property and multiple data augmentations, we have built our model based on nnU-Net. To optimize boundary performance, we apply weight maps [6] to encourage the loss function to focus on boundary quality. Overall, we utilize the above network in a self-training-based semi-supervised framework to make use of unlabeled data.

2.3 Network Architecture

Our model is based on the generic U-Net architecture, which includes skip connections between different resolution stages of the encoder and decoder. Both the encoder and decoder are composed of stacks of convolutional blocks. In the encoder, we use 9 convolutional blocks, each of which consists of two sets of consecutive 3×3 convolution layers followed by instance normalization and leaky ReLU activation. We also include dropout layers. In the decoder, we use 8 convolutional blocks, each of which takes the output of the previous block, upsamples feature maps with a 2×2 transposed convolutional layer, and concatenates the output with the corresponding shortcut output from the encoder. We then feed the concatenated features into a convolutional block similar to the blocks in the encoder.

2.4 Weight Map

Due to the variety in modality, it is challenging to accurately detect cell boundaries. To address this, we adapted the weight map method proposed in [6] to emphasize boundary loss, forcing the network to learn more about boundary features. A weight map for a case is an array with the same size as the case image, which gives a weight for the loss of each pixel. To emphasize boundary areas, we give boundary pixels more weight than background and cell interior pixels. Following [6], we define the weight map of a case $\mathbf{x}$ as

$$w(\mathbf{x}) = w_c(\mathbf{x}) + w_0 \cdot exp\bigg(- \frac{(d_1(\mathbf{x}) + d_2(\mathbf{x}))^2}{2\sigma^2} \bigg)$$

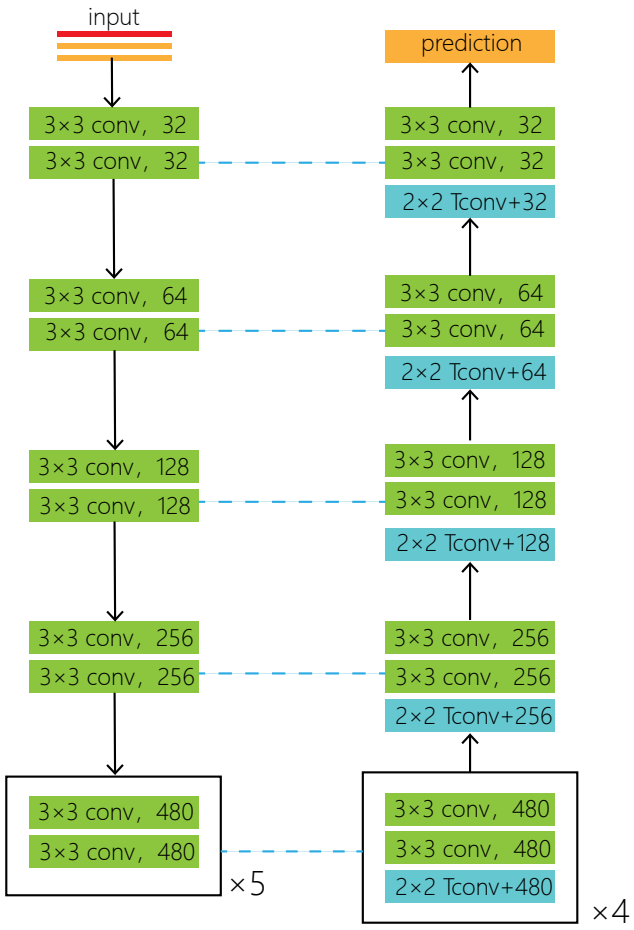

Figure 1: Network architecture. in every convolution block, the kernel size and output channel number is given. 'Tconv' denotes transposed convolution layer. The arrows denote data flows, and dotted lines shows skip connections between encoder and decoder.

where $w_c(\mathbf{x})$ denotes the class-balanced map, where we assign each pixel with the inverse of the area that the class of the pixel accounts for. $d_1(\mathbf{x})$ and $d_2(\mathbf{x})$ denote the distance to the boundary of the nearest and second nearest cells, respectively. We set $w_0 = 10$ and $\sigma = 5$. In the training phase, we compute weight maps beforehand to prevent extra time cost.

2.5 Loss function

Our loss function is composed of the unweighted summation of the Dice loss and cross-entropy loss. This has been proven to be robust for various medical image segmentation tasks [9]. Following [7], we use the multi-class version of the Dice loss variant proposed in [10], which is defined as

$$L_{Dice}(u, y) = -\frac{2}{|C|}\sum_{c}^{C}\frac{\sum_{i}^{N} u_{i,c}y_{i,c}}{\sum_{i}^{N} u_{i,c} + \sum_{i}^{N} y_{i,c}}$$

The cross-entropy loss can be written as

$$L_{CE}(y, \hat{y}) = -\frac{1}{N}\sum_{i}^{N}\sum_{c}^{C}\mathbb{I}_{y_{i,c}=1}log(\hat{y}_{i,c})$$

where N denotes the total pixel number of a case, C denotes number of classes. $y_{i,c}$ is the one-hot encoding of the ground truth mask. $\hat{y}_{i,c}$ denotes the network output, the logit probability of that pixel i belongs to class c. And $u_{i,c}$ is the softmax output of $\hat{y}_{i,c}$.

