Segmentation of Dense Bacterial Populations in Fluorescence Microscopy Images

Trevor V. Palmiotto*
CSCI 4961 Deep Learning
Rensselaer Polytechnic Institute
Troy, NY 12180
palmit3@rpi.edu

1 Introduction

1.1 Motivation

Biological research often faces significant delays due to the time-intensive nature of experiments, sample shipping, and data analysis. While experiment duration and sample transportation are largely inflexible, analysis time remains a key area for optimization. In particular, fluorescence microscopy tasks such as identification of individual bacteria within a field of view (FOV) can turn data processing into a significant bottleneck when approached manually or with rule-based segmentation algorithms.

To this end, we explore the practical implementation of a deep learning semantic segmentation model built using a U-Net architecture to automate bacteria identification in fluorescence microscopy images. Rather than proposing a novel model, we demonstrate that such a model can be effectively trained even under the challenges of **limited and imperfect image-mask pairs**. These challenges are common in biological datasets, where annotations are often inconsistent or generated by suboptimal masking tools. Despite these challenges, our model performs well enough to integrate into a real-world image analysis pipeline, successfully segmenting hundreds of cells per image. This work highlights the importance of robust training and fine-tuning strategies to achieve deployable results, even when ideal datasets are unavailable.

This project builds on previous efforts within the Royer Lab at Rensselaer Polytechnic Institute (RPI). Earlier attempts to generate bacterial segmentation masks relied on fully manual annotation or the use of general-purpose deep learning segmentation models. Specifically, Royer Lab recently collaborated with Shuang Zhang of Wang Lab to produce bacterial cell masks using the robustly pre-trained foundation model 'Segment Anything Model' (SAM), a transformer-based approach made by Meta AI. As seen in Figure 1, SAM employs a ViT image encoder, prompt encoder, and mask decoder and requires little to no fine-tuning. In contrast, the success of our supervised model approach was entirely reliant on the availability of manually labeled image-mask pairs. This dependency on previous expert annotations emphasizes both the challenges and the potential of tailoring deep learning models to specific biological applications.

All data used for training and evaluation was created in collaboration with the Royer Lab, and the resulting model was deployed in an operational pipeline for bacterial image analysis. Moreover, the pipeline was created during the Spring 2025 semester as an undergraduate researcher in the Royer Lab and is currently in use, accelerating data analysis time.

*https://github.com/Trevor-Palmiotto/Royer-Lab-Bacterial-Semantic-Segmentation

Preprint. Under review.

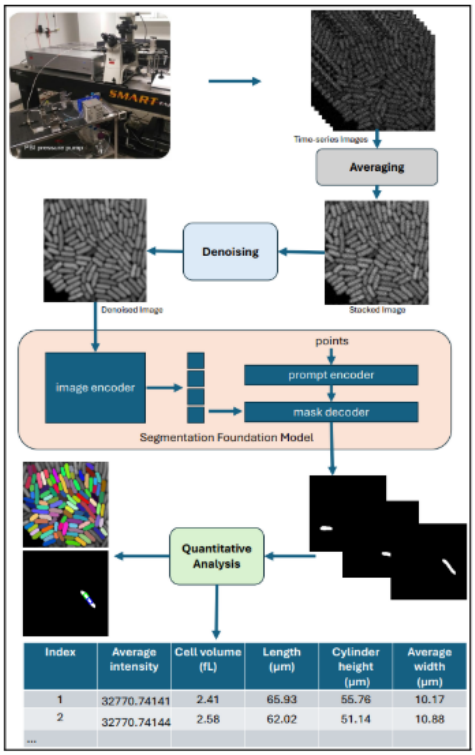

Figure 1: Numerous fluorescence scans are received from the microscope and averaged pixel-wise. Then the averaged image is denoised using the Block Matching and 3D filtering (BM3D) algorithm and passed into the SAM. Finally relevant quantitative analysis is extracted from masked cells.[3]

2 Methodologies

2.1 The Pipeline

To provide context for the deep learning component of this project, we first outline the broader image analysis pipeline that integrates the model and highlight its significance to the overall objective. Specifically, this pipeline is integral to supporting the Royer Lab's research on proteins involved in cell cycle and metabolism of extremophiles, where accurate single-cell measurements are essential. As illustrated in Figure 2, the pipeline processes fluorescence microscopy data to extract per-cell metrics (e.g., cell volume and protein concentration) using three specialized MATLAB scripts.

