Cell Segmentation of Bright field Microscope Images

Potchara Boonrat
Stanford University
potchara@stanford.edu

Abstract

Bright field microscope has enabled us to study the dynamics of cells and their morphological changes over time. While various methods for cell segmentation have been developed to extract single cell data from biological images, most of them have been created for multiplexed images, which has multiple channels with homogenous background and can simply be segmented using nuclear and cytoplasm channels. Bright field microscope images can have different gradients of background from images to images, making it difficult to perform cell segmentation. In this study, I applied the method used for cell segmentation of multiplexed-ion beam imaging (MIBI) data to bright field images. I used PanopticNet, a convolutional network architecture, to segment cells in bright field image data. I trained the model on the DeepBacs dataset, which contains bright field images of bacteria. The final model was trained using softmax loss, Adam optimizer, and learning rate of 0.0001.

1. Introduction

A bright field microscope is a crucial tool for studying microorganisms and diagnosing diseases. In a clinical setting, physicians use it to detect infections or abnormalities in blood and tissue biopsies. The pathology of cells and tissues can be investigated through examining morphological alterations of specific cell types. In basic science research, the most common use of a microscope is for cell counting and studying cell morphology. Cell culture is fundamental to molecular and cellular biology research, and most experiments involve examining cell dynamics by counting the number of cells at different time intervals or conditions. Manual cell enumeration can be time-consuming and prone to error in large datasets. In addition, the development of novel microscopes has expanded microscope ability to include live-cell imaging and fluorescent microscopes for measuring protein expression. While these instruments are built on bright field microscope and can still function in that capacity, they can be used to identify unique protein expressions in individual cells and monitor cell migration at various time points.

Due to this technological advancement, the amount of data generated in biological imaging studies is continuously growing, which can become challenging for classical data analysis and interpretation, requiring more complex computational approaches to extract relevant features from images. Single-cell analysis is critical for studying how cells respond to different stimuli. Yet, it is still difficult to extract protein expression pattern in tissues or inside the cells under a microscope. Cell segmentation is a crucial tool to achieve single-cell level analysis from biological images. Several cell segmentation algorithms have been developed for multiplexed imaging equipment, including fluorescence microscopes and MIBIs [1, 2]. These tools generate TIFF files with numerous channels, each containing expression data for a single protein. The nuclear channel defines the nucleus, while the cytoplasm channel identifies the boundaries of each cell. Because every cell contains nucleus and cytoplasm, these two channels typically have minimal brightness gradients and are useful for delineating boundaries. Most current methods rely on non-trainable algorithms or manual segmentation by expert pathologists. The algorithms generally require defining a threshold to discern between an image's background and foreground. Because of the brightness gradients, a single thresholding does not work well in bright field microscope images.

In this study, I applied a tool developed and used in my lab, Mesmer, which is built on a PanopticNet architecture, to segment bright field microscope images. To explore this idea, I used DeepBacs dataset [3], which includes images of 2 species of most common bacteria, *Escherichia coli* (E. coli) and *Bacillus subtilis* (B. subtilis), for training and testing.

2. Related Work

Before deep learning models, there are several automated segmentation methods to segment cells on biological images. Most of them rely on thresholding [4, 5, 6], where a single threshold is selected, and a pixel is classified as foreground if it has higher values and background if it has lower values. One of the most prominent methods was Otsu's method [4]. The threshold for this method is chosen based on a value that minimizes the intra-class variance of background and foreground pixels. However, there is still limitation to this method when it comes to small objects or images with high noise. Moreover, images can have brightness gradients, which cannot be solved with a single threshold. A new algorithm developed to deal with this problem, adaptive thresholding, decides whether a pixel is in foreground or background by calculating average pixel intensity around a pixel and measure if the pixel is higher or lower that the average intensity [5]. This has improved the ability of segmentation method to perform images with brightness variability.

Another class of segmentation algorithm was developed using active contours, such as Snake active contour algorithm [7]. This method involves using energy minimization to fit splines to the contour objects in images and requires user to move splines out of local minima. This model was able to perform well on image with high noises. The watershed algorithm [8], which is used as a post-processing step in my study, uses image as a topological relief map and has an intuitive explanation. Every image is flooded at the local minima, and the boundaries of segmented images are determined by the points where the water hits. While it works well in biological images, the standard watershed approach can result in over segmentation and incorrect edge segmentation.

