CPSC4800 Computer Vision Final Project Report

Yijun Pan
Yale University
yijun.pan@yale.edu

Jessy Xue
Yale University
jessy.xue@yale.edu

Boundary-Aware U-Net for Instance Cell Segmentation in Brightfield Microscopy

Names

The project is conducted by:

- Yijun Pan (Yale University)
- Jessy Xue (Yale University)

1 Introduction

Cell segmentation is a core computer vision task in biomedical imaging, enabling raw microscopy data to be converted into quantitative single-cell measurements. Segmentation outputs support downstream analyses such as cell counting, morphology quantification, growth dynamics measurement, and cell tracking, which are widely used in basic cell biology, phenotypic screening, and drug discovery.

In brightfield time-lapse microscopy, instance segmentation is especially challenging. Compared with fluorescence imaging, brightfield images often exhibit low boundary contrast, heterogeneous textures, and imaging artifacts, and cells frequently touch or overlap. As a result, a standard binary foreground/background model may achieve high pixel-level accuracy while still failing at the key practical requirement: preventing adjacent cells from being merged into a single connected component.

In this project, we study the BF-C2DL-HSC sequence from the Cell Tracking Challenge [1] and focus on improving separation of touching cells. We adopt a boundary-aware learning formulation that explicitly models regions near cell borders and cell–cell interfaces, rather than treating the entire cell as a single foreground class. Our pipeline uses a two-stage training strategy with a 2-class warm-up U-Net followed by a 4-class decomposition trained with a J-regularized cross-entropy objective to address severe class imbalance in thin boundary/interface regions. During inference, we retain only the predicted cell-interior region and discard boundary-related classes before connected-component labeling, which helps create separators between neighboring cells and improves instance-level segmentation quality.

Code is available at `https://github.com/charles-pyj/4800_Final_Project`

2 Related Work

Deep learning has become the dominant approach for biomedical image segmentation, with U-Net as a widely used baseline due to its encoder–decoder structure and skip connections that preserve fine spatial details Ronneberger et al. [2015]. In practice, U-Net-style models are often paired with strong

[1] `https://celltrackingchallenge.net/`

Preprint. Under review.

convolutional backbones (e.g., ResNet encoders) to improve feature extraction and generalization He et al. [2015] . For microscopy benchmarks such as the Cell Tracking Challenge (CTC), these models can achieve high pixel-level accuracy, yet still struggle with a key instance-level failure mode: when cells touch and boundaries are weak (common in brightfield time-lapse imaging), binary foreground/background predictions tend to merge neighboring cells into a single connected component.

To address instance separation more directly, Cellpose predicts spatial vector fields that guide pixels toward cell centers, enabling robust segmentation across diverse microscopy modalities and reducing merges in crowded scenes Stringer et al. [2020]. Another complementary direction is boundary-aware learning that explicitly emphasizes thin interfaces under severe class imbalance. In particular, augmenting cross-entropy with a J-regularization term has been shown to improve imbalanced multiclass segmentation and encourage consistent separation among adjacent regions Peña et al. [2019]. Our approach follows this boundary-aware direction: we use a U-Net backbone with a multi-class label decomposition that includes boundary/interface regions, and train with a J-regularized objective to reduce under-segmentation of touching cells in brightfield time-lapse microscopy.

3 Methodology

3.1 Problem Formulation

Problem formulation. Given a microscopy image $I \in \mathbb{R}^{H \times W \times 3}$ containing an unknown number of densely packed cells, the goal of multi-cell segmentation is to assign each pixel p a unique instance label indicating which cell it belongs to, or background if no cell is present. This task is substantially more difficult than binary foreground–background segmentation, because cells often touch, overlap in projection, or exhibit low-contrast boundaries that make them indistinguishable in raw pixel space. Traditional semantic segmentation models predict only per-pixel class probabilities and therefore cannot directly separate adjacent cells that share the same semantic category; when two cells touch, their predicted masks merge into a single connected component, causing systematic under-segmentation. Conversely, introducing instance-level cues naïvely (e.g., via connected components) fails when boundaries are weak or when small gaps between cells are missing due to imaging noise. These challenges motivate the need for segmentation formulations that incorporate explicit geometric and contextual information about the structure of cell boundaries, enabling reliable disentanglement of tightly clustered instances.

