

Biological Image Computing Final Project

Jolie Abadir, Yuki Ogawa, Serena Pen, Simai Wang

1. Statement of Need

Provide a detailed description of the problem your code is designed to solve. For assigned projects, summarize the key challenges that must be addressed to achieve the project's objectives.

The goal of this project is to build an image analysis workflow to detect Terosyne Hydroxylase-stained cells in a microscopy dataset in terms of the populations' intensity, morphology, and spatial organization. The dataset we used is TH_mRNA_xy2pt34_z5.tif, uploaded in Courseworks under Demo.

The problem is to identify individual TH-positive cells in a fluorescence image stack and quantify biologically meaningful features for each cell. The main challenges include high background noise from tissue autofluorescences, variable cell brightness across the image, overlapping cells in the 2D projection of a 3D image, and the lack of cell boundaries in staining. Further, the highly saturated central region interferes with normalization across the full image for segmentation.

2. Dependencies and Installation Instructions

List the modules and packages required for your project. If additional installations are necessary, provide clear, step-by-step instructions. Ensure that users can easily set up the required environment with minimal effort.

Our project requires os, skimage.io, scipy.ndimage, matplotlib, skimage, numpy, stackview, and cellpose.

3. Developer Documentation

3.1 Core Functionality

Describe the main functionality of your code. Explain how it works, the key algorithms used, and the overall structure of the program. Provide a step-by-step overview of the workflow, from data loading to final output.

step	description
Load image	The stack is read with skimage.io imread from Google Drive.
Project stack	A maximum-intensity projection is created along the z-axis to emphasize bright

	TH-positive cell-like objects.
Preprocess projection	Gaussian smoothing and median filtering were tested.
Segment cells	Cellulose is applied to the median filtered projection. Parameters tested.
Measure regions	Skimage.measure.regionprops is used with the max projection as the intensity image.
Measure spatial organization	Centroids are converted to microns and KDTree is used to query each cell's nearest neighboring cell.

3.2 Algorithms Used

Specify the image processing algorithms and techniques implemented in your project. Explain the purpose of each method, such as thresholding, Gaussian smoothing, or other relevant approaches.

Maximum-intensity projection: collapse the z-stack into a 2D image where each pixel is the brightest value across z. This makes bright cell signals easier to detect and segment.

Median filtering: reduces small-scale noise while preserving local cell-like structures better than heavy smoothing.

Cellpose segmentation: uses a pretrained deep learning segmentation model to predict cell masks from the preprocessed 2D image.

4. User Documentation

Explain how to run your code. Clearly outline input requirements, including file formats, color depth, and image dimensions. Describe any required parameters or configuration settings, such as file paths, threshold values, or user-defined options.

The pipeline can be run as the ipynb provided. The input file is provided as a .tif file, this path should be changed to point to where your image is. Other key parameters to configure to liking are the intensity normalization range, cellpose parameters like diameter, flow threshold, and cellprob threshold, and the size filtering of cellpose.

5. Example Runs / Results

Provide concrete examples demonstrating your code in action. Include sample input data and corresponding outputs to showcase performance and effectiveness. Use images or datasets where appropriate.

We ran the full pipeline on TH_mRNA_xy2pt34_z5.tif, a fluorescence z-stack with a pixel size of 2.34 μm . The stack was collapsed into a 2D image with maximum-intensity projection (figure 1) and lightly denoised with a median filter (radius 1) before being passed to Cellpose

Max Intensity Projection (z-axis)



[Figure 1: Maximum-intensity projection of TH_mRNA_xy2pt34_z5.tif along the z-axis.]

5.1 – Segmentation

Cellpose was run on the median-filtered projection using the pretrained CellposeModel on GPU, with diameter=3, flow_threshold=0.0, cellprob_threshold=-2.0, and intensity normalization clipped to [180, 700]. To remove segmentation noise, regions with an area outside [25, 125] pixels ($\approx 137\text{--}684\ \mu\text{m}^2$) were dropped, and the remaining regions were re-labeled with `scipy.ndimage.label`. After filtering, 486 cells remained for downstream analysis. A representative overlay of the filtered cell boundaries on the projection is shown in Figure 2. Most isolated bright cells are cleanly outlined, though touching cells along the dense midline are occasionally enclosed within a single contour, and some dim cells in lower-signal regions are missed.