With the introduction of convoluted neural network (CNN) [9, 10], we can train a model on multidimensional data efficiently. Multiple models have been developed for the task of cell segmentation. In 2012, deep neural network (DNN) [11] was developed for transmission electron microscopy (TEM) image segmentation and composed of successive layers of convolutional, max-pooling and fully connected layers. The major difference between DNN and other early CNN models is that it uses max-pooling layers, which is non-trainable, instead of subsampling layers. For each pixel, DNN determines whether it falls into the membrane or non-membrane category by estimating the probability of a pixel *p* being classified as a membrane and non-membrane, based on the raw intensity values within a square window centered on *p*, where the width of the window is w pixels. When a pixel is near the edge, the window extends beyond the limits and the out-of-bounds pixels are created by mirroring the nearby pixels from the image. The model was trained using a patch around classifying pixel as an input. It could efficiently localize pixels on TEM images. Yet, there are still drawbacks of this method. The model runs very slowly because there are multiple patches for training and there is redundancy between overlapping patches. Moreover, larger patches can reduce the localization accuracy because of the usage of more max-pooling layers, whereas small patches do give only little context of images.

Recently, our lab has created a deep learning model, Mesmer, for cell segmentation of MIBI images [1]. Mesmer is a CNN-based model built upon PanopticNet architecture, which consists of Resnet50 backbone with a Feature Pyramid Network and four prediction heads [12]. The model receives 2-dimensional TIFF image as an input, where the first dimension is nuclear channel and the second is cytoplasm channel. It produces centroid and boundary predictions as output, which is then used as an input in watershed algorithm to create masks. Mesmer outperforms other cell segmentation algorithms in the same category. Therefore, it would be beneficial if we could use a PanopticNet model to segment bright field microscope images.

3. Methods

PanopticNet Architecture. A PanopticNet is a deep learning model based on Feature Pyramid Network (FPN) connected to a CNN backbone, such as ResNet, DenseNet, or EfficientNet (Figure 1) [1]. In this study, the backbone was constructed from an EfficientNetV2L backbone connected to a feature pyramid network. I used backbone layers C1-C5 and pyramid layers P1-P7. The model receives images that were concatenated with a coordinate map as input and compute 3 transform during training: inner distance transform, outer distance transform, and foreground-background transform. Three semantic heads are used for model training. Inner distance captures the distance between each pixel to cell's centroid. We can compute inner distance transform where the distance of the pixel and its centroid is r as: transform = $\frac{1}{1+abr}$, where a = $\frac{1}{\sqrt{cell\ area}}$ and b is a hyperparameter that is normally set to 1. The outer distance transform is the Euclidean distance of the image. The foreground-background transform predicts whether a pixel belongs to the foreground or background. During testing, only inner distance and outer distance transforms are used for predictions. The mean squared error (MSE) is used for inner distance and outer distance transforms. Softmax loss is used for the foreground-background transform. The softmax loss is scaled by 0.01 to stabilize its value. These transforms are used as an input for watershed algorithms to generate cell segmentation masks. The model was trained using the Adam optimizer with a learning rate of 10^{-4}, a clipnorm of 10^{-3}, and a batch size of 8 images. Training was performed for 32 epochs. After each epoch, the learning rate was reduced using the function lr = lr × 0.99^{epoch}.

Feature Pyramid Network. The feature pyramid consists of bottom-up pathway, top-down pathway, and lateral connections (Figure 2) [12]. The algorithm takes an image of arbitrary size as input and outputs proportionally sized feature maps. The bottom-up pathway is a feed forward computation of the backbone used for feature extraction. The pathway consists of multiple convolutional modules, where each module has many convolutional layers. The bottom-up pathway generates a feature hierarchy consisting of feature maps at various scales with a scaling step of 2. The semantic values of data increases as

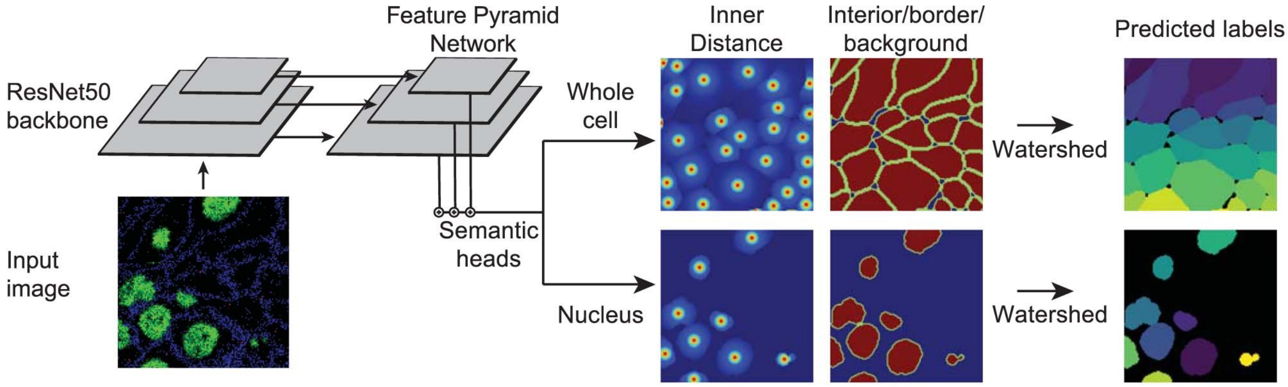

Figure 1. Panoptic architecture. Images are fed into the ResNet50 backbone (EfficientNetV2L in my case) connected with a feature pyramid network. Semantic heads produce inner and outer distance transforms. These values are then fed into the watershed algorithm.

data flows from bottom to top as the model can detect high-level structure, whereas the resolution decreases due to successive convolutional layers. In contrast, the top-down pathway is used to create higher resolution features from semantic rich layers by upsampling coarser resolution feature maps from higher pyramid levels that are semantically richer but spatially coarser. These features are connected to the output of the bottom-down pathway through lateral connections. Each lateral connection integrates features of the same spatial size from top-down pathway and bottom-up pathway by element-wise addition iteratively until the finest resolution map is generated.