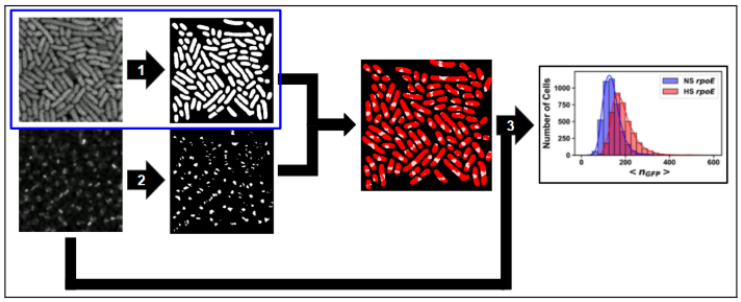

Figure 2: 1) Cells are identified and masked using the full-cell masking script. 2) Subcellular proteins of interest are identified and masked using the sub cell masking script. 3) Masks are used in combination with subcellular protein fluorescence microscopy data to calculate relevant metrics like concentration and cell size using the scanning number and brightness script. Note that in this diagram, the **task of the deep learning model is highlighted in blue.** [1]

Highlighting this project's contribution, the full-cell masking script (GUI seen in Figure 3) previously used a slow, imperfect manual approach, but now incorporates the U-Net model (shortly described) for rapid masking and higher accuracy. Specifically for the full-cell masking script, initial testing demonstrated a roughly 12-fold speedup in processing time. This reduction has significant returns to scale, especially considering that a single experiment generates upwards of 30 images - manually, each image takes roughly 10 minutes to mask.

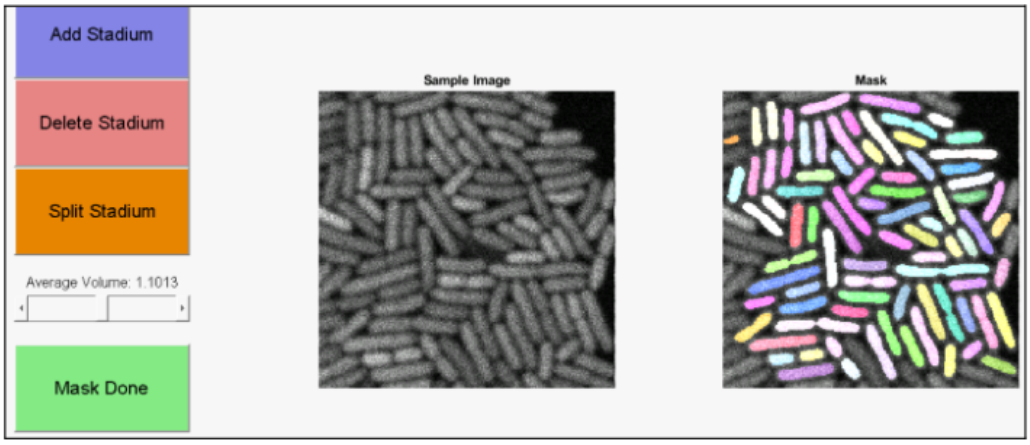

Figure 3: Full-cell masking script GUI. Various buttons (Add, Delete, Split) are used to correct the deep learning model's segmentation mask - colored to enhance the contrast of cell-specific masks. The script also includes a threshold slider to adjust which pixels are considered cellular, ensuring that only true cell volumes are retained for more accurate measurements. Note the script aligns the model prediction with analysis goals by effectively removing cells cut off by the edge of the FOV that do not represent true cell volumes. This ensures more accurate cell-volume results.

