MAUNet: Modality-Aware Anti-Ambiguity U-Net for Multi-Modality Cell Segmentation

Wangkai Li[1], **Zhaoyang Li**[1], **Rui Sun**[1], **Huayu Mai**[1], **Naisong Luo**[1], **Yuan Wang**[1], **Yuwen Pan**[1], **Guoxin Xiong**[1], **Huakai Lai**[1], **Zhiwei Xiong**[1,2], **Tianzhu Zhang**[1,2*]
[1]University of Science and Technology of China
{lwklwk, lizhaoyang, issunrui, mai556, lns6, wy2016, panyw, xgx, tbhk}@mail.ustc.edu.cn
[2]Institute of Artificial Intelligence, Hefei Comprehensive National Science Center
{tzzhang, zwxiong}@ustc.edu.cn

Abstract

Automatic cell segmentation enjoys great popularity with the development of deep learning. However, existing methods tend to focus on the binary segmentation between foreground and background in a single domain, but fail to generalize to multi-modality cell images and to exploit numerous valuable unlabeled data. To mitigate these limitations, we propose a Modality-aware Anti-ambiguity U-Net (MAUNet) in a unified deep model via an encoder-decoder structure for robust cell segmentation. The proposed MAUNet model enjoys several merits. First, the proposed instance-aware decode endows pixel features with better cell boundary discrimination capabilities benefiting from cell-wise distance field. And the ambiguity-aware decode aims at alleviating the domain gap caused by multi-modality cell images credited to a customized anti-ambiguity proxy for domain-invariant learning. Second, we prepend the consistency regularization to enable exploration of unlabeled images, and a novel post-processing strategy to incorporate morphology prior to cell instance segmentation. Experimental results on the official validation set demonstrate the effectiveness of our method. Code and models are available at `https://github.com/Woof6/neurips22-cellseg_saltfish`.

1 Introduction

Cell segmentation plays a vital role in medical image analysis, which is usually the first step for downstream single-cell analysis in microscopy image-based biology and biomedical research [1, 2, 3]. However, manual delineation is impractical since cell image datasets can be petabytes in size. Recently, with the development of computer vision, deep learning technology has been widely used in the field of semantic segmentation [4, 5, 6, 7, 8]. Inspired by the conspicuous achievements of natural image segmentation, researchers are resorting to deep learning to achieve automatic cell segmentation [9, 10, 11]. However, since the complexity of cell microscopic images (e.g., inhomogeneous illumination, diverse cell appearance, adherent cells), how to fully exploit valuable information from complicated cell images for accurate segmentation is thus extremely challenging.

Top-performing cell segmentation methods [12, 10, 13, 14, 15] tend to utilize the binary mask to represent the cells, concentrating on distinguishing between foreground and background but neglecting to focus on the edges of cells. To exploit more cell prior information, some methods [16, 17] consider using cell boundaries to further impose constraints, but they struggle to discriminate between different individual cell entities, leading to sub-optimal results ascribed to simple classifiers utilized. Some

*Corresponding author

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

methods [18, 19] introduce predicting the distance-transform map for joint learning to improve the performance, but they only consider this shape prior as an auxiliary task in the training phase, while we further take full advantage of it in the inference stage based on image morphological processing to enhance segmentation results. In other words, these methods can neither deal with the phenomenon of cell adhesion nor handle the massive differences in cell morphology, resulting in **instance confusion**.

In this paper, we propose a solution to the NeurIPS 2022 Cell Segmentation Challenge [20] which aims at efficiently segmenting cell instances in multi-modality microscopy images (see Figure 1a), contains massive labeled and unlabeled data for microscopy biology and biomedical research. Starting from the characteristics of the competition data, we deem that two aspects need to be considered. **Multi-modality ambiguity.** It is essential to differentiate the features near the decision boundary of the model in the feature space. Due to the large gap among the modalities of microscopy images, the distribution of pixel features from different domains is of great variety. As shown in Figure 1b, the model which usually separates foreground and background well on certain domains tends to suffer from ambiguity in the domains close to the decision boundary. This ambiguity will lead to not only poor performance in these indistinguishable regions but also inferior generalization ability of models in other domains. **Utilization of unlabeled data.** It is vital to make full use of the precious unlabeled data in the model training process. Limited labeled data will lead to the risk of model underfitting and overfitting, while unlabeled data is relatively easier to obtain, containing more beneficial information that can be mined.