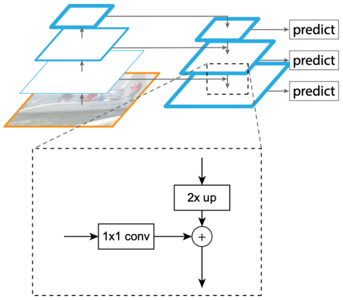

Figure 2. Feature Pyramid Network (FPN). A top-down architecture and bottom-up architecture is connected through lateral connections.

EfficientNetV2L. EfficientNet2VL is an improved version of EfficientNet [13, 14]. EfficientNet is a class of models that uses training-aware neural architecture search (NAS) to search for the baseline model EfficientNet-B0. The model has a trade-off between accuracy and FLOPs. There are 3 major problems with the first version of EfficientNet. (1) Large datasets training is very slow; (2) early layer depthwise convolutions are slow. EfficientNet uses MBConv, which is slower than a recently published Fused-MBConv (Figure X); (3) scaling up every level equally is inefficient. EfficientNet equally scales up all layers using a compound scaling rule. To address these issues, EfficientNetV2 uses both MBConv and Fused-MBConv in the early layers, has smaller expansion ration to reduce the memory usage, and uses smaller 3x3 kernel sizes with more layers to compensate for the reduced receptive field. They also removed the last stride-1 stage in the original EfficientNet. EfficientNetV2L is a scaled-up version of EfficientNetV2 using compound scaling.

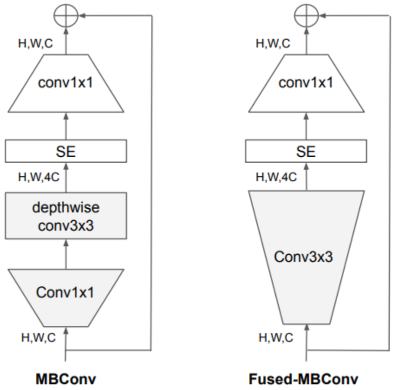

Figure 3. MBConv and Fused-MBConv structures

Watershed Algorithm. We obtain inner distance transform and outer distance transform predictions from the model. These values are on continuous scales. To create a discrete label from these values, I used them as an input for a Marker-based watershed algorithm [15]. I first applied a peak-