3.2 Warmup Segmentation Training

Setup We first train a U-Net Ronneberger et al. [2015] with a ResNet-34 He et al. [2015] encoder on a simple binary segmentation task where the model is only required to tell the cell-region from the background. This warmup phase has two motivations: (i) it provides a strong initialization for coarse cell localization, and (ii) it allows the network to learn general morphological features of cellular structures before solving the harder task.

After the warmup, the learned weights, except for the final segmentation head, are transferred to a new model configured for later training. Note that the predictor head of the warm-up model will not be transferred because of class count misnatch.

Objective Function During the warmup stage, we train the model only to distinguish foreground cells from background. Let the model output a logit map over two classes, and let $p \in [0, 1]^{H \times W}$ denote the predicted foreground probability after the softmax, and let $gt \in \{0, 1\}^{H \times W}$ be the binary ground-truth mask. We optimize a hybrid loss that combines pixel-wise cross-entropy with a Dice loss to improve stability on highly imbalanced cell regions. The cross-entropy term is

$$\mathcal{L}_{\text{CE}} = -\sum_i \big[gt_i \log p_i + (1 - gt_i) \log(1 - p_i)\big],$$

and the Dice component measures the overlap between prediction and ground truth,

$$\text{Dice}(p, gt) \ = \ \frac{2 \sum_i p_i gt_i}{\sum_i p_i + \sum_i gt_i + \varepsilon}, \qquad \mathcal{L}_{\text{Dice}} = 1 - \text{Dice}(p, gt),$$

where ε is a small constant for numerical stability. The final warmup loss is a weighted combination:

$$\mathcal{L}_{\text{warmup}} = \mathcal{L}_{\text{CE}} \; + \; \tfrac{1}{2}\,\mathcal{L}_{\text{Dice}}.$$

This formulation encourages the model not only to classify pixels correctly but also to produce spatially coherent masks that align well with entire cell regions.

3.3 Four-Class Semantic Decomposition

To enable finer separation of adjacent cells, we replace the binary labels with a four-class semantic decomposition to help the model better understand the area in between cells. Let the instance annotation be an integer mask where each cell has a unique ID. From this mask we algorithmically derive four classes:

1. background (far from any cell);
2. cell interior (high-confidence cell core);
3. near-cell ring (a thin band immediately outside the cell interior);
4. inter-cell interface (pixels near boundaries between two or more touching cells).

The motivation is that classes (2) and (3) provide the network with geometric and topological cues about where one cell ends and another begins. In contrast to a purely binary formulation, the network is now explicitly encouraged to recognize boundary structures. During inference, only class (1) is retained as the "true foreground," while classes (2)–(3) are removed, effectively carving out separators between touching cells before connected-component analysis. This procedure significantly improves instance disentanglement.

Label Construction To obtain class 3 and 4 labels, we derive two additional geometric structures from the instance mask. The inter-cell interface (class 3) is defined by identifying pixels whose 4-connected neighbors belong to different nonzero instance IDs. Formally, for each pixel i with label $gt_i > 0$, we check its neighbors $j \in \mathcal{N}_4(i)$ and assign class 3 whenever $gt_j > 0$ and $gt_j \neq gt_i$. This produces a thin, typically one-pixel band that marks where two distinct cells touch.

The near-cell outer ring (class 2) is constructed by dilating the union of all cell regions. Let $\mathcal{C} = \{i : gt_i > 0\}$ and let $\mathrm{Dilate}(\mathcal{C}, r)$ denote a binary dilation with radius $\texttt{ring\_width} = r$. We define the outer ring as

$$\mathcal{R} = \mathrm{Dilate}(\mathcal{C}, r) \setminus \mathcal{C},$$

excluding pixels already assigned to the interface class. The ring width r controls how much of the local cell neighborhood is exposed to the model: larger r values create a thicker context band that improves the model's ability to separate crowded or overlapping cells, whereas smaller r keeps the ring tight, reducing the risk of introducing unnecessary ambiguous regions. Together, the interface band and the outer ring provide structured boundary cues that significantly enhance the model's capacity for fine-grained cell separation.