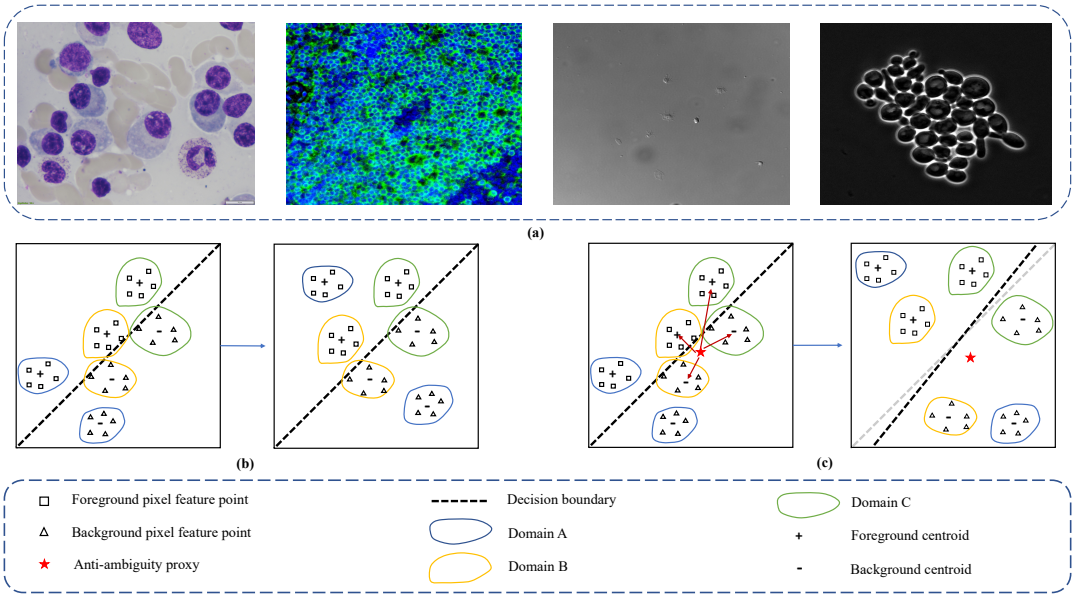

Figure 1: Illustration of our motivation. (a) shows the multi-modality cell images, including brightfield, fluorescent, differential interference contrast, and Phase-contrast. (b) shows the model that usually separates foreground and background well in certain domains tends to suffer from ambiguity on the domains close to the decision boundary. (c) shows the customized anti-ambiguity proxy for domain-invariant learning to mitigate the domain gap caused by multi-modality cell images.

Based on the above discussion, we propose a **M**odality-aware **A**nti-ambiguity **U-Net** (MAUNet), which consists of a representation encoder, an instance-aware decoder, and an ambiguity-aware decoder. In the **representation encoder**, we obtain multi-scale features from different stages of the CNN backbone [21, 22] for fine-grained cell representation. In the **instance-aware decoder**, we introduce the cell-wise distance field prediction to better distinguish each cell instance. Specifically, we fuse the hierarchical features derived from the representation encoder to obtain high-resolution pixel-level features following U-Net [12]. Then we feed the features into a regression head to get the distance of each pixel from the background region, which is used in the post-processing to alleviate the problem of cell adhesion. In the **ambiguity-aware decoder**, we propose an anti-ambiguity proxy to improve decision boundary discrimination capabilities in different domains, resulting in the explicit foreground, background, and boundary mask. As shown in Figure 1(b), the classifier can not separate

foreground pixels and background pixels in some specific domains (e.g., *Domain B* and *C*). As shown in Figure 1(c), the anti-ambiguity proxy serves as an additional class to push foreground pixels and background pixels away in the confusion area. Thus we can learn features with better-discriminating properties across different domains and alleviate the performance degradation caused by inter-domain varieties. Additionally, we attempt to implement the semi-supervised learning strategy based on consistency regularization (CR) [23] for training to exploit the abundant unlabeled data. Based on the cell masks and distance fields output by two decoders, we propose a novel post-processing process to capture the cell morphology and predict each cell instance segmentation. Our main contributions are summarized as follows:

– We propose a novel Modality-aware Anti-ambiguity U-Net (MAUNet) in a unified deep model via an encoder-decoder structure for robust cell segmentation. The proposed instance-aware decode endows pixel features with better boundary discrimination capabilities. And the ambiguity-aware decode aims at alleviating the domain gap credited to a customized anti-ambiguity proxy for domain-invariant learning.
– We prepend the consistency regularization to enable exploration of unlabeled images, and a novel post-processing strategy to incorporate morphology prior to cell instance segmentation.
– Experimental results on the official validation set demonstrate the effectiveness of our method.

2 Method

In this paper, we devise a Modality-aware Anti-ambiguity U-Net (MAUNet) including a representation encoder, an instance-aware decoder, and an ambiguity-aware decoder (see Figure 2). For each cell image, we perform preprocessing (Sec. 2.1) to normalize the image and obtain image features by representation encoder (Sec. 2.2.1), and then the resultant features are endowed with better cell boundary discrimination capabilities benefiting from the the instance-aware decoder (Sec. 2.2.2). Furthermore, assembled with the ambiguity-aware decoder (Sec. 2.2.3), the domain gap caused by multi-modality cell images can be alleviated credited to a customized anti-ambiguity proxy for domain-invariant learning. Besides, we prepend the consistency regularization (Sec. 2.2.4) to enable exploration of numerous unlabeled cell images. Finally, a novel post-processing strategy (Sec. 2.3) is developed to capture the cell morphology and predict each cell instance segmentation.