2.2 Model

The U-Net used in this project is a fully convolutional encoder-decoder architecture, as depicted in Figure 4. Model training was performed using PyTorch, with the specific implementation sourced from the Segmentation Models PyTorch (SMP) package. This SMP model introduces several improvements to the original U-Net, particularly in terms of stability and information flow. Key modifications include:

- **Encoder Architecture**: The final model incorporates ResNet50 pre-trained on the ImageNet dataset as the encoder.
- **Batch Normalization**: BatchNorm is applied after the first convolutional layer to enhance training stability and speed.
- **Network Depth**: The SMP model increases network depth to five layers, compared to the original's four layers.
- **Decoder Upscaling**: Instead of using transposed convolutions, the decoder upscales via bilinear scaling.
- **Padding**: Unlike the original U-Net, which lacks padding and results in a smaller semantic mask, our model uses 'same' padding to ensure consistent mask size. However, this padding may cause minimal fringe effects.

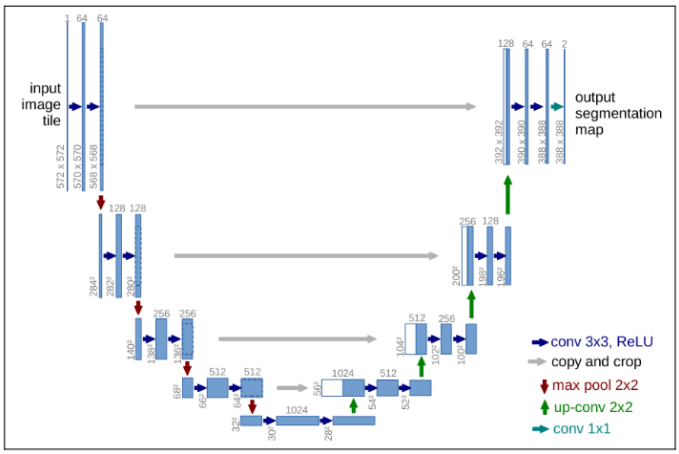

Figure 4: The aptly named U-Net is a fully convolutional network with a clear encoder-decoder structure. Diagram courtesy of [2]

2.3 Spatial Weight Maps and Data Handling

The authors of the original U-Net paper introduced *weight maps* to enforce instance segmentation by emphasizing the separation between adjacent cells during training. These maps assign pixel-wise weights to the loss function and are derived from the binary segmentation masks. The per-pixel weighted loss, visualized in Figure 5, is computed as:

$$w_i = \exp[-\frac{(d_1 + d_2)^2}{2\sigma^2}] \tag{1}$$

where:

- $d_1 \equiv$ distance to closest cell,
- $d_2 \equiv$ distance to second closest cell,
- $\sigma \equiv$ tunable hyperparameter

We adopt this formulation and introduce the **'image, mask, weight-map triplet'** as the core training unit. However, unlike the original U-Net implementation, which relied on highly precise instance masks, our training masks are coarser and blob-like due to limitations of the deprecated full-cell masking script. These inflexible, tubular masks lack form fitting and accurate boundaries, reinforcing the need for soft-labels and weight maps to guide the model in learning semantic and instance segmentation. The comparison of the first two images in row 2 of Figure 5 perfectly illustrates how poorly the binary masks match the cells and stresses the necessity for elastic deformation augmentation.

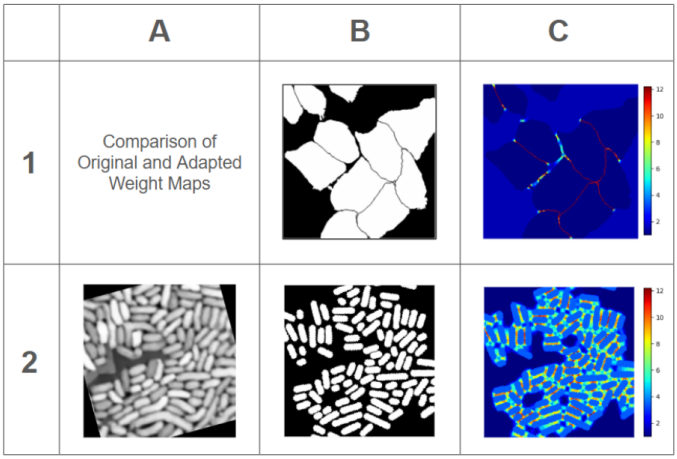

Figure 5: Row 1 is the original paper's implementation. Row 2 is our implementation. Column B compares the original paper's binary masks (1B) with our masks (2B). Column C compares the original paper's weight maps (1C) with our weight maps.

