Cell Segmentation in Multi-modality High-Resolution Microscopy Images with Cellpose

Kwanyoung Lee†
Department of Intelligence Computing
Hanyang University
222 Wangsimni-ro, Seongdong-gu, Seoul, Republic of Korea
mobled37@hanyang.ac.kr

Hyungjo Byun†
School of Electrical and Computer Engineering
University of Seoul
163 Seoulsiripdaero, Dongdaemun-gu, Seoul, Republic of Korea
qusjo8@uos.ac.kr

Hyunjung Shim[*]
Kim Jaechul Graduate School of Artificial Intelligence
KAIST
85 Hoegi-ro, Dongdaemun-gu,Seoul 02455, Republic of Korea
kateshim@kaist.ac.kr

Abstract

Deep learning has achieved significant improvement in cell segmentation of microscopy images in the field of Biology. However, a lack of generalization has been a major bottleneck of segmentation models since the performance is largely degraded with out-of-distribution data or unseen class data. Developing a generalized segmentation model is challenging due to the diversity of modalities, different staining methods, complicated cell shapes, and extremely high image resolution in microscopy images. The dataset for the "Weakly Supervised Cell Segmentation in Multi-modality High-Resolution Microscopy Images" challenge consists of images with these diverse characteristics. To address these challenges, we trained the Cellpose[1] model to competently perform instance segmentation on datasets with various characteristics. For that, we 1) specified the model to only use green and blue channels for all types of cell images, and 2) investigated the effect and performance of the existing diameter estimation model to determine the areas where it performs best, using images of various resolutions. As a result, we achieved an F1 score of 0.7607 for the validation (Tuning) set.

1 Introduction

Deep learning has made remarkable advancements in processing microscopy images of biological samples, including tissue cells and bacteria, in the field of biology[2][3]. Fully supervised cell segmentation models have demonstrated outstanding performance on training data. However, they tend to suffer from significant performance degradation during test time due to overfitting, which is

[*] corresponding author
† Equal contribution

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

caused by memorization[4]. This overfitting issue becomes more significant when the same model is applied to unseen categories of microscopy images. In addition, the performance of such models is insufficient for transfer learning[1]. Therefore, it is important to develop a generalized model that can be applied to a variety of cell types.

Various attempts in cell segmentation has made to create generalized models for microscopic images. However, there is no significant method to achieve generalized peformance. There are two main reasons for this. Firstly, the relationship between input and output is highly complex due to the diversity of cell observations and staining methods[5]. Microscopic modalities, such as bright field, fluorescent, phase-contrast, and differential interference contrast, have diverse image channels and expression formats. Different staining methods can further complicate the data distribution even within the same modality. Also, images are generated in various ways according to cellular morphology, RNA expression, and protein expression, resulting in variations in dye and fluorescence wavelength. Secondly, defining the relationship between input and output is challenging due to the diversity of cell shapes[1]. Even cells of the same type are difficult to separate when tightly packed together or the shape of the colonial morphology varies. Furthermore, differences in image resolution affect the shape of cells' boundaries and the distribution of colonies in images[6]. This diversity of cell shapes makes it challenging for deep-learning models to robustly extract features.

The "Weakly Supervised Cell Segmentation in Multi-modality High-Resolution Microscopy Images" challenge aims to address the challenges of cell image variation and to develop a generalized cell segmentation model. The challenge dataset includes diverse microscopic modalities such as bright field, fluorescent, phase-contrast, and differential interference contrast, with varying dyeing methods, image sizes, and channels. The dataset also contains various types of cells with different shapes and sizes. Furthermore, there is a lack of uniformity in image resolution, with resolutions ranging from 512×512 to 10,000×10,000.

To address the above-mentioned issues, we employed Cellpose[1] as our baseline model. Cellpose is a U-Net-based instance segmentation model[2] that takes a microscopy image as input and produces cell probability and a gradient vector field, which converge to the center of cells for each pixel. We selected Cellpose over other state-of-the-art models because we believed it would better handle the diversity of modality and cell shapes. Cellpose provides a pretrained model for each clustering, where each cluster stands for each modality. Owing to the modality-specific nature, its pretrained model can exhibit significantly better performance for individual modalities. Thus, we hypothesized that considering the modality simultaneously during training would help generalized performance. Additionally, Cellpose is suitable for solving problems caused by various cell shapes for three reasons. 1) Unlike models that only output pixel-specific cell probabilities, Cellpose segments cells using the gradient of flow, making it work well even with minor classification errors. 2) Cellpose does not assume a predetermined polygon, such as star-convex polygons like Stardist[3]. The flexibility of Cellpose leads to effectively handling elongated and complex cell shapes. 3)Cellpose is expected to be resilient to diverse resolutions since it performs resizing based on diameter estimation.