3.4 J-Calibrated Loss for Multi-Class Consistency

In the second training phase, where the model predicts four semantic classes, we augment the standard cross-entropy loss with a J-regularization Peña et al. [2019] term that improves class balance and encourages consistent discrimination among the four regions. Let $p_i(p)$ denote the predicted probability of class $i \in \{0, \ldots, 3\}$ at pixel p, and let $y_i(p) \in \{0, 1\}$ be the corresponding one-hot ground-truth label. Define $n_i = \sum_p y_i(p)$ as the number of pixels belonging to class i in the batch. The base loss is the usual pixel-wise cross-entropy,

$$\mathcal{L}_{\text{CE}} = -\sum_p \sum_{i=0}^{3} y_i(p) \log p_i(p).$$

To incorporate relational information between classes, we introduce for every pair (i, k) the contrast term

$$\Delta_{ik}(p) = \tfrac{1}{2}\left(\frac{y_i(p)}{n_i} - \frac{y_k(p)}{n_k}\right),$$

which weights each pixel according to how strongly it distinguishes class i from class k. For each class pair we compute

$$S_{ik} = \sum_{p} p_i(p)\, \Delta_{ik}(p),$$

and form the J-regularization contribution

$$\mathcal{R}_J = \sum_{i=0}^{3} \sum_{k=0}^{3} \lambda \, \log\!\big(0.5 + S_{ik} \big),$$

where λ is a scalar controlling the strength of the regularization. Intuitively, positive S_{ik} values indicate that the model places higher probability on class i at pixels that uniquely support class i (relative to k), and the log term rewards such consistent separation.

The final 4-class training loss is

$$\mathcal{L}_{\text{4class}} = \mathcal{L}_{\text{CE}} \; - \; \alpha \, \mathcal{R}_J,$$

where α controls the overall influence of the J-regularization. This formulation encourages the network not only to fit pixel labels, but also to preserve coherent relationships between the four geometric regions, improving boundary quality and cross-class separation during instance reconstruction.

3.5 Instance Reconstruction and Evaluation

At test time, model predictions are thresholded by selecting only the **cell-interior class** as foreground. Classes (2) and (3), which correspond to near-boundary regions, are discarded to yield a foreground mask in which touching cells have been separated. Connected-component labeling is then applied to reconstruct individual cell instances. Performance is measured using the SEG accuracy metric, which matches predicted components to ground-truth instances via IoU-based pairing.

3.6 Summary

In summary, the overall pipeline consists of: (i) warmup binary training for coarse cell localization, (ii) transfer-learning to a four-class model for boundary-aware semantic decomposition, (iii) J-calibrated loss for improved multi-class confidence structure, and (iv) instance reconstruction that exploits the predicted boundary classes. Together these components address the core challenge of separating densely clustered cells and yield substantially improved instance-level segmentation performance.

3.7 Optimization Algorithm

Optimization. For the all training, the model is optimized for 30 epochs using the Adam optimizer with a learning rate of $1{\times}10^{-3}$. Unless otherwise specified, we employ the standard cross-entropy loss over the four semantic classes, which provides stable supervision during this stage. Model parameters are updated end-to-end, and all training batches contribute equally to the loss. This configuration balances optimization stability with sufficient capacity to refine boundary-aware representations learned in the warmup phase.

4 Data

We use the BF-C2DL-HSC dataset from the Cell Tracking Challenge (CTC) Maška et al. [2014b], a public benchmark for time-lapse microscopy segmentation and tracking. BF-C2DL-HSC contains 2D brightfield time-lapse sequences of mouse hematopoietic stem cells (HSCs) in hydrogel microwells. The dataset documentation and download links are available on the CTC website at `https://celltrackingchallenge.net/2d-datasets/` (BF-C2DL-HSC) and the benchmark results page at `https://celltrackingchallenge.net/latest-csb-results/`.

Each sample is a single grayscale 2D frame stored as a TIFF image (e.g., `tXXXX.tif`). Ground-truth annotations are provided as instance segmentation masks (e.g., `man_segXXXX.tif`), where background pixels are 0 and each cell instance is encoded by a unique positive integer ID. We use the labeled training sequences `01` and `02` and treat each frame–mask pair as one supervised example. Since CTC test labels are not publicly released, we pool frames from both sequences and randomly split them into 90% training and 10% validation using random_split with a fixed seed (42) for reproducibility, and select the best checkpoint based on validation performance.