2.1 Preprocessing

We apply normalization to the images before they are fed into the network. In detail, We transform the images from grayscale mode to RGB mode firstly by copying images by three times and stacking them in channel dimensions. Then we scale the intensity level of the RGB images to the range of [0.01, 0.99]. For the cell instance labels, we convert them into three semantic categories: cell interior, boundary, and background. In view of the differences in the resolution of images and the size of cells, we adaptively set the thickness of the cell boundary according to the size of each cell instance: $t_i = \frac{\sqrt{S_i}}{20}$, where S_i denotes the number of pixels for the i-th cell instance. Then we generate the corresponding distance transformation for pixels belonging to each cell instance: $d_i = \frac{1}{1+\alpha\beta\gamma}$, where $\alpha = \frac{1}{\sqrt{S_i}}$, γ represents the distance between every pixel in i-th cell to the center of the cell, and β is a hyper-parameter (we set to 1) to control the distribution of d_i. For those pixels of the background, the value of distance transformation is 0.

2.2 Modality-aware Anti-ambiguity U-Net

2.2.1 Representation Encoder

The representation encoder aims at extracting multi-scale cell representation, which is compatible with any backbone architecture. To reduce the computational cost, we use the convolution-based backbone ResNet50 [21] and WideResNet50 [22] pretrained on ImageNet [24] to obtain generic hierarchical feature representation. We remove the last stage of WideResNet50 [22] in order to make its number of parameters close to that of ResNet50 [21].

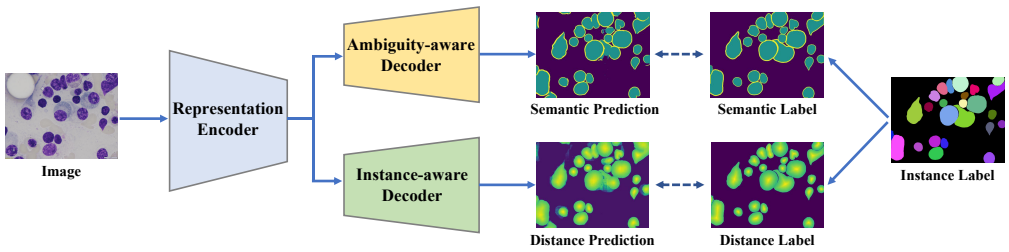

Figure 2: Network architecture of the MAUNet, which consists of a representation encoder, an instance-aware decoder, and an ambiguity-aware decoder.

2.2.2 Instance-aware Decoder

The instance-aware decoder is responsible for predicting cell-wise distance field by the upsampling pathway following U-Net [12]. Concretely, features with different resolutions extracted by the representation encoder are fed to the instance-aware decoder via skip connection. Features from stage i of the encoder are reshaped to the size of $\frac{H}{2^i} \times \frac{W}{2^i}$ and input into the residual block composed of two convolutional layers with 3×3 kernel, followed by an instance normalization layer. Then, the resolution of the feature maps is increased by a factor of 2 using a deconvolutional layer and the output is concatenated with the one of the previous stages. Repeatedly, the concatenated features are fed into the next residual block. Finally, we utilize a regression head, which is implemented by a 1×1 convolution layer, to predict the cell-wise distance field (i.e., the distance of each pixel from the background region).

2.2.3 Ambiguity-aware Decoder

The ambiguity-aware decoder is responsible for predicting the foreground, background and boundary mask through a similar upsampling architecture with the instance-aware decoder followed by three classification heads. However, since the dataset includes images from different domains, the centroids of features vary with domains. When we push away centroids of two categories (i.e., foreground and background) in one modality, the centroids in another modality may be pulled close, leading to poor discrimination. Besides, when the data is from an unknown domain, the learned classifier has no ability for domain generalization without extra domain constraints.

To alleviate the multi-modality ambiguity problem, we propose an anti-ambiguity proxy to improve decision boundary discrimination capabilities in different domains. We assume that there exists a domain-invariant category and introduce a proxy to represent it. By pushing features away from the proxy, it will provide the common reference for multi-domain features, resulting in better discriminating property. Specifically, we add an additional classification head at the end of the decoder and it produces the predication of the proxy. Then the segmentation will be calculated by the soft-max operation on predictions of all four classifiers. Our experiments show that the introduced anti-ambiguity proxy can improve domain generalization, resulting in higher performance.

2.2.4 Consistency Regularization

We train the model with unlabeled data by imposing consistency regularization between the classification and regression results from two decoders. As shown in Figure 3, we set up our framework with two branch networks (Net_1 and Net_2) with the same architecture but different initialization, which is a popular paradigm in semi-supervised learning. In the training stage, for the ambiguity-aware decoder, the argmax operation is applied to the classification result from both Net_1 and Net_2, and the output is considered as the pseudo label to supervise each other. For the instance-aware decoder, we directly minimize the distance between the regression result from two networks. For efficient inference, we just utilize Net_1 to generate the final segmentation result in the inference stage.