2.4 Data Handling

As with many biological segmentation tasks, the time-intensive nature of manual cell masking resulted in a limited number of image-mask pairs. However, because each FOV contains upwards of 100 cells, it is able to represent a sizable portion of the expected distribution of features. Additionally, to improve segmentation quality and free up model capacity, all 'image, mask, weight-map triplets' underwent key preprocessing steps. Initially, high-intensity pixels often corresponding to artifacts or photobleaching were removed from the input images, after which pixel values were standardized to have zero mean and unit variance. This standardization was key for the consistency of the subsequent denoising.

As visible in Figure 3, the sample images contain a notable degree of noise, which can hinder model performance. Intuitively, when noise is present, part of the model's capacity is spent compensating for irrelevant spatial variation rather than learning relevant features. To reduce this burden and improve signal clarity, images were denoised using the BM3D algorithm (resulting in as employed in Shuang Zhang's implementation using SAM. It is important to note we did initially train on unmodified images, but the model exhibited overfitting and poor cell instance segmentation, motivating the introduction of the denoising step.

2.5 Data Augmentation

To increase the model's exposure and reduce overfitting, training-time data augmentation techniques were employed as well. These included rotation, horizontal and vertical flipping, random cropping, and light elastic deformations. Elastic deformation was particularly important because as mentioned, the utilized manual masking tool assumes all cells are straight and uniformly shaped. In reality, bacterial cells may exhibit curvature or slight deviations in shape. Therefore, we used elastic deformation to introduce a warping effect to improve the model's semantic segmentation capabilities.

2.6 Custom Loss Function

This project incorporates a combination of two loss functions. Namely, predictions were evaluated against the manual mask using Binary Cross-Entropy and Dice Loss, seen in equations 2 and 3 respectively.

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{N}\sum_{i=1}^{N} y_i w_i \log(p_i) \tag{2}$$

where:

- $y_i \in [0, 1]$ ground truth label for pixel i,

- $p_i \equiv \sigma(x_i) \in [0, 1]$ predicted probability,
- $w_i \in [0, 1]$ pixel-wise weight,
- $N \equiv$ total number of pixels

$$\mathcal{L}_{\text{Dice}} = 1 - \frac{\epsilon + 2\sum_{i=1}^{N} y_i p_i w_i}{\epsilon + \sum_{i=1}^{N} y_i w_i + \sum_{i=1}^{N} p_i w_i} \tag{3}$$

where:

- $y_i \in [0, 1]$ ground truth label,
- $p_i \in [0, 1]$ predicted probability,
- $w_i \in [0, 1]$ pixel-specific loss weight,
- $N \equiv$ total number of pixels,
- $\epsilon \equiv$ numerical stability constant.

Each function focuses on a different purpose. Weighted BCE was used to learn fine-grained, per-pixel correctness - important for soft labels and smooth transitions between objects. Dice was used to ensure structural overlap, i.e., the predicted shapes and regions match the target. Together, these weighted loss functions are added via an α value such that:

$$\mathcal{L} = (1 - \alpha)[1 - \frac{\epsilon + 2\sum_{i=1}^{N} y_i p_i w_i}{\epsilon + \sum_{i=1}^{N} y_i w_i + \sum_{i=1}^{N} p_i w_i}] - \alpha[\frac{1}{N}\sum_{i=1}^{N} y_i w_i \log(p_i)] \tag{4}$$

where:

- $y_i \in [0, 1]$ ground truth label,
- $p_i \in [0, 1]$ predicted probability,
- $w_i \in [0, 1]$ pixel-wise weight,
- $N \equiv$ total number of pixels,
- $\epsilon \equiv$ small constant added for numerical stability.