The Dice loss is weighted in response to weight map method. When calculating true positive, true negative, false positive and false negative arrays, we multiply them with the weight map in pixel-wise manner.

The total loss function is

$$L_{Total} = L_{Dice} + L_{CE}$$

2.6 Semi-supervision strategy

To utilize the unlabeled cases, we employ an iterative self-training framework [8]. We first fine-tune U-Net with weight map on the labeled cases, which serves as the initial 'teacher network'. The learned network then generates pseudo-annotations for the unlabeled cases. The pseudo-labels are thresholded into one-hot vectors. With the labeled and pseudo-labeled cases, we re-train the teacher network to obtain the updated 'student network'. This re-training process is repeated iteratively, with the updated student network serving as the new teacher network until convergence.

2.7 Result conversion

In this section, we convert our semantic segmentation results to instance segmentation result. Specifically, we divide the cell part into different connected blocks and label them according to connectivity between the pixels. In the same time, we will convert the nnU-Net result format to standard result format.

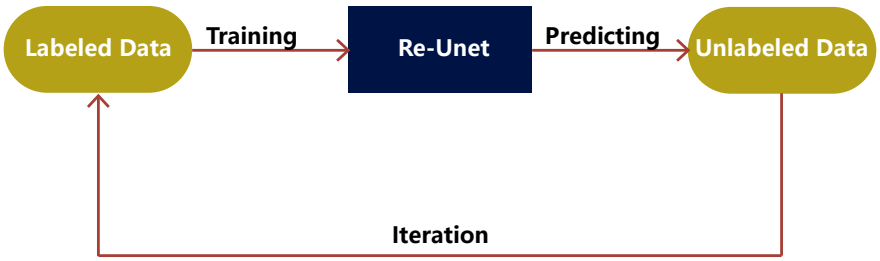

Figure 2: Schematic diagram of the iterative structure of the entire model. Our final model parameters are the structure obtained after 3 iterations

3 Experiments

3.1 Dataset

We trained our model using only the official training set. The labeled images were used to train the basic nnU-Net model, while the unlabeled images were used to train the final nnU-Net model.

3.2 Implementation details

We use SGD[11] optimizer with an initial learning rate of 1e-2 in traing baseline, and 3e-3 infinetuning. The code in implemented using PyTorch[12] and nnU-Net[7] with somie midifications. The base model structure is from nnU-Net[7]. We use 1 RTX3090 cards for training. More details are showed in followed sections.

3.2.1 Environment settings

The development environments and requirements are presented in Table 1.

Table 1: Development environments and requirements.

System	Ubuntu 18.04.6 LTS
CPU	Intel(R) Xeon(R) Silver 4210R CPU @ 2.40GHz
RAM	125GB
GPU (number and type)	Four NVIDIA GeForce RTX 3090 24G
CUDA version	11.4
Programming language	Python 3.7.13
Deep learning framework	Pytorch (Torch 1.12.1, torchvision 0.13.1)
Code	nnU-Net

3.2.2 Training protocols

Our training process consists of two parts: Baseline and Retraining. We use the Baseline model to generate a template mask for unlabeled images and then use these masks to train the Retraining model. The main training protocols are presented in Tables 3 and 4. Since our model is based on the nnU-Net baseline, the data augmentation and other processing methods are dependent on it.

Data augmentation (Based on the winning solutions in FLARE 2021, we recommend using extensive data augmentation)

patch sampling strategy during training (e.g., randomly sample 1024×1024 patches) and inference (slide window with a patch size 1024×1024)

optimal model selection criteria

Table 2: Training protocols. If the method includes more than one model, please present this table for each model seperately.

Network initialization	“he" normal initialization
Batch size	2
Patch size	80×192×160
Total epochs	1000
Optimizer	SGD with nesterov momentum ($\mu = 0.99$)
Initial learning rate (lr)	0.01
Lr decay schedule	halved by 200 epochs
Training time	72.5 hours
Number of model parameters	41.22M[1]
Number of flops	59.32G[2]

Preprocess We used the official method to normalize the images and converted the labels to three classes: cell, boundary, and background. The image and labels were then converted to the ".nii.gz" format with three modalities.