2.2.5 Loss Function

For images with ground truth Y_{cls} and Y_{reg}, denoting the classification label and distance transformation label respectively, we use the summation of Dice loss and focal loss to supervise the classification

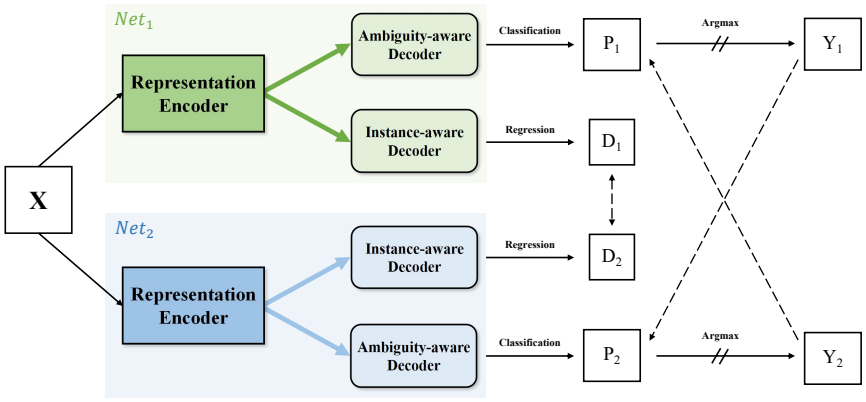

Figure 3: Illustration of Consistency Regularization. This framework consists of two networks with the same architecture and the pseudo labels are generated to supervise each other.

result of two branches of the ambiguity-aware decoder.

$$\mathcal{L}_{cls} = \frac{1}{N}\sum_{i=1}^{N}(\mathcal{L}_{dice}(P_{cls}, Y_{cls}) + \mathcal{L}_{focal}(P_{cls}, Y_{cls})), \tag{1}$$

where N denotes the number of labeled images, P_{cls} means the classification result. The weighted average L1 loss is used to supervise the regression result for distance transformation from two branches of instance-aware decoder:

$$\mathcal{L}_{reg} = \frac{1}{N}\sum_{i=1}^{N}(\frac{1}{\|M\|_1}\||Y_{reg} - P_{reg}| \cdot M\|_1 + \frac{1}{\|1-M\|_1}\||Y_{reg} - P_{reg}| \cdot (1-M)\|_1), \tag{2}$$

where P_{reg} means the regression result for distance transformation, $M = \mathbb{1}(Y_{reg} > 0)$ represent the mask of foreground, $\cdot$ denotes element-wise multiplication and $\|\cdot\|_1$ means L1 norm of the matrix (e.g., element-wise summation of the matrix).

For images without ground truth, we employ cross entropy loss and L2 loss for classification result and regression result to impose consistency regularization respectively:

$$\mathcal{L}_{cr} = \frac{1}{N}\sum_{i=1}^{N}(\mathcal{L}_{CE}(P_{cls,1}, \hat{P}_{cls,2}) + \mathcal{L}_{CE}(P_{cls,2}, \hat{P}_{cls,1}) + \mathcal{L}_2(P_{reg,1}, P_{reg,2})) \tag{3}$$

where $P_{cls,j}$ and $P_{reg,j}$ $(j = 1, 2)$ denote classification result and regression result from j-th branch, $\hat{P}_{cls}$ means the argmax output of P_{cls}. As a result, our network is trained by minimizing the overall objective as follows:

$$\mathcal{L} = \mathcal{L}_{cls} + \mathcal{L}_{reg} + \lambda\mathcal{L}_{cr}, \tag{4}$$

where λ is the trade-off weight.

2.3 Post-processing

Our post-processing strategy includes the following steps and is shown in Figure 4:

- **Step 1.** According to the probability map of classification, we obtain the mask M_{fo} with a background probability less than 0.3 as the representation for the foreground. Then we get the mask M_{in} with cell interior probability greater than 0.7 as the representation for the main body of the cell. We also calculate the distance transformation map noted as D_t.
- **Step 2.** We can roughly calculate the average size of cells according to M_{in}, and then adaptively select the size of the Gaussian kernel for fuzzy processing of predicted distance transformation to obtain the D_{tf}. Specifically, we choose values larger than 0.8 in the distance transform map as cell skeleton D_{ts}, which is considered as the marker of the watershed algorithm, and M_{in} is taken as the template to get the preliminary segmentation result in P_1. This step enables us to find most of the cell instances.

- **Step 3.** The peak selection algorithm is adopted to seek the extreme points in the distance transformation map to locate the small cells in the image that are ignored in step 2 and we note it as D_{tp}. Then the watershed algorithm takes these points as markers and M_{in} as the template. This step supplements the original segmentation result and obtains the relatively refined one P_2.
- **Step 4.** Finally, we use the aggregation of P_1 and P_2 as the marker and M_{fo} as the template to conduct the watershed algorithm the third time. This enables us to obtain a more accurate segmentation map. A remove-small-object algorithm will be applied in the end to filter the noise in the final segmentation result in P_{final}.