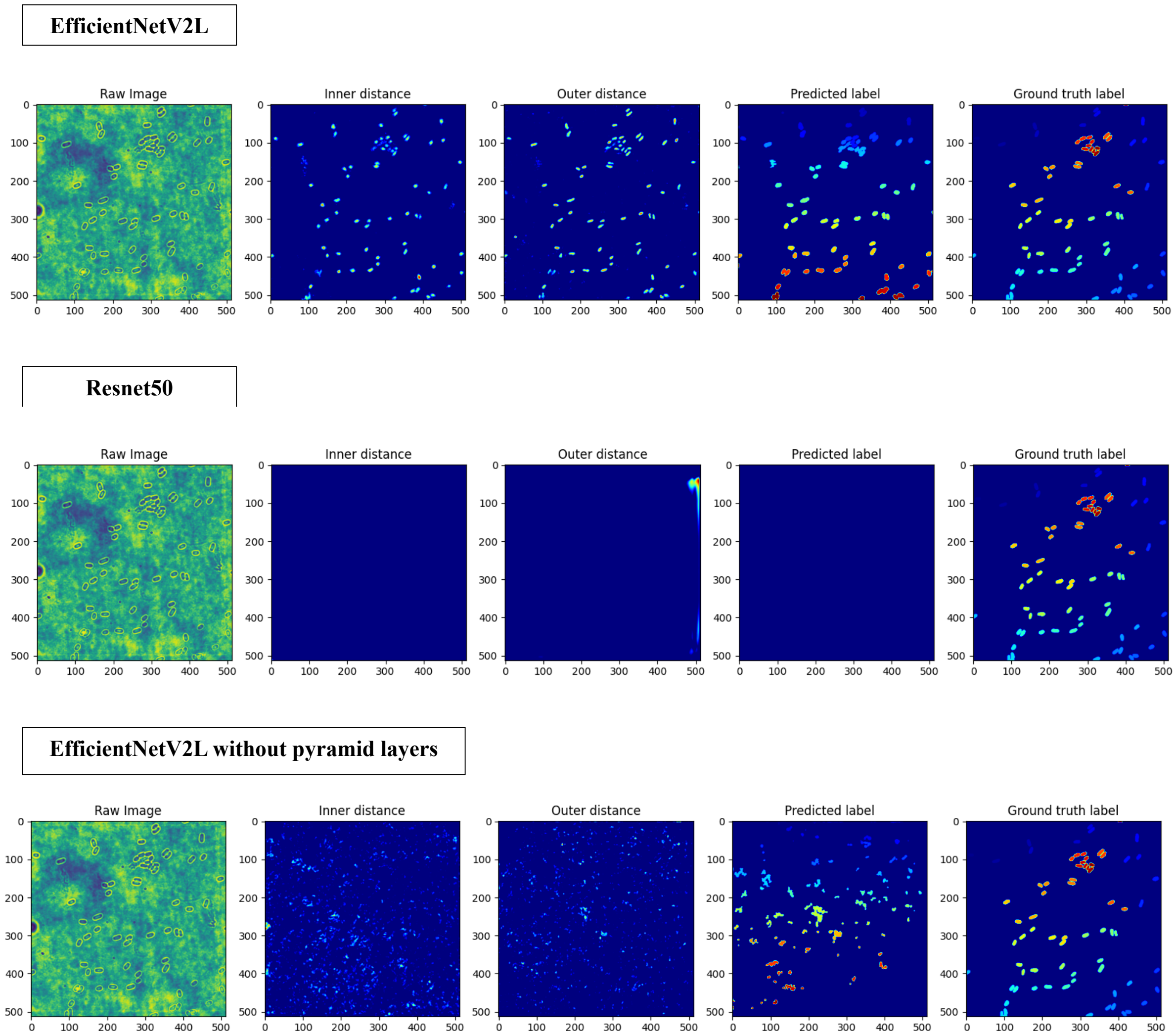

Figure 4. Representative Predicted Images. Two predictions are generated from raw images: Inner distance and Outer distance. The representative images here are from a sample that EfficientNetV2L were able to correctly predict all cells in the image. The first row shows images from EfficientNetV2L; The second row shows images from ResNet50 backbone. The third row shows images from EfficientNet2VL backbone without pyramid layers.

finding algorithm to the inner distance transform and outer distance transform predictions to define the centroid of each cell in the image. These values are thresholded at value 0.01 and 0.025, respectively. The cell centroid and boundary are used as inputs to the watershed algorithm to generate the label image.

Segmentation Evaluation. I evaluated the model on a per-cell basis instead of per-pixel basis [16]. The cost matrix is built between cells in ground truth and prediction. I used the intersection over union (IoU) as the cost for each pair of cells. IoU is calculated as:

$$IoU(x,y) = \frac{x\ \cup\ y}{x\ \cap y} = \frac{x \cap y}{|x| + |y| - |x \cap y|}$$

The cost matrix then underwent a linear assignment process, with a cost of 0.4 assigned to unassigned cells, to determine the cells that match the ground truth labels. For other remaining cells, the graph in which an edge connects between ground truth and predicted cell if the IoU is more than zero was constructed. I classified the error type for each subgraph based on the connectivity of the graph. Node without edges were classified into either a false positive if the graph had only predicted cell, or a false negative if the graph only contained ground truth cells. A merge error was defined as a predicted node that is connected to multiple ground truth nodes. A split error, on the contrary, was defined as a ground truth label node that is connected to multiple predicted nodes. Lastly, catastrophe is any subgraphs that have multiple ground truth or predicted