Data augmentation All of the augmentation methods we used depend on nnU-Net. We randomly crop the images to fit the size 1024×1024. Then we apply the following transformation methods to enhance the images, including **mirror transform, gamma enhancement, rotate, scale, gauss** and so on.

Deep Supervision In order to train the model effectively, we utilize Deep Supervision to calculate the loss. For the upsampling results of each size in nnU-Net, we convert them to the output space and calculate loss between corresponding size targets and the output.

Model Selection All model parameters are determined using 5-fold cross-validation. After finding the best model, the nnU-Net combines the 5 folds together and applies post-processing methods such as removing small connected components to enhance the model performance.

Predict After finding the best model, we can apply it to the testing images to obtain the segmentation result. We convert the result to three channels where each channel represents one of the three classes: cell, boundary, and background. We differentiate different cells by using the boundary channel, which separates the cells from each other.

Table 3: Training protocols for Baseline.

Network name	Baseline
Batch size	2
Patch size	3×1024×1024
Total epochs	5×500
Cross-Validation	5 fold
Optimizer	SGD with nesterov momentum ($\mu = 0.99$)
Initial learning rate (lr)	0.01
Lr decay schedule	halved by 250 epochs
Training time	42 hours
Loss function	CrossentropyLoss and DiceLoss
Number of model parameters	52.56M
Number of flops	262.61G

Table 4: Training protocols Retraining.

Network name	Retraining
Batch size	2
Patch size	3×1024×1024
Total epochs	5×200
Cross-Validation	5 fold
Optimizer	SGD with nesterov momentum ($\mu = 0.99$)
Initial learning rate (lr)	0.003
Lr decay schedule	halved by 100 epochs
Training time	16 hours
Loss function	CrossentropyLoss and DiceLoss
Number of model parameters	52.56M
Number of flops	262.61G

4 Results and discussion

4.1 Quantitative results on tuning set

Our F1 score on tuning set is 0.6101. If we only use the labeled cases (fully-supervised), the F1 score on tuning set is 0.6021. With our semi-supervised framework, though without many iterations, the unlabeled cases do provide useful information.

4.2 Qualitative results on validation set

We show an example of good segmentation results and an example of bad segmentation results are shown in Fig 2 and 3.

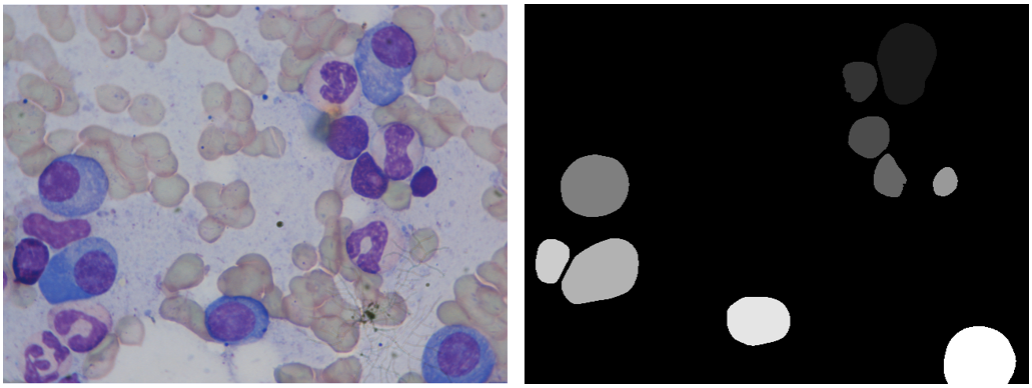

Figure 3: An example of good segmentation results.

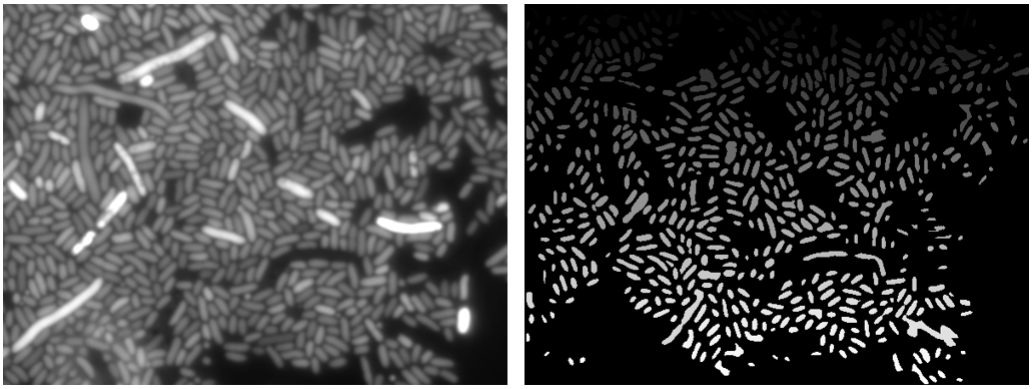

Figure 4: An example of bad segmentation results.