However, naíve adoption of Cellpose is not compatible with handling our dataset, and thus two issues should be addressed in advance. The first issue pertains to the differences in channel definition between the pretraining process and the challenge dataset. Cellpose requires two channels as input: the first channel depicts the cytoplasm, and the second channel shows the cell nucleus. However, there are cells in the challenge dataset that lack nuclei and some cells require dye color to distinguish them, resulting in different numbers of channels. It is challenging to convert images into Cellpose's two-channel format, making it difficult to generalize the model to all images of our Challenge dataset. For that, we filled the second channel with zero for images with only one channel, following Cellpose's suggestion, and utilized any two channels of the original image for images with two or more channels. Furthermore, since many images in the dataset possess green cytoplasm and blue nuclei, we fixed the first channel as green and the second channel as blue, allowing the model to adapt itself to other multi-channel images.

The diversity of resolution poses the second challenge when using Cellpose on the challenge dataset. The dataset contains images with different resolutions, some of which surpass Cellpose's pretrained dataset. Consequently, assuming a fixed cell diameter will lead to varying the accuracy depending on the input resolution[7]. To handle the various resolutions, we estimated the cell size based on the original image and resized it to the cell size used for training. After adopting the Cellpose model to fit the challenge dataset, we trained and tested it on the validation set, finally achieving an F1 score of 0.7607.

This paper will describe our method by first discussing the superiority of Cellpose over other models, followed by explaining the structure and training approach of the model. The experiments section will investigate the characteristics of the data employed in Cellpose pretraining and the data for this Challenge. We will then provide implementation details. In the results section, we will showcase the final outcomes for the qualitative evaluation. Finally, we will analyze limitations and discuss future work.

2 Method

In this section, we will discuss the challenges that arise when applying models other than Cellpose to the challenge dataset. We will then describe the preprocessing steps, followed by a detailed explanation of the Cellpose model and postprocessing techniques. This paper's information related to Cellpose refers to the original Cellpose paper[1].

2.1 Weekness of Existing Methods

Among existing methods, simple approaches like the Watershed algorithm[8] have been employed for cell segmentation. The Watershed algorithm generates a topological map based on the grayscale values with a threshold and separates regions according to pixel intensity. However, the basin method, a variant of the Watershed algorithm, is sensitive to noise and becomes inaccurate when boundaries are unclear due to cell adhesion. Moreover, if cell size and shape change, factors such as thresholds need adjustment[9]. Consequently, it is unsuitable for datasets requiring accurate segmentation of various cell images. Methods like the U-Net[2] model differentiate background, cells, and boundaries. However, these approaches are more prone to predicting pixels as part of the boundaries in images with high cell density and a high boundary ratio. Besides, some pixels within large cell boundaries may be mistakenly predicted as cells, resulting in two or more cells being predicted as a single large cell. Such errors negatively impact performance. Models using star-convex polygons such as Stardist[3] produce more than one star-convex polygon for elongated cells, which results in the mask failing to fill an entire cell[10].

Since Cellpose is a U-Net-based neural network with skip connections and uses a style vector to encode input style, it is more robust to noise and diverse cellular morphologies than Watershed methods. In addition to distinguishing cells and backgrounds per pixel, Cellpose predicts the gradient of the flow towards the center of the cell, providing more information compared to the inference method used in U-Net. In contrast to Stardist, Cellpose works well for elongated cells, as it does not assume the center and predict the distance to the outer boundary of the cell, but predicts the point where the flow gradient of each pixel converges as the center of the cell. Therefore, we chose the Cellpose model for the cell segmentation task of the challenge.

2.2 Preprocessing

Since the Cellpose model predicts the gradient of the flow, the mask of the original label must be converted to the gradient of the flow. The image obtained by converting the original mask image according to the output format of Cellpose is a vector field. The vector field points to the center of the cell. However, if the cell's convexity is low, it does not necessarily point directly to the center of the cell. Thus, it is calculated to eventually converge to the center of the cell when translating between pixels along the vector field. Following the vector field inside the same cell converging at one point, it can be said that pixels can be included in the same cell if output vector fields converge at the same point.