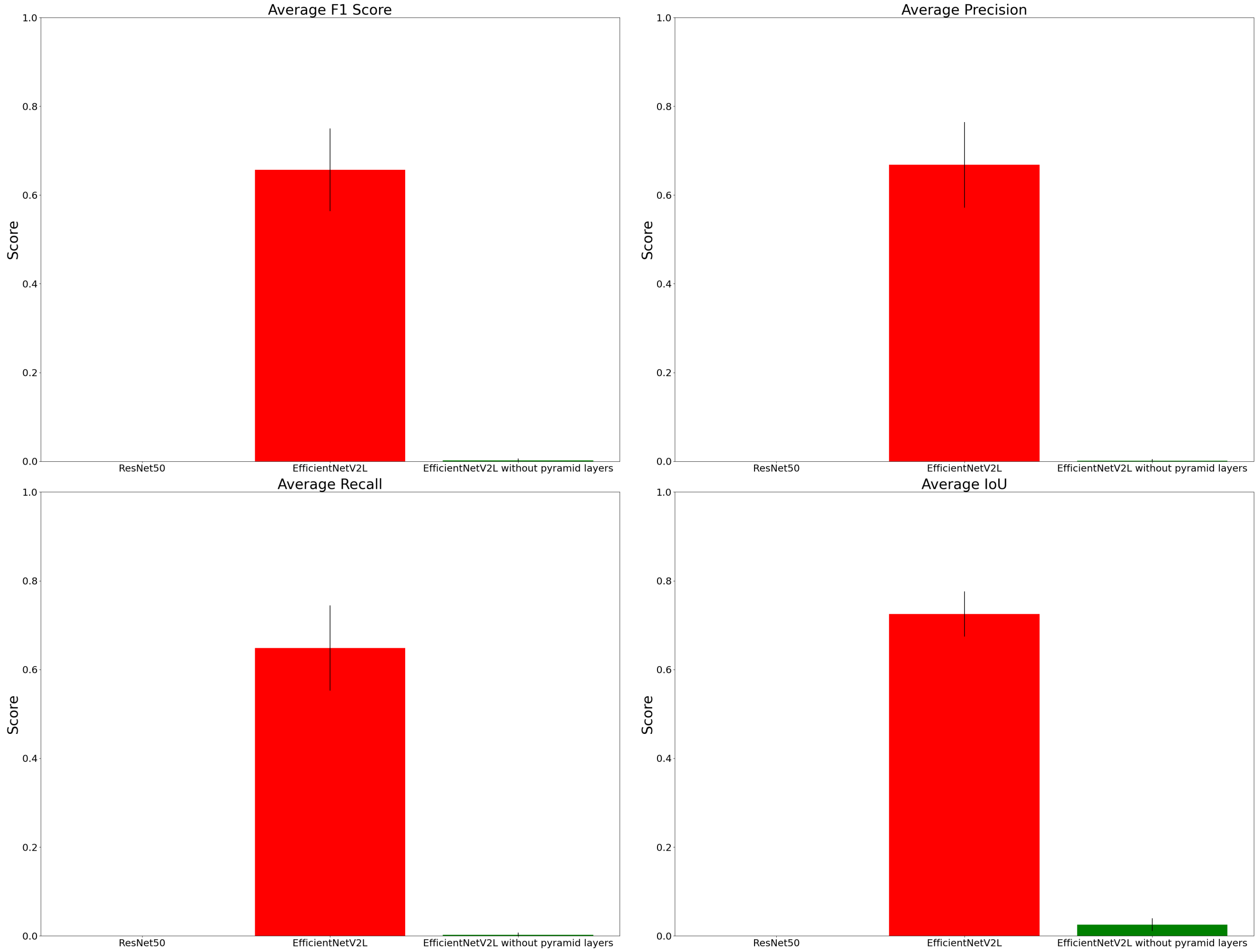

Figure 5 Comparison of average precision, F1 score, and Recall between ResNet50 (Blue), EfficientNetV2L (Red), and EfficientNetV2L (Green) without pyramid layers.

nodes. F1 score is also used as an additional metric to assess the model performance. F1 score is calculated as:

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

$$F1\ score = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

4. Dataset and Features

Datasets. I trained a model on DeepBacs datasets that consist of multiple bacterial species: *Escherichia coli* (E. coli), a *Staphylococcus aureus* (S. aureus) and *Bacillus subtilis* (B. subtilis). To train our model for bright field microscope image segmentation, I only used E. coli and B. subtilis for model training and testing. Each dataset is split into training, validation, and testing sets using a 60-20-20 split. Each image in E. coli datasets is three-channel, 512 x 512 pixel image, whereas each image in B. subtilis is three-channel, 1024 x 1024 pixel image. There were 113 images in the training set with over 1,337 cells in total from a diverse cultural density. B. subtilis is long rod-shape bacteria, whereas E. coli are rounded bacteria. This ensures that the model can segment images of bacteria with any shape. To generate label data, they manually annotated cells in training data by filling the annotated cells with a grey value of 1 and cell boundaries were drawn with a grey value of 2.

Data augmentation and processing. Training data were augmented with random flips, rotations, and scaling to increase the diversity of data.

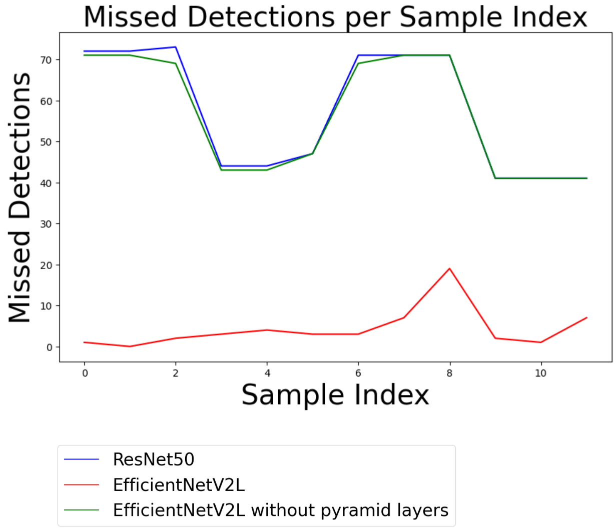

Figure 6 Comparison of missed detections per sample index between ResNet50 (Blue), EfficientNetV2L (Red), and EfficientNetV2L (Green) without pyramid layers. **(**Bottom) Comparison of missed detections per sample index between ResNet50 (Blue), EfficientNetV2L (Red), and EfficientNetV2L (Green) without pyramid layers.