4.3 Segmentation efficiency results on validation set

The total running time on tuning set on our own work station is 656s. This time is the running time of the whole pipeline, including the time for data preprocessing and enhancement. The time of the core reasoning part is far less than this value.

4.4 Testing Results

Testing Results are shown in table 5

Table 5: Testing Results

Median F1-ALL	Median F1-BF	Median F1-DIC	Median F1-Fluo	Median F1-PC
0.4492	0.5889	0.4183	0.0274	0.6831
Mean F1-ALL	Mean F1-BF	Mean F1-DIC	Mean F1-Fluo	Mean F1-PC
0.4397	0.6247	0.4203	0.1159	0.5882

4.5 Limitation and future work

The way we utilize the unlabeled cases is still naive. Self-training method is an entropy minimization problem, but lacks prior knowledge. We believe that the segmentation performance can be further improved with introduction of carefully-designed restrictions. In the future, we should look deeper into the mathematical structures of these restrictions and form new models. Besides, medical images

can have very large size, which requires faster interference. We will put our efforts in the acceleration of the model in the future.

Furthermore, we hope to develop targeted data augmentation models based on data characteristics in the future. Our model does not deeply mine the characteristics of different modality data. Using generative models to generate data from different modalities may further improve our results.

It is worth mentioning that although we use a semi-supervised method to mine the information of unlabeled data, its improvement in accuracy is very limited and does not bring fundamental changes. We expect methods like self-supervised learning[13, 14, 15] to bring more improvements.

5 Conclusion

This paper proposes a multi-modality cell segmentation framework based on the nnU-Net pipeline. We apply the weight map method to improve boundary performance and utilize unlabeled cases with the self-training method. Our Re-Unet model achieves a F1 score of 0.6101 on the tuning set and 0.4492 on the testing set. Our model fully considers the information of cell intervals and enriches the original data. The self-iterative model is used to make full use of semi-supervised data. However, we did not overcome the heterogeneity of the mode and failed to make effective adjustments to the data center. Future work should focus on efficiently exploiting multimodal and unlabeled data features based on Re-Unet.

Acknowledgement

The authors of this paper declare that the segmentation method they implemented for participation in the NeurIPS 2022 Cell Segmentation challenge has not used any private datasets other than those provided by the organizers and the official external datasets and pretrained models. The proposed solution is fully automatic without any manual intervention.

References

[1] Michael Boutros, Florian Heigwer, and Christina Laufer. Microscopy-based high-content screening. *Cell*, 163(6):1314–1325, 2015.

[2] Christoph Sommer, Christoph Straehle, Ullrich Koethe, and Fred A Hamprecht. Ilastik: Interactive learning and segmentation toolkit. In *2011 IEEE international symposium on biomedical imaging: From nano to macro*, pages 230–233. IEEE, 2011.

[3] Lucas von Chamier, Romain F Laine, Johanna Jukkala, Christoph Spahn, Daniel Krentzel, Elias Nehme, Martina Lerche, Sara Hernández-Pérez, Pieta K Mattila, Eleni Karinou, et al. Democratising deep learning for microscopy with zerocostdl4mic. *Nature communications*, 12(1):2276, 2021.

[4] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[6] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[7] Fabian Isensee, Jens Petersen, Andre Klein, David Zimmerer, Paul F Jaeger, Simon Kohl, Jakob Wasserthal, Gregor Koehler, Tobias Norajitra, Sebastian Wirkert, et al. nnu-net: Self-adapting framework for u-net-based medical image segmentation. *arXiv preprint arXiv:1809.10486*, 2018.

[8] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896, 2013.

[9] Jun Ma, Jianan Chen, Matthew Ng, Rui Huang, Yu Li, Chen Li, Xiaoping Yang, and Anne L. Martel. Loss odyssey in medical image segmentation. *Medical Image Analysis*, 71:102035, 2021.

[10] Michal Drozdzal, Eugene Vorontsov, Gabriel Chartrand, Samuel Kadoury, and Chris Pal. The importance of skip connections in biomedical image segmentation. In *Deep learning and data labeling for medical applications*, pages 179–187. Springer, 2016.

[11] Léon Bottou. Stochastic gradient descent tricks. *Neural Networks: Tricks of the Trade: Second Edition*, pages 421–436, 2012.

[12] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[13] Zhaowen Li, Zhiyang Chen, Fan Yang, Wei Li, Yousong Zhu, Chaoyang Zhao, Rui Deng, Liwei Wu, Rui Zhao, Ming Tang, et al. Mst: Masked self-supervised transformer for visual representation. *Advances in Neural Information Processing Systems*, 34:13165–13176, 2021.

[14] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[15] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022.