3 Experiments

The Royer Lab studies both large and small bacterial strains; accordingly, we trained an initial model to learn generalizable cell features using images masked by annotators of varying expertise, resulting in diverse, but generally accurate masks on a wide range of cell sizes. Then, using a highly-limited number of size-specific masks, we fine tuned two models - one tailored to small tightly packed cells, and another tailored to large cells. In both pre-training and fine tuning, the model was trained using a crop scheduler, a learning rate scheduler, and early stopping.

3.1 Pre-Training

Pre-training used **a total of just 16 image-mask pairs**, with a train-test split of (12 images vs 4 images). Note that in this stage, we used α = 0.2 to motivate learning of general cell features. The following is an overview of the experiment setup for initial training:

- **Device:** Google Colab, T4 GPU
- **Augmentation:** crop size scheduling, α = 0.2, flipping (p = 0.5), elastic deform (p = 0.5)
- **Optimizer:** AdamW(lr = 1e-4, weight_decay = 1e-3) w/ learning rate scheduler
- **Early Stopping Patience:** 50

3.2 Fine Tuning

Two models were then fine-tuned to better fit both the large and small cell feature distributions. Each model used a total of 8 image-mask pairs for fine tuning - 6 for training and 2 for testing. To ensure better precision on these high accuracy masks, we set α=0.4 to further increase the influence of the Weighted BCE portion of the loss function.

- **Device:** Google Colab, T4 GPU
- **Augmentation:** crop size scheduling, α = 0.4, flipping (p = 0.5), elastic deform (p = 0.5)
- **Optimizer:** AdamW(lr = 1e-4, weight_decay = 1e-3) w/ learning rate scheduler
- **Early Stopping Patience:** 30

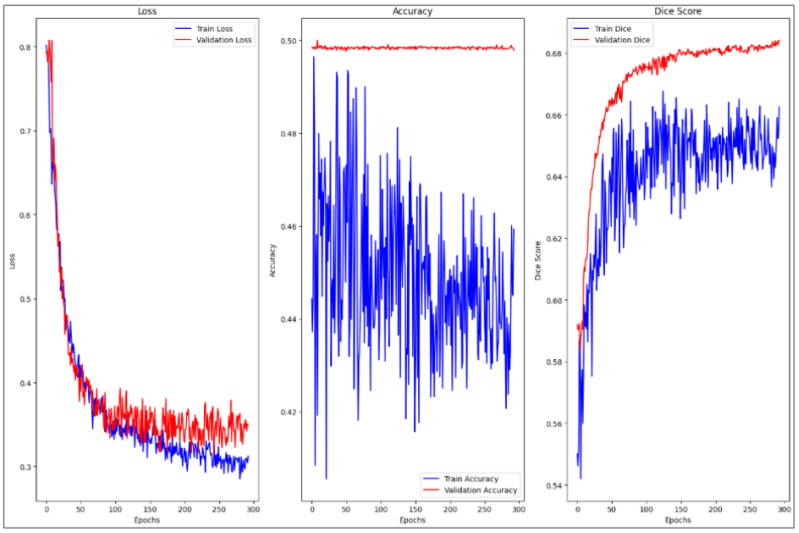

Figure 6: Pre-training learning visualization. Note strong generalization to test set. **The evaluation accuracy (MSE) function was not properly implemented** in this step. Note however, evaluation was done with gradient disabled, and was therefore ignored in gradient calculations