The method to obtain the vector field from the mask uses the heat diffusion method. First, the midpoint of the vertical and horizontal boundaries of the cell is defined as the center point. Besides, if the central point is outside the cell, the nearest point inside the cell is defined as the central point. This central point is regarded as the source of heat, and the pixel value of the source is increased by 1 for each iteration. At the same time, each pixel inside the cell is updated to the average value of the 3×3 pixel region, including itself. However, the outside of the cell is fixed to zero. The number of iterations is determined to be twice the vertical and horizontal ranges to ensure that the heat is sufficiently diffused. The gradient of the energy distribution generated in this way is calculated to generate the final label to be used for training.

Meanwhile, in the input image, 1 to 99 percentiles of pixel intensity were converted from 0 to 1. In addition, images with more than three channels are discarded, leaving only two designated channels. The second channel is filled with zero for the grayscale image. More details about the channel will be explained in 2.3.2.

2.3 Cellpose

2.3.1 Model Structure and Loss Function

The basic structure of Cellpose is based on U-Net as seen in Figure 1. Downsampling and upsampling passes consist of four spatial scales. Each spatial scale consists of two residual blocks which then consist of two 3×3 convolution layers. In other words, one spatial scale contains four convolution layers. The output of each convolution layer performs batch norm and ReLU operations. Not only are there skip connections in the residual block, but also from downsampling to upsampling passes in the same spatial scale. The operations performed at each layer in the downsampling pass can be expressed as shown in Equation 1.

$$\begin{aligned} &\mathbf{x'}_t = D_{2\times 2}\left(\mathbf{x}_{t-1}\right) \\ &\mathbf{x^*}_t = F\left(F\left(\mathbf{x'}_t\right)\right) + P_{1\times 1}\left(\mathbf{x'}_t\right) \\ &\mathbf{x}_t = F\left(F\left(\mathbf{x^*}_t\right)\right) + \mathbf{x^*}_t\,, \end{aligned} \tag{1}$$

$D_{2\times 2}$is a downsampling operation and F is a sequential operation of convolution, batch norm, and ReLU. Note that each F has different parameters and $P_{1\times 1}$is a 1×1 convolution.

A style vector is a vector encoding a style of an input image. Each dimension of the style vector is the global average pooling per channel of the feature map, which is the last output of the downsampling pass. The feature map has 256 channels, resulting in a 256-dimensional style vector. Since the density of cells may vary in each input, the style vector is normalized. The style vector is then used as input to the residual block of each spatial scale of the upsampling pass.

The structure of the upsampling pass is the opposite of the downsampling pass in terms of the order of the spatial scale, but the rest is the same. In addition, skip connections and a style vector are added during the upsampling pass. The skip connection between the downsampling and the upsampling passes is connected by adding the feature map of the same spatial scale from the downsampling pass to the second convolution output of the corresponding upsampling pass. The style vector is added to the remaining three outputs of each residual block, except for the first of the four convolution outputs. Mathematically, this process can be expressed as shown in Equation 2.

$$\begin{aligned} &\mathbf{z'}_t = U_{2\times 2}\left(\mathbf{z}_{t+1}\right) \\ &\mathbf{z^*}_t = G\left(\mathbf{s}^*, F\left(\mathbf{z'}_t\right) + \mathbf{x_t}\right) + P_{1\times 1}\left(\mathbf{z'}_t\right) \\ &\mathbf{z}_t = G\left(\mathbf{s}^*, G\left(\mathbf{s}^*, \mathbf{z^*}_t\right)\right) + \mathbf{z^*}_t\,, \end{aligned} \tag{2}$$

$U_{2\times 2}$ is upsampling operation and G is operation that add style vector s^* to feature map.

The input shape of Cellpose is a pre-processed 2-channel microscopy image, while the output consists of the probability of the cell, as well as horizontal and vertical flow field for each pixel. Consequently, the final output of the convolution during the upsampling pass is a 3-channel map by a 3-channel 1×1 convolution operation. The horizontal and vertical flow fields are subjected to L2 loss, as they are real numbers, whereas the probability of a cell undergoes cross-entropy loss L_{CE} following a sigmoid operation σ. Thus, the final loss function L_{final} can be expressed as Equation 3.

$$L_{final} = \left\|\mathbf{y_0} - 5\mathbf{H}\right\|^2 + \left\|\mathbf{y_1} - 5\mathbf{V}\right\|^2 + L_{CE}\left(\sigma\left(\mathbf{y_2}\right), \mathbf{P}\right) \tag{3}$$

Where $\mathbf{y_0}$,$\mathbf{y_1}$,$\mathbf{y_3}$ means horizontal, vertical gradient and probability of cell. $\mathbf{H}$,$\mathbf{V}$,$\mathbf{P}$ is ground truth horizontal, vertical gradient, and probability of cell.