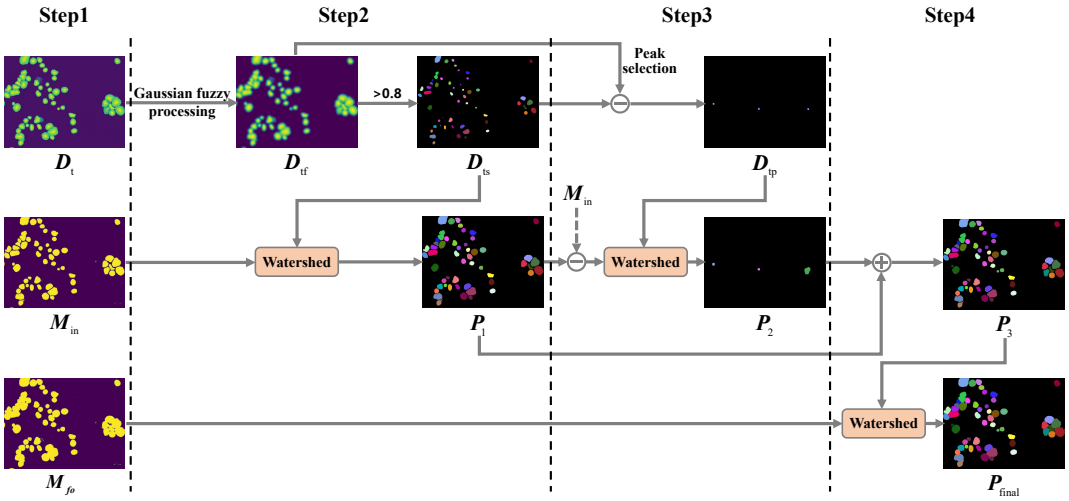

Figure 4: Pipeline for post-processing. This multi-step process enables the model to generate fine-grained segmentation results.

3 Experiments

3.1 Dataset and Evaluation Metrics

We only use the dataset provided by the NeurIPS 2022 Cell Segmentation Challenge [20]. The training set includes 1,000 labeled image patches from various microscopy types (e.g., brightfield, fluorescent, phase-contrast, and differential interference contrast), tissue types, staining types, and more than 1,500 unlabeled images. The validation set contains a total of 101 images from different domains, including a whole-slide image (about 10,000×10,000). The testing set includes 200+ images, some of which are from unseen domains.

The evaluation metrics consist of the $F1$ score with 0.5 as the threshold and running time, both of them will be used in the ranking scheme. Besides, the GPU memory consumption has a 10 GB tolerance and RAM has a 28 GB tolerance.

3.2 Implementation Details

3.2.1 Environment Settings

The development environments and requirements are presented in Table 1.

3.2.2 Training Protocols

Data Augmentation We first apply random resize to input images in a range from 0.25 to 2, with padding and random cropping to sample 512×512 patches. Then we adopt data augmentation including flip, rotation, Gaussian noise, contrast adjustment, Gaussian smoothing, and histogram shift during training. Our inference adopts slide window with a patch size 512×512.

Table 1: Development environments and requirements.

System	Ubuntu 22.04.1 LTS
CPU	Intel(R) Xeon(R) CPU E5-2695 v4 @ 2.10 GHz
RAM	16×4 GB; 2.67 MT/s
GPU (number and type)	2 NVIDIA Titan RTX (24G)
CUDA version	11.3
Programming language	Python 3.10.4
Deep learning framework	Pytorch (Torch 1.11.0, torchvision 0.12.0)
Specific dependencies	monai 0.9.0
Code	https://github.com/Woof6/neurips22-cellseg_saltfish

Parameter Setting We train two MAUNet with different backbones respectively and merge them together in the prediction stage. The setting for fully-supervised protocol is presented in Table 2. The setting for semi-supervised protocol is presented in Table 3, and we set λ to 1.5.

Table 2: Fully-supervised protocols.

Network initialization	default normal initialization by pytorch
Batch size	16
Patch size	512×512
Total epochs	1,000
Optimizer	Adamw
Initial learning rate (lr)	6e-4
Lr decay schedule	multiply by 0.95 for every 10 epochs
Training time	24.0 hours
Loss function	$\mathcal{L}_{dice} + \mathcal{L}_{focal} + w\mathcal{L}_1$
Number of model parameters	39.40 M (Res50), 28.84 M (WideRes50)[1]
Number of flops	10.09 G (Res50), 14.31 G (WideRes50)[2]

Table 3: Semi-supervised protocols.

Network initialization	default normal initialization by Pytorch
Batch size	8 for labeled images and 8 for unlabel images
Patch size	512×512
Total epochs	700
Optimizer	AdamW
Initial learning rate (lr)	6e-4
LR decay schedule	multiply by 0.95 for every 10 epochs
Training time	36.0 hours
Loss function	$\mathcal{L}_{dice} + \mathcal{L}_{focal} + w\mathcal{L}_1 + \lambda\mathcal{L}_{cr}$

4 Results and Discussion

4.1 Quantitative Results on Validation Set

To analyze the effect of the proposed method, we sample 100 labeled images from the train set and use the other 900 to train our model. And we report the mean dice score for the semantic segmentation metric and $F1$ score with 0.5 as the threshold for the instance segmentation metric. All the networks are built with resnet50 as the backbone and we evaluate the effect of regression loss ($\mathcal{L}_{reg}$), Consistency Regularization (CR), and Anti-Ambiguity Proxy (AAP) respectively. Table 4 shows our results, where $F1_{\text{ori}}$ represents the $F1$ score without our proposed post-process strategy.