For preprocessing, we keep images at their native spatial resolution (no resizing/cropping), convert them to single-channel float32, and apply per-image z-score normalization $(x - \mu)/(\sigma + 10^{-6})$. No additional geometric or color augmentations are applied. For the warm-up (binary) stage, instance masks are binarized (0 for background and 1 for any cell). For the boundary-aware stage, we convert each instance mask into a 4-class semantic target using make_4class_labels with ring width 1 pixel: 0 = background, 1 = cell interior, 2 = outer ring around each cell, and 3 = interface between touching cells.

5 Implementation Details

Data preprocessing. We use the BF-C2DL-HSC datasetMaška et al. [2014a](sequences 01 and 02) and read each frame and corresponding instance mask from the original .tif files using a custom PyTorch Dataset. All images are kept at their native resolution and converted to single–channel float32. For every image we perform per-image Z-score normalization, $(x - \mu)/(\sigma + 10^{-6})$, and no additional geometric or color augmentations are applied. For the warm-up (binary) model, instance masks are binarized (0 for background and 1 for any cell). For the boundary-aware 4-class model, we convert the instance masks into four semantic classes using make_4class_labels with a ring width of 1 pixel: 0 = background, 1 = cell interior, 2 = a thin outer ring around each cell, and 3 = the interface between touching cells. For both phases, we pool frames from both sequences and randomly split them into 90% training and 10% validation using random_split with a fixed seed (42) for reproducibility.

Models and hyperparameters. We use the Unet implementation from segmentation_models_pytorch with a ResNet-34 encoder and one input channel. In the warm-up phase, we train a 2-class U-Net (background vs. cell) with an ImageNet-pretrained encoder using the Dice loss and the Adam optimizer. In the second phase, we instantiate a 4-class U-Net (same encoder, 4 output channels) and initialize it from the 2-class network by copying all parameters whose shapes match, so that encoder and decoder weights are warm-started. The 4-class model is then fine-tuned on the 4-class labels using the J-regularized cross-entropy loss (JCrossEntropyLoss with $\lambda_{\text{val}} = 0.5$ and reg_scale = 1.0) and Adam. In both phases we use a fixed learning rate of 1×10^{-3}, batch size 4, and train for 30 epochs, keeping the checkpoint with the best validation SEG score. We do not use weight decay, momentum (beyond Adam's defaults), or explicit dropout beyond what is built into the backbone.

Hyperparameter	Value
Optimizer	Adam
Learning rate	1×10^{-3}
Batch size	4
Epochs (2-class warm-up)	30
Epochs (4-class model)	30
Weight decay	0
Loss (2-class)	Dice loss
Loss (4-class)	JCrossEntropyLoss (λ_{val}=0.5, reg_scale=1.0)

Table 1: Key training hyperparameters for both segmentation phases.

Hardware and software. All experiments are implemented in Python using PyTorch 2.7.1+cu118, torchvision 0.22.1+cu118, and segmentation_models_pytorch 0.5.0, together with standard scientific Python packages (NumPy, SciPy, tifffile, and matplotlib). Training and evaluation are performed on a single CUDA-capable GPU using the PyTorch DataLoader API for mini-batch loading.

6 Results

We report both quantitative and qualitative results on the BF-C2DL-HSC brightfield microscopy sequence. Our evaluation focuses on two complementary aspects: (i) pixel-level overlap quality, and (ii) instance-level separation of touching cells after connected-component labeling.

We compare three settings: a baseline U-Net, our 2-class warm-up U-Net, and our boundary-aware 4-class U-Net trained with the J-regularized objective. Quantitative results are summarized in Table 2. Overall, the warm-up model achieves a high Dice score but only moderate instance separation (SEG), whereas the boundary-aware 4-class model substantially improves SEG, indicating better disentanglement of touching cells. Qualitative comparisons are shown in Fig 1.

6.1 Evaluation Metrics

We use Dice to measure pixel-level foreground overlap, and SEG to evaluate instance-level segmentation accuracy under the Cell Tracking Challenge (CTC) matching rule.

Dice coefficient. Given a binary prediction $P \subseteq \Omega$ and binary ground truth $G \subseteq \Omega$, the Sørensen–Dice coefficient is

$$\text{Dice}(P, G) \;=\; \frac{2|P \cap G| + \varepsilon}{|P| + |G| + \varepsilon}, \tag{1}$$

where $|\cdot|$ counts foreground pixels and ε is a small constant for numerical stability (we use $\varepsilon = 10^{-6}$ in code).