2.3.2 Adapting Cellpose to Challenge Dataset

Adapting channel The given dataset consists of images obtained with various microscopy modalities. Therefore, the number of channels, file size, and file format of each modality are different.

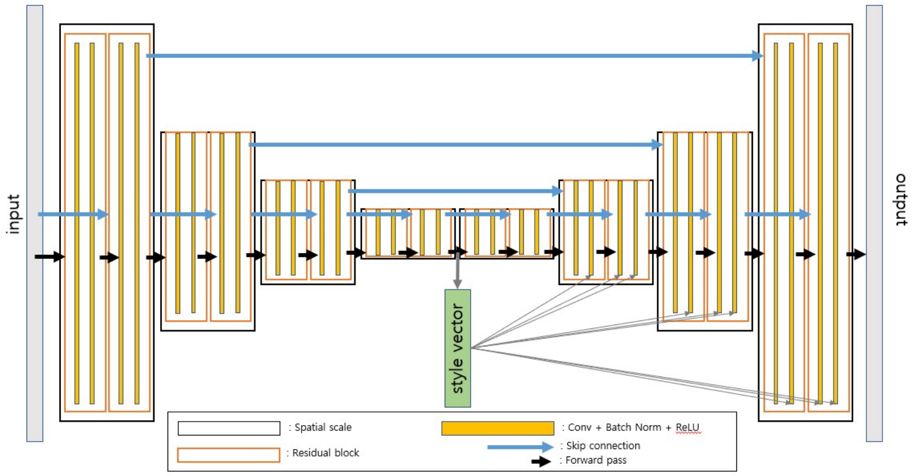

Figure 1: U-Net model used in Cellpose.

However, if the channel arbitrarily changed or modify the file format is modified, the pretrained Cellpose model may not be able to provide accurate inference. For this reason, we applied processing the image in the original source code of Cellpose. The Cellpose model we used is a cytoplasm model trained with a 2-channel image. The cytoplasm model was pretrained in the image, where the first channel is the channel to segment and the second channel is the optional channel representing the cell nucleus. However, in order to take advantage of the important colors of images in the dataset, such as dye, we used these images without changing them to gray scales. Therefore, like the original Cellpose code, the grayscale image was made into a two-channel image with the second channel as zero, and the image with three or more channels was made into a 2-channel image maintaining only two designated channels. In the given dataset, we used green for the first channel and blue for the second channel because there are many green-stained cell images with blue-stained cell nuclei.

Size estimation The given dataset comprises various resolutions, leading to differences in the model's receptive field even if the modality and cell type are identical, which may decrease accuracy. Cellpose addresses this issue with a size estimation model. During testing, the downsampling pass of U-Net outputs a style vector derived from the input image. With a given style vector, the size estimation model predicts the diameter of the cells present in the image. Subsequently, the ratio of the estimated diameter to the diameter used during training is computed to resize the entire image. We retrained the size estimation model without using the pretrained Cellpose model to deal with the challenge dataset's various resolutions. The size estimation model is developed using a style vector generated by the training set after the U-Net model has been trained completely. The size estimation model is not an additional neural net, but a matrix A that satisfies equation 4.

$$\begin{gathered}(XX^T + L)A = XY \\ where\, X = (S - \mu_S)^T\ ,\, Y = (D - \mu_D)\end{gathered} \tag{4}$$

S denotes the style matrix, composed of style vectors, while D represents the diameter matrix. μ_S and μ_D correspond to the style vector and diameter, respectively. In the equation $L = rI$, the constant r serves as a regularizer term, and I is an identity matrix with the same shape as S. In summary, the size estimation model matrix A estimates the diameter through a weighted sum of each dimension of the style vectors.

2.4 Post-processing

The output inferred by the model from the input image is a three-channel map that has the same size as the input image. Each channel represents a cell probability, a horizontal, and a vertical gradient. Cellpose uses this information to create a mask. It finds the center of the cell through gradient tracking, and pixels with a cell probability of 0.5 or more are in the range of transfer along the gradient. When moving from pixels predicted to be cells to the neighboring pixels along the gradient, the model predicts the converging point as the center of the cell and predicts that pixels converging to the same center belong to the same cell. Finally, a mask image is generated by allowing pixels belonging to the same cell to be included in one mask.