[1] `https://github.com/sksq96/pytorch-summary`

[2] `https://github.com/facebookresearch/fvcore`

Table 4: Ablation study for our method.

Method	**Mean Dice**	$F1_{\text{ori}}$	$F1$
MAUNet-R50	0.7499	0.7310	0.8309
MAUNet-R50 w/o $\mathcal{L}_{reg}$	0.7458	0.7284	-
MAUNet-R50 w/ CR	0.7443	0.7075	0.8217
MAUNet-R50 w/ CR*	0.7403	0.7101	0.8243
MAUNet-R50 w/ AAP	0.7543	0.7318	0.8494

As shown in Table 4, we implement the MAUNet with only the Ambiguity-aware Decoder and there is only classification loss for training. We find that the introduced Instance-aware Decoder leads to the performance gain for classification. This paradigm for multi-task learning provides more information for supervision and is beneficial for representation learning. The CR method doesn't work well in our task. We argue that since our data set from different domains includes multiple centers but only three categories for classification, it will lead CR to generate a large number of inaccurate labels, especially for some unusual domains. We observe that these consistency constraints will degrade the performance greatly when our model makes inferences for some rare data. We add CR directly to the classification loss (CR*) and obtain similar results. We also find that the introduced Anti-Ambiguity Proxy can improve the performance in this case. We will discuss it in the next section. For the submission, we do not use CR and only labeled data is used to train our model. Table 5 shows our $F1$ score on the official validation set, we adopt the test time augmentation by merging multi-scale inference from 1, 1.25, and 1.5 times the original size. As mentioned before, our complete model includes two MAUNet using different backbones.

Table 5: Results on official validation set under different backbone protocols.

Method	MAUNet-R50	MAUNet-WR50	Ensemble
$F1$	0.8162	0.8211	0.8250

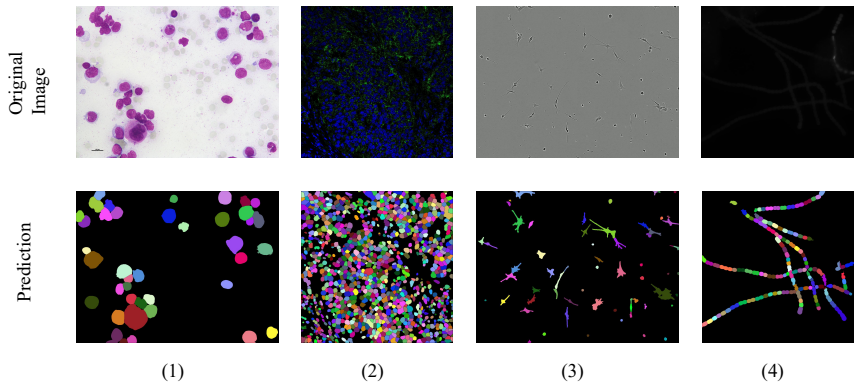

Figure 5: Examples on the validation set. We present the segmentation results for images from different microscopy types.

4.2 Qualitative Results on Validation Set

Some examples from our segmentation results for the validation set are shown in Figure 5. We find that our method works well in most cases. But when cells have an irregular shape or are not convex (e.g., Figure 5(3)), the watershed algorithm in post-processing will lead to over-segmentation. And if the cells are not imaged intensely enough (e.g., Figure 5(4)), they will be regarded as background. We deem the reason is that we just apply a simple intensity normalization during training and inference.

We also visualize the features learned with AAP, as shown in Figure 6. For each image, we respectively calculate the average features within the background mask and the cell interior mask and apply the PCA algorithm for dimension reduction to obtain the scatter diagram in Figure 6(a). We find that

the distribution of features has multiple centroids, and the introduction of the proxy can significantly enhance the discriminability of features. Then we visualize the distribution of features from two specific domains and define the weight of classifiers as proxies in Figure 6(b). We estimate the decision boundary according to the class proxies. With the introduction of AAP, the network learns the decision boundary with better performance, which exactly conforms to the illustration of motivation in Figure 1. We visualize some examples of the ablation of AAP. As shown in Figure 7, there is bright light pollution in the original image, the initial model performs extremely poorly in this case. While this performance loss is effectively mitigated after the introduction of the extra classification head.