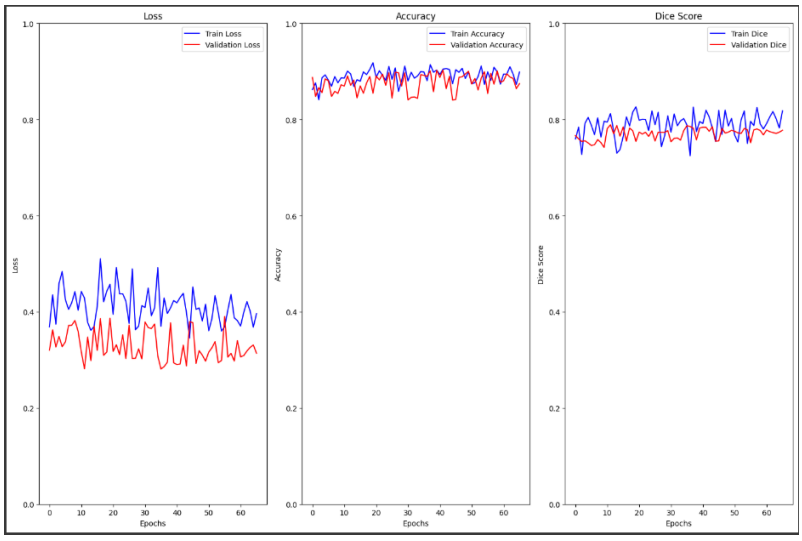

Figure 7: Fine tuning training visualization of large cell model. Note consistent generalization of test set. Accuracy metric calculated correctly, and proper axis scaling shows larger picture.

4 Conclusion

Despite the seemingly low accuracies and dice scores seen in Figures 6 and 7, when the model performs proper instance segmentation, we see that the predicted mask conforms to the cell shape far more faithfully than the original mask and we believe this explains the seemingly low metrics. The numerical evaluation is misleading, and visual evaluation is a more telling metric. See Figure 8 below. Note, the model predicts all cells, regardless of location on the image; however, edge cells are removed later in the full-cell masking script to ensure accurate cell-size metrics.

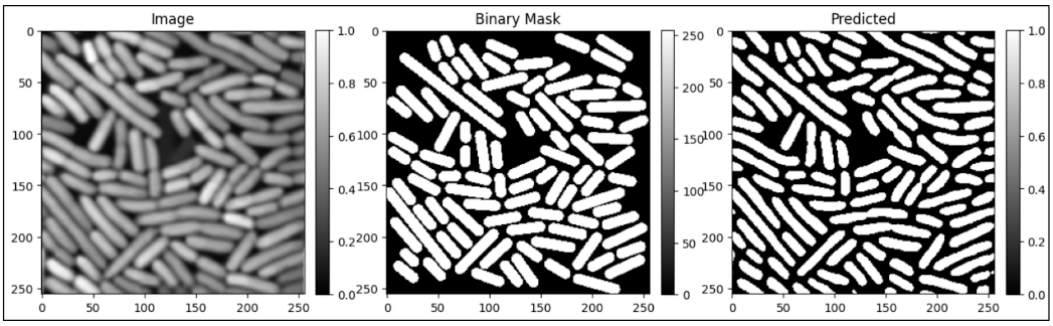

Figure 8: Visual comparison of true mask against large cell model's predicted mask. Note that the model performs extremely well with separating side-by-side cells, but struggles in instances of end-to-end overlay.

This work presents a deep learning-based image analysis pipeline that significantly optimizes bacterial cell segmentation in fluorescence microscopy images. By replacing slow, variable-quality manual annotations with a U-Net model trained and fine-tuned on diverse cell types we achieved rapid, consistent full-cell masking. The introduction of spatial weight maps helped the model learn instance-like segmentation behavior and data augmentation techniques ensured precise semantic segmentation. These models and the encompassing pipeline enable the Royer Lab to extract highly accurate per-cell measurements to better understand extremophiles.

References

[1] Carleton H. Coffin, Luke A. Fisher, Sara Crippen, Phoebe Demers, Douglas H. Bartlett, and Catherine A. Royer. Response and adaptation of the transcriptional heat shock response to pressure. *Frontiers in Microbiology*, Volume 15 - 2024, 2024.

[2] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015.

[3] Shujie Li Karyn L. Rogers Catherine Royer Ge Wang Shuang Zhang, Carleton Coffin. Ai-driven high-resolution imaging and quantitative analysis of extremophile microbes. Poster, 2025.