3 Experiments

3.1 Dataset

Cellpose was trained with various images. The author of Cellpose mainly composed datasets with images obtained on the internet by searching keywords such as ’cytoplasm’, ’cellular microscopy’, and ’fluorescent cells’. The dataset consists of 361 images of fluorescently labeled proteins on two DAPI-stained channels, 50 brightfield microscopy images, 58 membrane-labeled cells, 86 other microscopy images, and 98 non-microscopy images.

The given dataset consists of 300 brightfield, 300 fluorescent, 200 phase-contrast, and 200 differential interference contrast patches. The cell types consist of red and white blood cells, plasma cells, hanseniaspora, animal tissue cells, etc.

3.2 Implementation details

3.2.1 Environment settings

The development environments and requirements are presented in Table 1.

Table 1: Development environments and requirements.

System	Ubuntu 22.04.1 LTS
CPU	AMD Ryzen 7 5800X 8-Core Processor CPU@4.85GHz
RAM	32GB; 2933 MT/s
GPU (number and type)	NVIDIA 1080Ti
CUDA version	11.4
Programming language	Python 3.9
Deep learning framework	Pytorch (Torch 1.12.1, torchvision 0.2.2)
Specific dependencies	install Cellpose
Code	`https://github.com/mobled37/cvml_omnipose`

3.2.2 Training protocols

Data augmentation During training time, we flipped images with random rotations and resized them to 224x224 size in order to train the U-Net model. To create a size estimation model, each image was passed through U-Net to create a style vector after being randomly rotated, flipped, and resized to 512x512.

The number of sampled style vectors to create a style matrix S is 10 times the size of the entire dataset. After sampling the style vectors, we can find A from S with linear regression. On the other hand, at test time, the image was divided into 224x224 tiles and inferred. In this case, the ratio of overlapping tiles is 10%.

Hyper parameter settings First, we initialize the network with "LeCun" uniform initialization. We trained the Cellpose model 300 epochs using SGD optimizer with Nesterov momentum (μ = 0.9) and weight decay($\beta = 1e - 5$). The learning rate warmed up during the first 10 epochs from 0 to 0.001. After epoch 210, the learning rate was halved for every 10 epochs. The batch size was 8, and we resized images to 224×224, so the patch size was 8×224×224. More details about training protocols are presented in Table 2.

Table 2: Training protocols

Network initialization	"LeCun" Uniform Initialization[11]
Batch size	8
Patch size	8×224×224
Total epochs	300
Optimizer	SGD with nesterov momentum ($\mu = 0.9$) and weight decay($\beta = 1e - 5$)
Initial learning rate (lr)	0.001
Lr decay schedule	epoch<10: start 0 and annealed linearly to 0.001, epoch>210: halved for every 10 epochs
Training time	150min
Loss function	same as equation 3
Number of model parameters	6.6M
Number of flops	15.7 G

Table 3: Quantitative Results on tuning set

Method	F1 Score
Baseline	0.3112
Baseline + Fine-Tuning + Adaptive Segmentation Channel	0.6387
Baseline + Fine-Tuning + Adaptive Segmentation Channel + Diameter Estimation	**0.7607**

4 Results and discussion

4.1 Quantitative results on tuning set

In Table. 3, experiments were conducted in a supervised setting without using any unlabeled data. We used Cellpose as the backbone and chose the cyto2 model as a baseline. The baseline model did not show good performance on the challenge dataset. As mentioned in the Method, we thought the reason was that the model failed to adapt to the diversity of the challenge dataset. In particular, the baseline model requires fine-tuning on the challenge dataset, which has no special meaning for the channel, since the baseline model has been trained to operate on images of each channel representing cells and nuclei. Therefore, we fine-tuned the baseline model to the challenge dataset. At this time, the channel was designated to use green and blue, taking advantage of the fact that the dataset contains many images of green cells with blue nuclei. Images with three or more channels not containing green cells and blue nuclei were also using green and blue channels regardless of the definition of the channel. On grayscale images, we designated a model to use a single gray channel and a channel representing nuclei which were filled with zeros. These processes are considered Adaptive Segmentation Channel selection according to the image channel. More details about fine-tuning are mentioned in table 3.2.2. As a result of fine-tuning, the F1 score increased by 0.3275 compared to that of the baseline, resulting in an F1 score of 0.6387. Finally, our proposed method combined with the above methods and diameter estimation on each image. Consequently, the F1 score was 0.7607, which was 0.4495 higher compared to that of the baseline, and 0.1220 higher than that of the Fine-Tuning + Adaptive Segmentation Channel method.