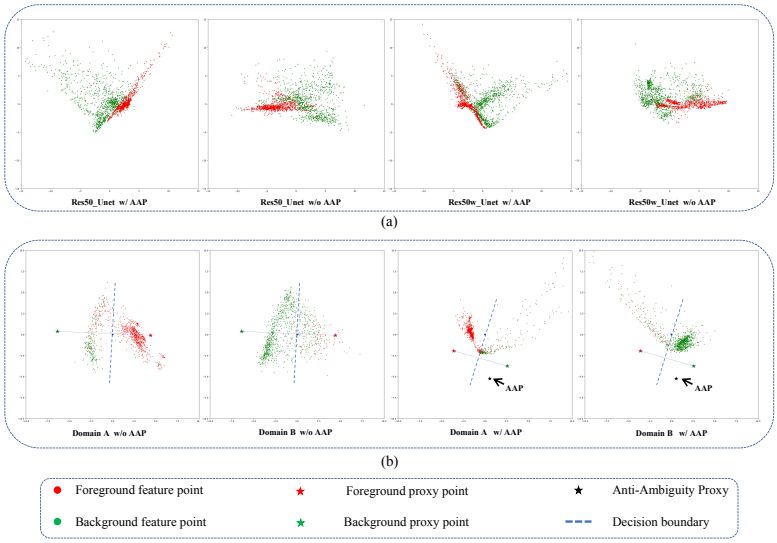

Figure 6: Visualization of features. The features encoded by models with AAP are more distinguishable and their distribution is more concentrated.

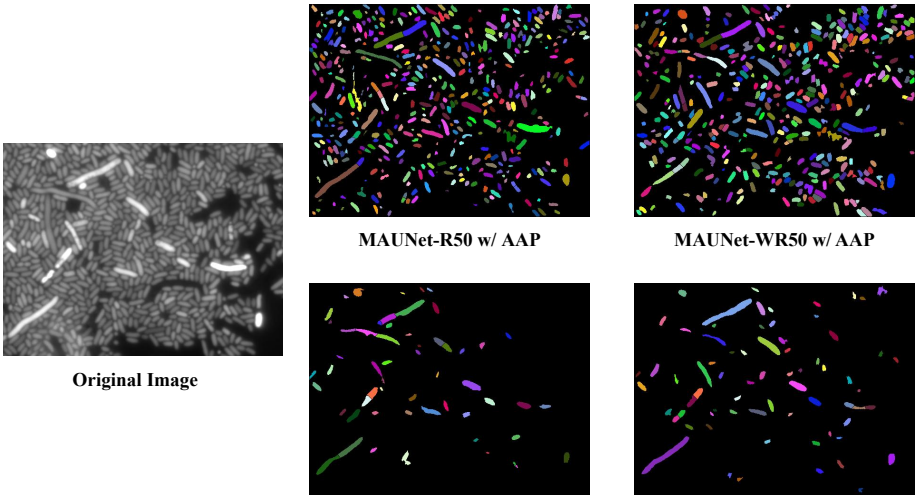

Figure 7: Example for ablation of AAP. In this case, the original models perform poorly because of bright light interference, while the networks with AAP reveal better-discriminating power.

4.3 Segmentation Efficiency Results on Validation Set

Our approach achieves efficient inference. In Table 6, we report the efficiency evaluation results on our personal server with 28 GB RAM, AMD EPYC 7713 CPU, and RTX-3090 GPU using the

official evaluation code. [2]. As Table 6 shows, the inference time and the GPU memory usage increase as the resolution of the testing image increases. But the inference time of our method does not exceed the time tolerance given by officials, demonstrating the efficiency of our method.

Table 6: Efficiency evaluation results of our submitted docker.

Resolution	Docker inference time(s)	GPU Memory(MiB)
640×480	9.67	2,884
1,024×1,024	10.35	3,024
3,000×3,000	18.56	5,886
8,415×10,496	74.16	8,592

4.4 Results on Final Testing Set

Table 7 shows the results of the proposed MAUNet on the final test set, which includes four microscopy modalities.

Table 7: The quantitative results on the final test set

$F1$-Score	Brightfield	DIC	Fluorescence	Phase Contrast	All
Median	0.9024	0.7293	0.235	0.8327	0.749
Mean	0.8908	0.683	0.3076	0.7314	0.6489

4.5 Limitations and Future Work

In a semi-supervised semantic segmentation task, the performance of the model can be improved by imposing consistency regularization on a large amount of unlabeled data. But we do not take advantage of such a learning paradigm since the distinguishing ability of the model will degrade when transferred to different domains. We will refer to the growing research progress and hope to utilize unlabeled data to improve the performance and generalization ability of our model in the future.

5 Conclusion

In this paper, we propose a Modality-aware Anti-ambiguity U-Net (MAUNet) in a unified deep model via an encoder-decoder structure for robust cell segmentation, and the proposed instance-aware decode endows pixel features with better cell boundary discrimination capabilities. And the ambiguity-aware decode aims at alleviating the domain gap credited to a customized anti-ambiguity proxy for domain-invariant learning. Besides, we prepend the consistency regularization to enable the exploration of unlabeled images, and a novel post-processing strategy to incorporate morphology prior to cell instance segmentation. Experimental results on the official validation set demonstrate the effectiveness of our method.

Acknowledgement

This work was partially supported by the National Nature Science Foundation of China (Grant 62022078, Grant 62021001).

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.