5. Experiments/Result/Discussion

I experimented with multiple backbones of PanopticNet model. I used E. coli dataset, which is relatively smaller than other datasets to optimize and selected the best model architecture. I qualitatively inspected ground truth labels and predicted masks to assess the model performance. I used IoU as the cost to quantitatively evaluate my model performance. After segmentation, I compared between the model with Resnet50 backbone, the backbone used in Mesmer, and EfficientnetV2L, the backbone used in cell tracking study.

Qualitatively, EfficientNetV2L could segment E. coli images more efficiently than Resnet50 (Figure 4). Resnet50 could not produce a single segmentation of bacterial cell in the image. The model seemed to detect noises in the background of images. This is likely because Resnet50 is more sensitive to brightness gradients in the image than EfficientNetV2L and predicted the noises instead of true signals. I assessed their performance qualitatively by comparing IoU for images outputted by ResNet50 and EfficientNetV2L models. ResNet50 could not predict any cells and had average precision, recall, and F1 score equal to 0 (Figure 5). On the other hand, EfficientNetV2L predicted 666 cells from the total of 688 cells and yielded average precision, recall, and F1 score of 68.32%, 66.13%, and 65.7%, respectively (Figure 5). Comparing missed detections across different samples, EfficientNetV2L performed the best with only few samples with high missed detections (Figure 6). As a result, I chose EfficientNetV2L for further training on B. subtilis dataset.

I compared the model's performance with and without pyramid layers. The evaluation of output images revealed that the model incorporating pyramid layers P1-P7 outperformed the one without these layers, particularly in images featuring aggregated bacteria (Figure 4). This improvement is likely due to the pyramid layers enhancing the detection of subtle structures in the images. Consequently, the model with pyramid layers achieved higher precision, recall, and F1 scores (Figure 5). Based on these findings, I proceeded to train a B. subtilis model using the same training settings.

Finally, I trained the E. coli-pretrained model with B. subtilis data and compared the model trained on both E. coli and B. subtilis with the models trained individually on the E. coli and B. subtilis datasets. While the models performed relatively the same in B. subtilis test data. The individually trained model performed much better than the combined model on E. coli test data (Figure 7). This discrepancy is likely due to the unequal amount of training data, with much less data available for E. coli compared to B. subtilis, and the different in brightness gradients of these two datasets. Upon inspection, the combined model generated more noise in the inner and outer distance transforms than the individually trained models (Figure 8).

6. Conclusion and Future work

In this project, I trained a model for segmenting cells in bright field microscope images. Cell segmentation is a critical step for extracting single cell features from image data. An automated cell segmentation method would greatly enhance the speed of labeling cells while prevent human bias. I experimented the model with two different backbones: Resnet50 and EfficientNetV2L. The final model that performed the best was constructed on EfficientNetV2L with Adam optimizer and learning rate of 0.0001. The model worked very well in E. coli data but not B. subtilis data, suggesting that it cannot distinguish cells that are clustered together. To address this issue, the next step should involve performing augmentation to enhance the model's sensitivity to subtle image details.

Future steps should include comparing the model with EfficientNetV2L to other backbones or pyramid layers and performing cell segmentation on larger datasets and unseen test data from independent sources. Other models, such as U-net, which is often used for biological image segmentation, might also work. Finally, bright field microscope image data is limited when compared to medical imaging data such as MRI. In the future, a larger dataset containing cells from various species is likely to increase model performance.