Effectiveness of flow threshold To investigate the difference in the recognition rate of colonial morphology according to the flow threshold value, we conducted an experiment with the flow threshold set to 0 and 0.4. We observed that the colonial morphology was recognized to some extent in the cytoplasm pretrained model, but the image of each cell was not accurately captured. However, when the flow threshold was set to 0.4, the overall recognition rate decreased, whereas it increased when the flow threshold was 0. Compared to the case where the flow threshold was set to 0, many recognized cells disappeared when the flow threshold was set to 0.4. This suggests that the model made uncertain inferences and deviated from the actual cell morphology. Contrary to the expectation that performance would improve due to a higher recognition rate, Table. 4 revealed a performance drop of 0.0523 from 0.7607 to 0.7084 when the flow threshold was set to 0.

4.2 Qualitative results on tuning set

Effectiveness of Our Proposed Method Fig. 2 shows the difference between the results of the cyto-pretrained model and the proposed method. In the case of columns 1 and 2, it was observed that the proposed method was color-trained. In the case of column 3, as shown in (b), the pretrained cytoplasm model was not trained on case 35 at all, but our proposed method was trained on non-circular shapes

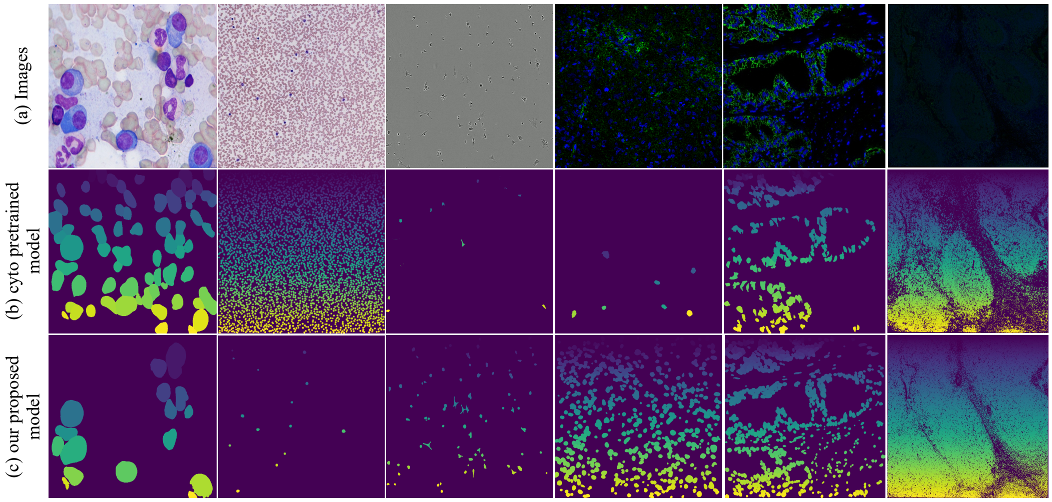

Figure 2: Qualitative results of the proposed method. (a) Input images, case 1, 4, 35, 49, 67, 101 from the left (b) Inference results on cytoplasm pretrained model. (c) Inference results of our proposed method from the left

as shown in (c). The result of column 3 demonstrates that our proposed method is robust for diverse shapes. Column 4 appears not to be recognized in (b), whereas in (c), the segmentation channel is recognized, and the inference is well established. In the case of columns 5 and 6, we observed that the recognition rate of our proposed method increased compared to the cytoplasm pretrained model.

Effectiveness of Diameter Estimation Fig. 3 shows the difference in the inference performance according to the diameter estimation. It has an image size of 640×480 for the images in column 1, and an image size of 2560×1920 for images in column 2. In the absence of the diameter estimation for large image sizes, performance significantly decreases. The results in column 2 (b) demonstrate that cells are not recognized during inference and the recognition rate increases when the diameter estimation is performed. The model estimated diameters of 58.32 and 125.02 for columns 1 and 2, indicating that performance deteriorates when there is a large difference in mean diameter value.

In examining column 4 (b), where the cell is not recognized at all, it is expected that the difference between the estimated average diameter and the default setting will be large. The predicted mean diameter values in columns 3 and 4 are 52.52 and 52.30, which are greater than 30. However, although the average diameter of column 1 is higher than 30, the inference was successful. From this, it can be inferred that when the image sizes of columns 3 and 4 are adjusted through preprocessing, the estimated diameter value can change, resulting in a significant difference from the mean diameter value of 30. Therefore, we expected that when the image sizes vary, not fixing the mean diameter value to a certain constant and performing diameter estimation separately for each image could lead to an increase in performance.