SEG score (instance segmentation accuracy). Let $\mathcal{R} = \{R_1, \dots, R_N\}$ be ground-truth instances and $\mathcal{S} = \{S_1, \dots, S_M\}$ be predicted instances obtained by connected-component labeling on the foreground mask. Define IoU (Jaccard index) as

$$J(A, B) \;=\; \frac{|A \cap B|}{|A \cup B|}. \tag{2}$$

For each ground-truth instance R_i, we consider predicted instances S_j that satisfy the CTC detection criterion

$$\frac{|R_i \cap S_j|}{|R_i|} > 0.5. \tag{3}$$

Among those, we take the best IoU match; if no prediction satisfies the criterion, the score is 0. The final SEG is the mean best-match IoU:

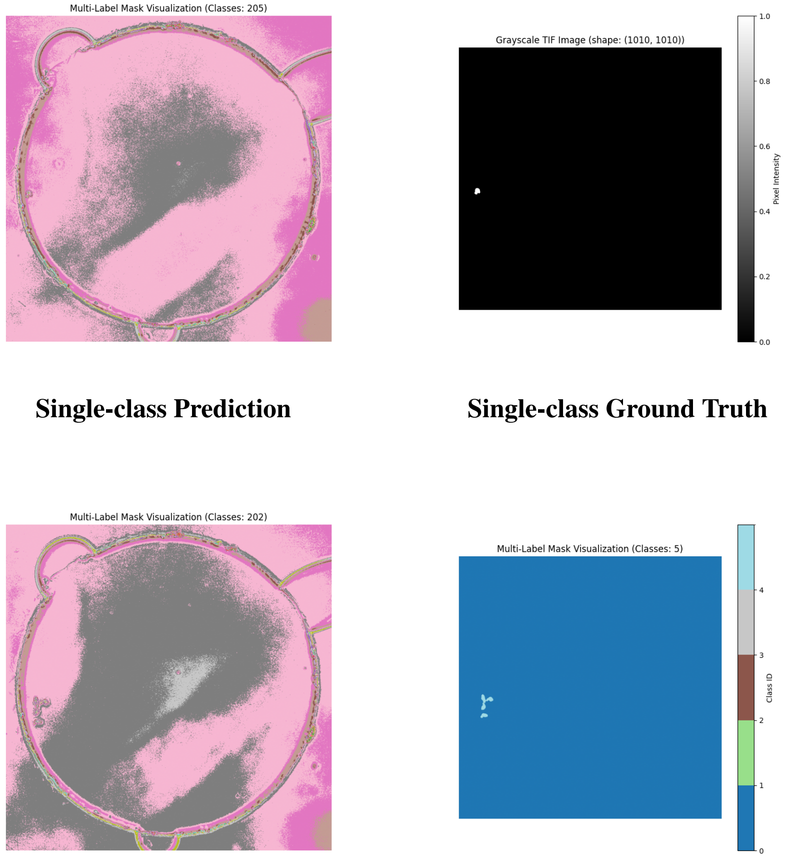
$$\text{SEG} \;=\; \frac{1}{N} \sum_{i=1}^{N} \max_{S_j \in \mathcal{S}} \Big\{ J(R_i, S_j) \;:\; \frac{|R_i \cap S_j|}{|R_i|} > 0.5 \Big\}. \tag{4}$$

Dice is appropriate for measuring overall foreground overlap (useful for checking whether the model correctly localizes cell regions), but it can remain high even when multiple touching cells are merged into one component. SEG complements Dice by explicitly evaluating instance-level correctness via IoU-based matching, making it sensitive to the primary failure mode in crowded brightfield microscopy: under-segmentation due to weak boundaries.

6.2 Quantitative Results

Model	Dice Score (single class)	SEG Score (multi-class)
Baseline U-Net	0.012	0.
2-class Warm-up U-net (Ours)	0.983	0.519
4-Class U-Net with J loss (Ours)	0.952	0.905

Table 2: Comparison of Dice and SEG performance across models.


Multi-Label Mask Visualization (Classes: 205)
Grayscale TIF Image (shape: (1010, 1010))
Pixel Intensity
0.0
0.2
0.4
0.6
0.8
1.0
Multi-Label Mask Visualization (Classes: 202)
Multi-Label Mask Visualization (Classes: 5)
Class ID
0
1
2
3
4

Single-class Prediction **Single-class Ground Truth**

Multi-class Prediction **Multi-class Ground Truth**

Figure 1: Comparison of single-class (top row) and multi-class (bottom row) segmentation.