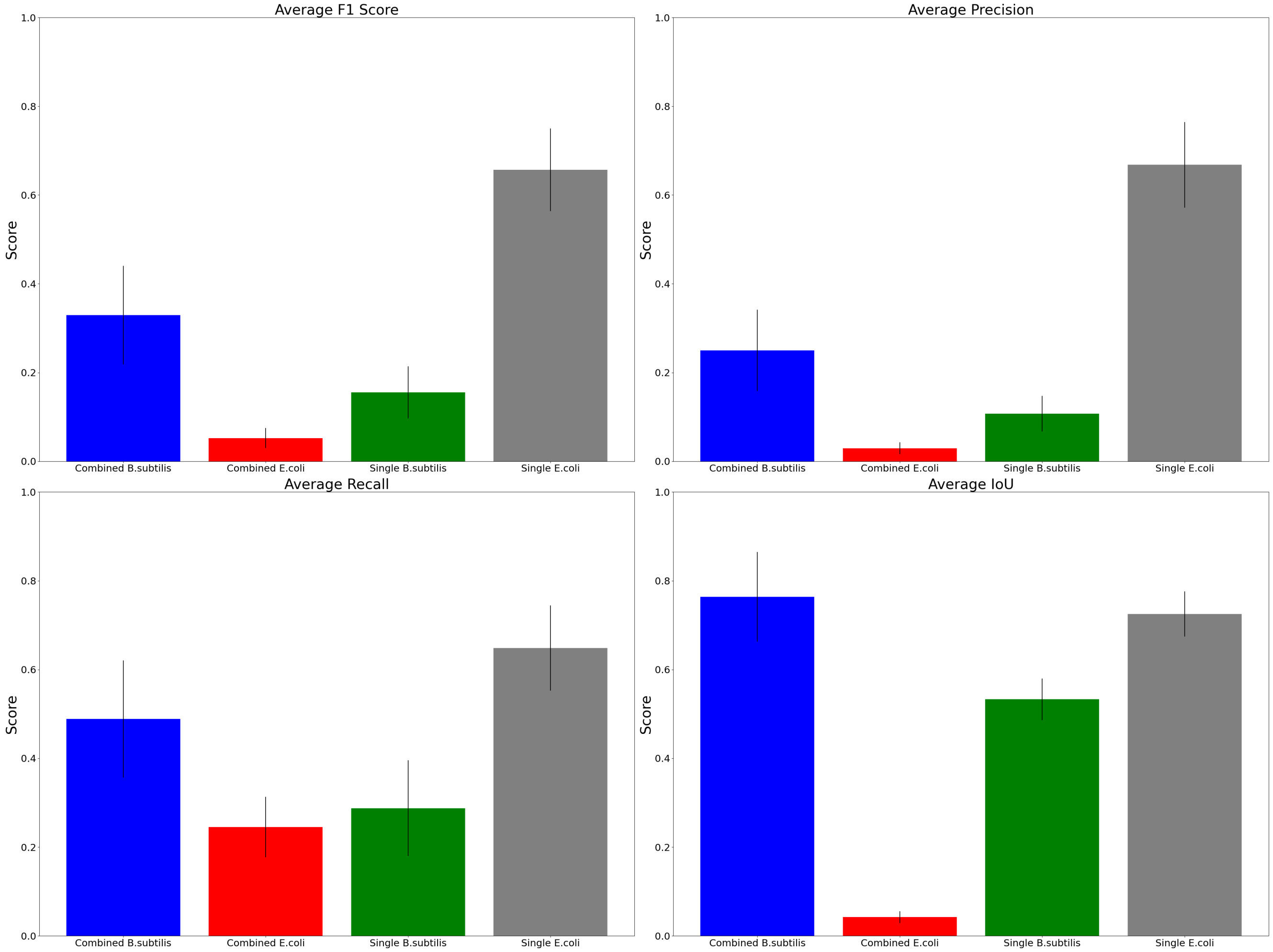

Figure 7 Comparison of combined model predictions on B. subtilis and E. coli test data and individually trained models. Individually trained models performed better in both datasets.

7. Contribution and Acknowledgements

PB designed experiments, trained the neural networks, evaluated and developed visualizations of model performance, and wrote the report.

8. References

[1] Greenwald NF, Miller G, Moen E, et al. Whole-cell segmentation of tissue images with human-level performance using large-scale data annotation and deep learning. Nat Biotechnol. 2022;40(4):555-565. doi:10.1038/s41587-021-01094-0A

[2] Stringer C, Wang T, Michaelos M, Pachitariu M. Cellpose: a generalist algorithm for cellular segmentation. Nat Methods. 2021;18(1):100-106. doi:10.1038/s41592-020-01018-x

[3] Spahn C, Gómez-de-Mariscal E, Laine RF, et al. DeepBacs for multi-task bacterial image analysis using open-source deep learning approaches. Commun Biol. 2022;5(1):688. Published 2022 Jul 9. doi:10.1038/s42003-022-03634-z

[4] N. Otsu, A Threshold Selection Method from Gray-Level Histograms, IEEE Transactions on Systems, Man, and Cybernetics, vol. 9, no. 1, pp. 62-66, Jan. 1979, doi: 10.1109/TSMC.1979.4310076

[5] Bradley, D., & Roth, G. (2007). Adaptive Thresholding using the Integral Image. Journal of Graphics Tools, 12(2), 13–21. https://doi.org/10.1080/2151237X.2007.10129236

[6] Prewitt JM, Mendelsohn ML. The analysis of cell images. Ann N Y Acad Sci. 1966;128(3):1035-1053. doi:10.1111/j.1749-6632.1965.tb11715.x

[7] Kass, M., Witkin, A. & Terzopoulos, D. Snakes: Active contour models. Int J Comput Vision 1, 321–331 (1988). https://doi.org/10.1007/BF00133570