On row 3, diameter estimation was performed through these qualitative results. It was observed that the recognition rate of cells increased for various image sizes. Our proposed method performed fine-tuning, Adaptive Segmentation Channel, and diameter estimation, and the F1 score at this time were 0.7607. On the other hand, when the diameter estimation was not performed, the F1 score was 0.6387, indicating a performance difference of 0.1220.

4.3 Limitation and future work

Limitation

The identified issue is that the data distribution of labeled bacilli training images is very small, accounting for only 0.3% of the provided dataset. In particular, this phenomenon was observed for images with a bacillus shape as in cases 74, 77, and 78. This suggests that the training may not be effective for this particular image style. Furthermore, the bacillus shape is recognized as uncertain due to its difference from the cell shapes that make up a significant portion of the dataset distribution. Moreover, since there are no images for these cases in the unlabeled data, even if the unlabeled data

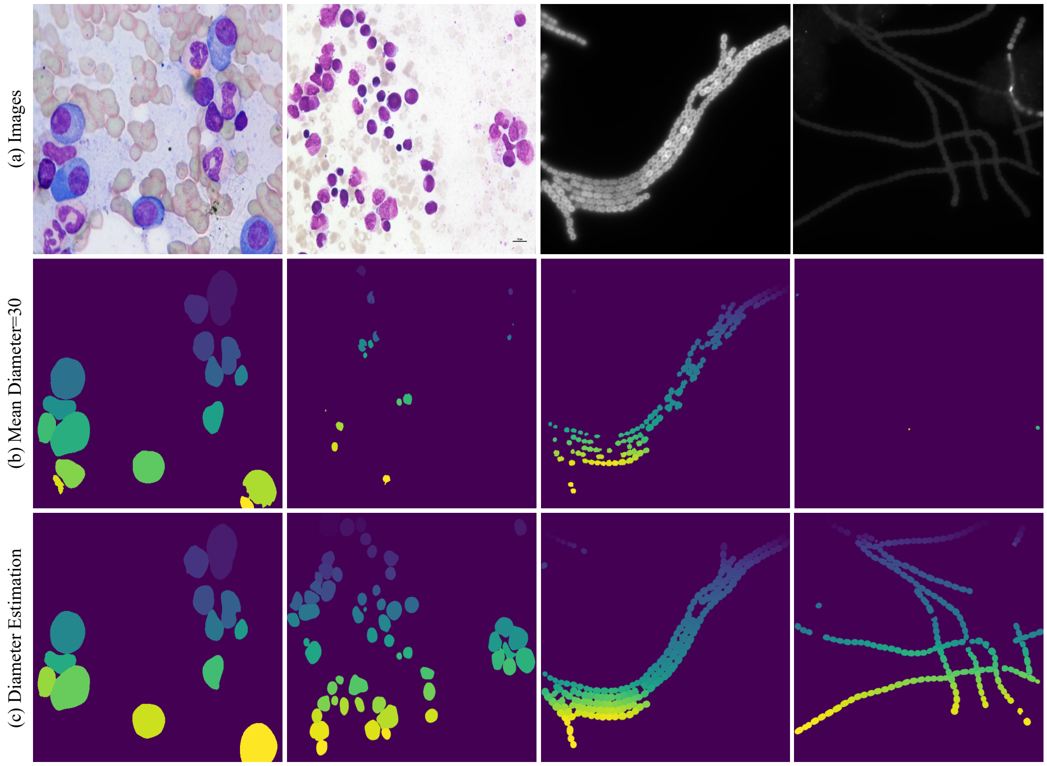

Figure 3: Qualitative results for the effects of diameter estimation. (a) Input images. From the left to the interference results in cases 1, 2, 72, 73 (b) Mean Diameter=30. (c) Diameter estimation for each image. Proposed method setting.

is used, the recognition rate for the corresponding image type might not improve. Furthermore, the Cellpose backbone uploads all images to RAM during preprocessing; however, due to capacity issues in the preprocessing stage when using unlabeled data, we were unable to conduct experiments with our computer specifications.

Future work We will apply semi-supervised learning through unlabeled data using consistency regularization, which maintains prediction consistency by giving perturbation to unlabeled data. Additionally, we aim to address the capacity issue in Cellpose's preprocessing process and conduct experiments using unlabeled data. Moreover, by optimizing the flow threshold value, we intend to develop a model that predicts cell shapes more closely resembling the actual cell morphology, with the goal of improving recognition rates for the relevant cases 74, 77 and 78.