7 Discussion

A key insight from this project is that pixel-level overlap metrics alone can be misleading for microscopy *instance* segmentation in crowded scenes. In brightfield time-lapse data, a model can achieve very high Dice by correctly labeling most cell pixels as foreground, yet still fail at the practical goal of separating touching cells. This motivated our use of an instance-sensitive metric (SEG) and a boundary-aware formulation that explicitly models cell interiors and cell–cell interfaces, which better targets the dominant failure mode of under-segmentation.

The main challenge was weak and ambiguous boundaries in brightfield images, where adjacent cells often appear as a single connected region. This problem is compounded by severe class imbalance: interface pixels are thin and rare compared with interior and background regions. We found that a two-stage strategy worked well in practice: a 2-class warm-up model learns robust localization, and a subsequent boundary-aware 4-class refinement with a J-regularized cross-entropy objective improves training stability and substantially boosts instance-level separation as reflected by SEG.

Despite these improvements, our pipeline has limitations. Instance reconstruction still relies on connected-component labeling, which can be sensitive to small holes, spurious gaps, or noisy predictions: slight errors may split a single cell into multiple components, while missed separators can still merge neighboring cells. The 4-class target construction also uses a fixed ring/interface width, which may not be optimal across varying cell sizes and densities. Finally, since CTC test labels are not publicly available, our evaluation is limited to a validation split of the training sequences and may not fully capture generalization.

With more time, we would explore adaptive or learned boundary representations (e.g., distance-transform targets or variable ring widths), add stronger data augmentation and incorporate temporal context from neighboring frames to improve boundary consistency over time, and replace simple connected components with a more robust instance reconstruction step (e.g., watershed seeded by learned centers or embedding-based clustering). We would also submit to the official CTC evaluation server to obtain fully standardized SEG/DET scores on the hidden test set for more direct comparison with prior work.

References

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. URL https://arxiv.org/abs/1512.03385.

Martin Maška, Vladimír Ulman, David Svoboda, Pavel Matula, Cristina Ederra, Ainhoa Urbiola, T. Espaa, Subramanian Venkatesan, Deepak Balak, Pavel Karas, T. Bolckov, S. Étreitov, Craig Carthel, Stefano Coraluppi, Nathalie Harder, Karl Rohr, Klas Magnusson, J. Jaldn, Helen Blau, and Carlos Ortiz-de Solorzano. A benchmark for comparison of cell tracking algorithms. *Bioinformatics*, 30:1609–1617, 06 2014a. doi: 10.1093/bioinformatics/btu080.

Martin Maška, Vladimír Ulman, David Svoboda, Pavel Matula, Petr Matula, Cristina Ederra, Ainhoa Urbiola, Tomás España, Subramanian Venkatesan, Deepak M.W. Balak, Pavel Karas, Tereza Bolcková, Markéta Štreitová, Craig Carthel, Stefano Coraluppi, Nathalie Harder, Karl Rohr, Klas E. G. Magnusson, Joakim Jaldén, Helen M. Blau, Oleh Dzyubachyk, Pavel Křížek, Guy M. Hagen, David Pastor-Escuredo, Daniel Jimenez-Carretero, Maria J. Ledesma-Carbayo, Arrate Muñoz-Barrutia, Erik Meijering, Michal Kozubek, and Carlos Ortiz-de Solorzano. A benchmark for comparison of cell tracking algorithms. *Bioinformatics*, 30(11):1609–1617, 02 2014b. ISSN 1367-4803. doi: 10.1093/bioinformatics/btu080. URL https://doi.org/10.1093/bioinformatics/btu080.

Fidel A. Guerrero Peña, Pedro D. Marrero Fernandez, Paul T. Tarr, Tsang Ing Ren, Elliot M. Meyerowitz, and Alexandre Cunha. J regularization improves imbalanced multiclass segmentation, 2019. URL https://arxiv.org/abs/1910.09783.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. URL https://arxiv.org/abs/1505.04597.

Carsen Stringer, Tim Wang, Michalis Michaelos, and Marius Pachitariu. Cellpose: a generalist algorithm for cellular segmentation. *bioRxiv*, 2020. doi: 10.1101/2020.02.02.931238. URL https://www.biorxiv.org/content/early/2020/04/01/2020.02.02.931238.