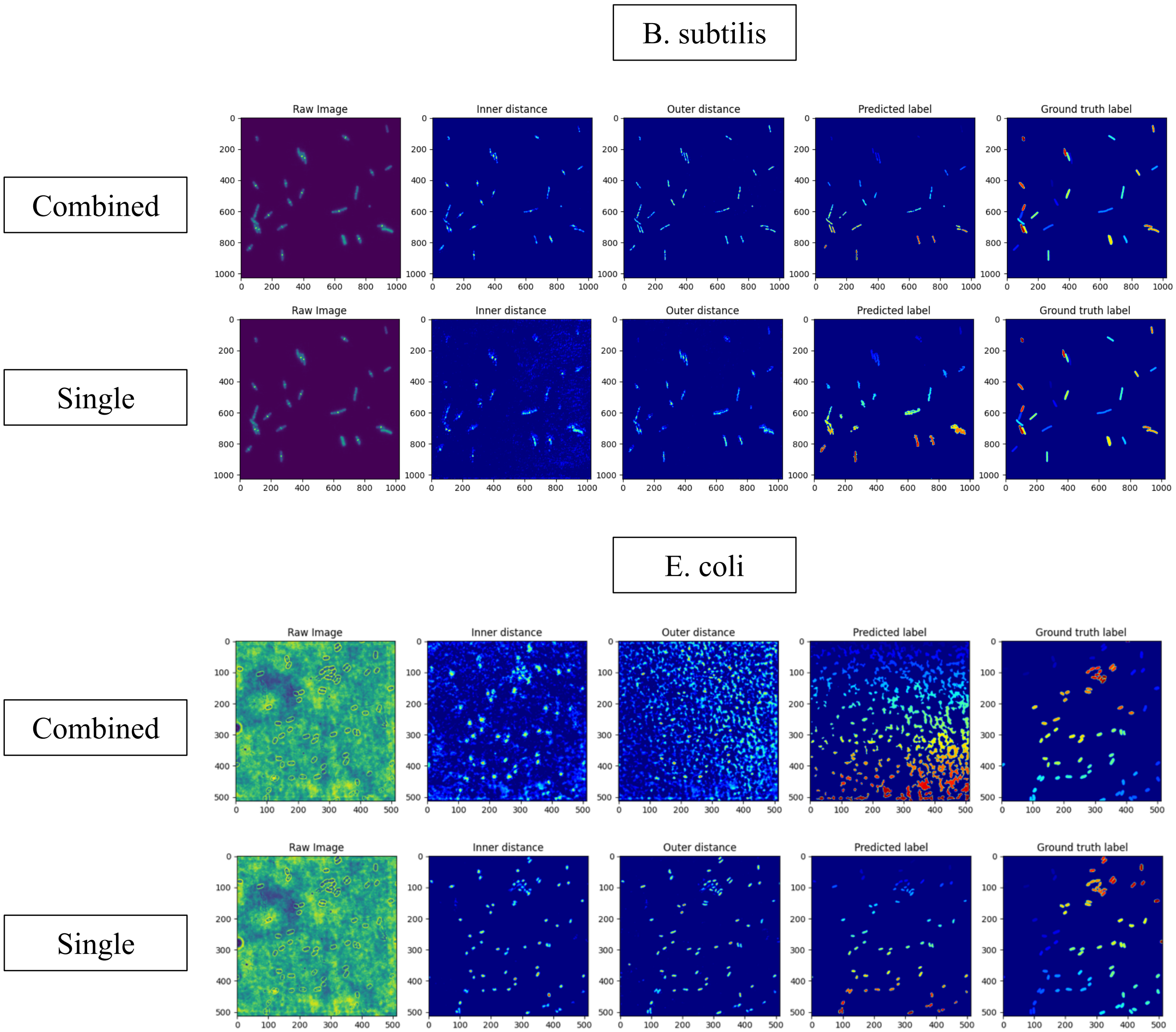

Figure 8 Representative Images of combined and single model predictions. Combined models detect more noises and produce inaccurate segmentation images.

[8] Beucher, S. Use of watersheds in contour detection. In Proceedings of the International Workshop on Image Processing, Astrophysics, Trieste, 4–8 June 1979

[9] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," in Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998, doi: 10.1109/5.726791.

[10] Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, CA, USA, 3–8 December 2012; pp. 1097–1105.

[11] Ciresan DC, Giusti A, Gambardella LM, Schmidhuber J. Deep neural networks segment neuronal membranes in electron microscopy images. NIPS. 2012

[12] T. -Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan and S. Belongie, "Feature Pyramid Networks for Object Detection," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 936-944, doi: 10.1109/CVPR.2017.106.

[13] Tan, Mingxing, and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks." International conference on machine learning. PMLR, 2019.

[14] Tan, Mingxing, and Quoc Le. "Efficientnetv2: Smaller models and faster training." International conference on machine learning. PMLR, 2021.

[15] Meyer, Fernand, and Serge Beucher. "Morphological segmentation." Journal of visual communication and image representation 1.1 (1990): 21-46.

[16] Moen, E. et al. Accurate cell tracking and lineage construction in live-cell imaging experiments with deep learning. Preprint at bioRxiv https://doi.org/10.1101/803205 (2019).