5 Conclusion

Our proposed method was optimized by various hyperparameter tunings provided by Cellpose. Specifically, by applying diameter estimation to each cell image, we discovered performance improvements through flow threshold tuning. Our experimental results demonstrate that our method achieves higher accuracy for the provided data compared to the cyto pretrained model. For the validation(tuning) set, the F1 score of the cyto-pretrained model was 0.3112, and our proposed method showed an F1 score of 0.7607, which improved the performance by 0.4495.

References

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

[2] 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.

[3] Uwe Schmidt, Martin Weigert, Coleman Broaddus, and Gene Myers. Cell detection with star-convex polygons. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 265–273. Springer, 2018.

[4] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[5] Juan C Caicedo, Allen Goodman, Kyle W Karhohs, Beth A Cimini, Jeanelle Ackerman, Marzieh Haghighi, CherKeng Heng, Tim Becker, Minh Doan, Claire McQuin, et al. Nucleus segmentation across imaging experiments: the 2018 data science bowl. *Nature methods*, 16(12):1247–1253, 2019.

[6] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[7] Libo Wang, Ce Zhang, Rui Li, Chenxi Duan, Xiaoliang Meng, and Peter M Atkinson. Scale-aware neural network for semantic segmentation of multi-resolution remote sensing images. *Remote sensing*, 13(24):5015, 2021.

[8] Gang Li, Tianming Liu, J Nie, L Guo, J Chen, J Zhu, Weiming Xia, A Mara, S Holley, and STC Wong. Segmentation of touching cell nuclei using gradient flow tracking. *Journal of Microscopy*, 231(1):47–58, 2008.

[9] Michael Y Lee, Jacob S Bedia, Salil S Bhate, Graham L Barlow, Darci Phillips, Wendy J Fantl, Garry P Nolan, and Christian M Schürch. Cellseg: a robust, pre-trained nucleus segmentation and pixel quantification software for highly multiplexed fluorescence images. *BMC bioinformatics*, 23(1):1–17, 2022.

[10] Kevin J Cutler, Carsen Stringer, Teresa W Lo, Luca Rappez, Nicholas Stroustrup, S Brook Peterson, Paul A Wiggins, and Joseph D Mougous. Omnipose: a high-precision morphology-independent solution for bacterial cell segmentation. *Nature Methods*, pages 1–11, 2022.

[11] Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer, 2012.

A Discussion

Table 4: Quantitative Results on tuning set

Diameter Estimation	Flow Threshold=0	Merge Channel	Style Vector=10	Resize=(512,512)	F1 Score
					0.6387
✓					**0.7607**
✓	✓				0.7084
✓		✓			0.7336
✓			✓		0.6805
✓				✓	0.6518

As shown in Table. 4, when the diameter estimation was set to 30, the performance degraded by 0.1220 compared to the no-diameter estimation setting, which had an F1 score of 0.7607. Consequently, subsequent experiments were conducted with diameter estimation applied by default. Setting the flow threshold to 0 increased the cell recognition rate, but the model's inferences differed from the actual cell morphology.

We also conducted an experiment involving merging channels. Merging channels involved adding the pixel values of the remaining channels, excluding the green and blue channels already chosen, to the green and blue channels and dividing by the combined number of channels. This was mathematically expressed in Equation 5.

$$\begin{gathered} I_k = \tfrac{1}{N_C - 1}\left(I_k + \textstyle\sum_{t \in C'} I_t\right) \\ where\ k \in \{g, b\}, C^{'} = C - \{g, b\} \end{gathered} \tag{5}$$

In this case, I_k represents the pixel value of channel k, and g and b correspond to the green and blue channels in an image with RGB channels. C is a set of channels in the image and N_C is the number of channels. When an image had only one channel, instead of filling the second channel with zero, the image was copied into two and assigned one as the second channel. The merging channel experiment resulted in reduced performance compared to the method that discarded the unselected channel, with an F1 score of 0.7336.

In the cytoplasm pretrained model, the style vector was fixed to 1 and training proceeded accordingly. When directly observing the dataset, the data could be classified into 10 types, so we set the style vector to 10 before training. In this case, performance was significantly reduced, likely due to having too many clusters to train. Finally, an experiment was performed to resize the training image to 512×512. As a result, the F1 score decreased by 0.1089 to 0.6518 compared to the existing 224×224 size with an F1 score of 0.7607.