[2] https://github.com/JunMa11/NeurIPS-CellSeg/blob/main/baseline/cellseg_time_eval.py

References

[1] Tony Yeung, Penelope C Georges, Lisa A Flanagan, Beatrice Marg, Miguelina Ortiz, Makoto Funaki, Nastaran Zahir, Wenyu Ming, Valerie Weaver, and Paul A Janmey. Effects of substrate stiffness on cell morphology, cytoskeletal structure, and adhesion. *Cell motility and the cytoskeleton*, 60(1):24–34, 2005.

[2] Elisabeth E Charrier, Katarzyna Pogoda, Rebecca G Wells, and Paul A Janmey. Control of cell morphology and differentiation by substrates with independently tunable elasticity and viscous dissipation. *Nature communications*, 9(1):1–13, 2018.

[3] Carsen Stringer, Tim Wang, Michalis Michaelos, and Marius Pachitariu. Cellpose: a generalist algorithm for cellular segmentation. *Nature methods*, 18(1):100–106, 2021.

[4] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[6] 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.

[7] Rui Sun, Yihao Li, Tianzhu Zhang, Zhendong Mao, Feng Wu, and Yongdong Zhang. Lesion-aware transformers for diabetic retinopathy grading. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10938–10947, 2021.

[8] Yuan Wang, Rui Sun, Zhe Zhang, and Tianzhu Zhang. Adaptive agent transformer for few-shot segmentation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIX*, pages 36–52. Springer, 2022.

[9] Noah F Greenwald, Geneva Miller, Erick Moen, Alex Kong, Adam Kagel, Thomas Dougherty, Christine Camacho Fullaway, Brianna J McIntosh, Ke Xuan Leow, Morgan Sarah Schwartz, et al. Whole-cell segmentation of tissue images with human-level performance using large-scale data annotation and deep learning. *Nature biotechnology*, 40(4):555–565, 2022.

[10] Thorsten Falk, Dominic Mai, Robert Bensch, Özgün Çiçek, Ahmed Abdulkadir, Yassine Marrakchi, Anton Böhm, Jan Deubner, Zoe Jäckel, Katharina Seiwald, et al. U-net: deep learning for cell counting, detection, and morphometry. *Nature methods*, 16(1):67–70, 2019.

[11] Carsen Stringer, Tim Wang, Michalis Michaelos, and Marius Pachitariu. Cellpose: a generalist algorithm for cellular segmentation. *Nature methods*, 18(1):100–106, 2021.

[12] 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.

[13] David A Van Valen, Takamasa Kudo, Keara M Lane, Derek N Macklin, Nicolas T Quach, Mialy M DeFelice, Inbal Maayan, Yu Tanouchi, Euan A Ashley, and Markus W Covert. Deep learning automates the quantitative analysis of individual cells in live-cell imaging experiments. *PLoS computational biology*, 12(11):e1005177, 2016.

[14] Reka Hollandi, Abel Szkalisity, Timea Toth, Ervin Tasnadi, Csaba Molnar, Botond Mathe, Istvan Grexa, Jozsef Molnar, Arpad Balind, Mate Gorbe, et al. nucleaizer: a parameter-free deep learning framework for nucleus segmentation using image style transfer. *Cell Systems*, 10(5):453–458, 2020.

[15] Linfeng Yang, Rajarshi P Ghosh, J Matthew Franklin, Simon Chen, Chenyu You, Raja R Narayan, Marc L Melcher, and Jan T Liphardt. Nuset: A deep learning tool for reliably separating and analyzing crowded cells. *PLoS computational biology*, 16(9):e1008193, 2020.

[16] Hao Chen, Xiaojuan Qi, Lequan Yu, and Pheng-Ann Heng. Dcan: deep contour-aware networks for accurate gland segmentation. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2487–2496, 2016.

[17] Tran Minh Quan, David Grant Colburn Hildebrand, and Won-Ki Jeong. Fusionnet: A deep fully residual convolutional neural network for image segmentation in connectomics. *Frontiers in Computer Science*, page 34, 2021.

[18] Fernando Navarro, Suprosanna Shit, Ivan Ezhov, Johannes Paetzold, Andrei Gafita, Jan C Peeken, Stephanie E Combs, and Bjoern H Menze. Shape-aware complementary-task learning for multi-organ segmentation. In *Machine Learning in Medical Imaging: 10th International Workshop, MLMI 2019, Held in Conjunction with MICCAI 2019, Shenzhen, China, October 13, 2019, Proceedings 10*, pages 620–627. Springer, 2019.

[19] Junlin Hou, Jilan Xu, Longquan Jiang, Shanshan Du, Rui Feng, Yuejie Zhang, Fei Shan, and Xiangyang Xue. Periphery-aware covid-19 diagnosis with contrastive representation enhancement. *Pattern Recognition*, 118:108005, 2021.

[20] Weakly supervised cell segmentation in multi-modality high-resolution microscopy images. `https://neurips22-cellseg.grand-challenge.org/`.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[22] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[23] Xiaokang Chen, Yuhui Yuan, Gang Zeng, and Jingdong Wang. Semi-supervised semantic segmentation with cross pseudo supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2613–2622, 2021.

[24] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.