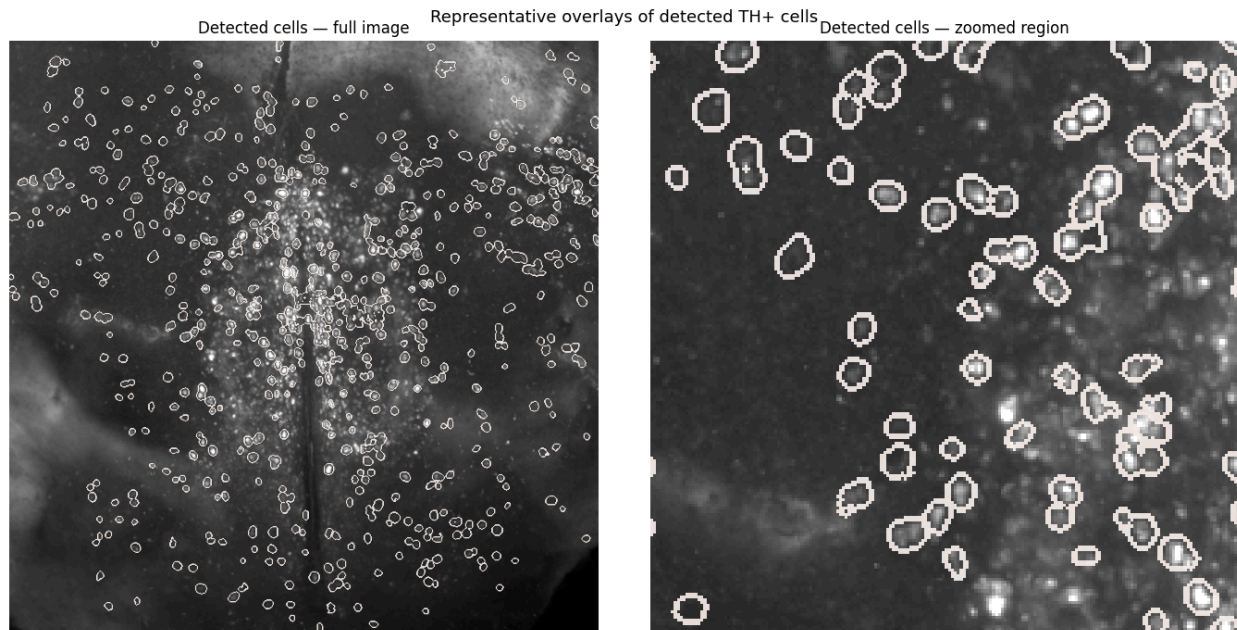
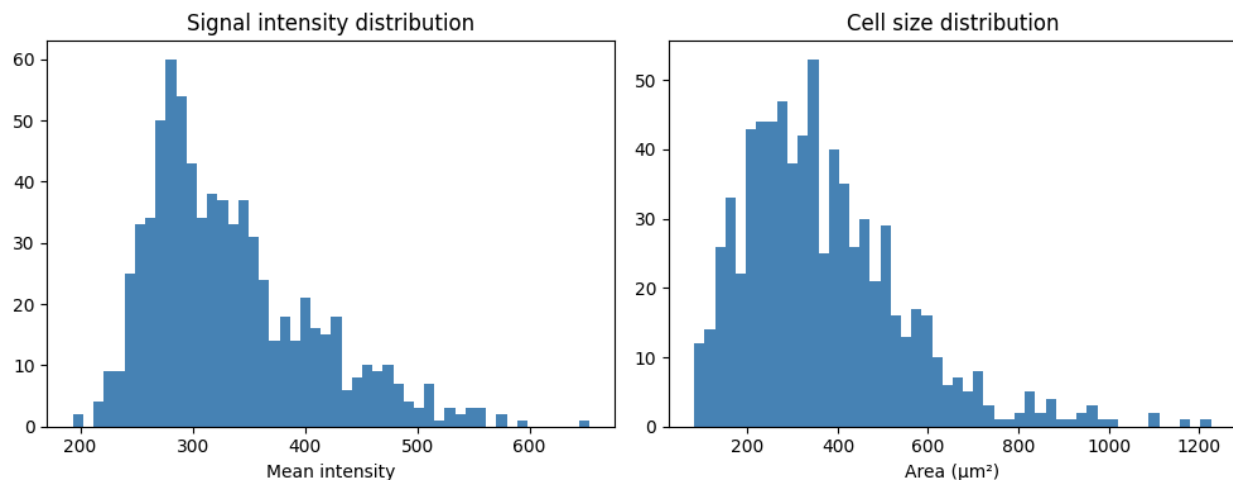


Figure 2. Representative overlays of detected TH-positive cells. Left: filtered Cellpose segmentation on the full max-intensity projection. Right: a zoomed-in region showing per-cell detail. Light contours show detected cell boundaries.

5.2 – Per-cell measurements

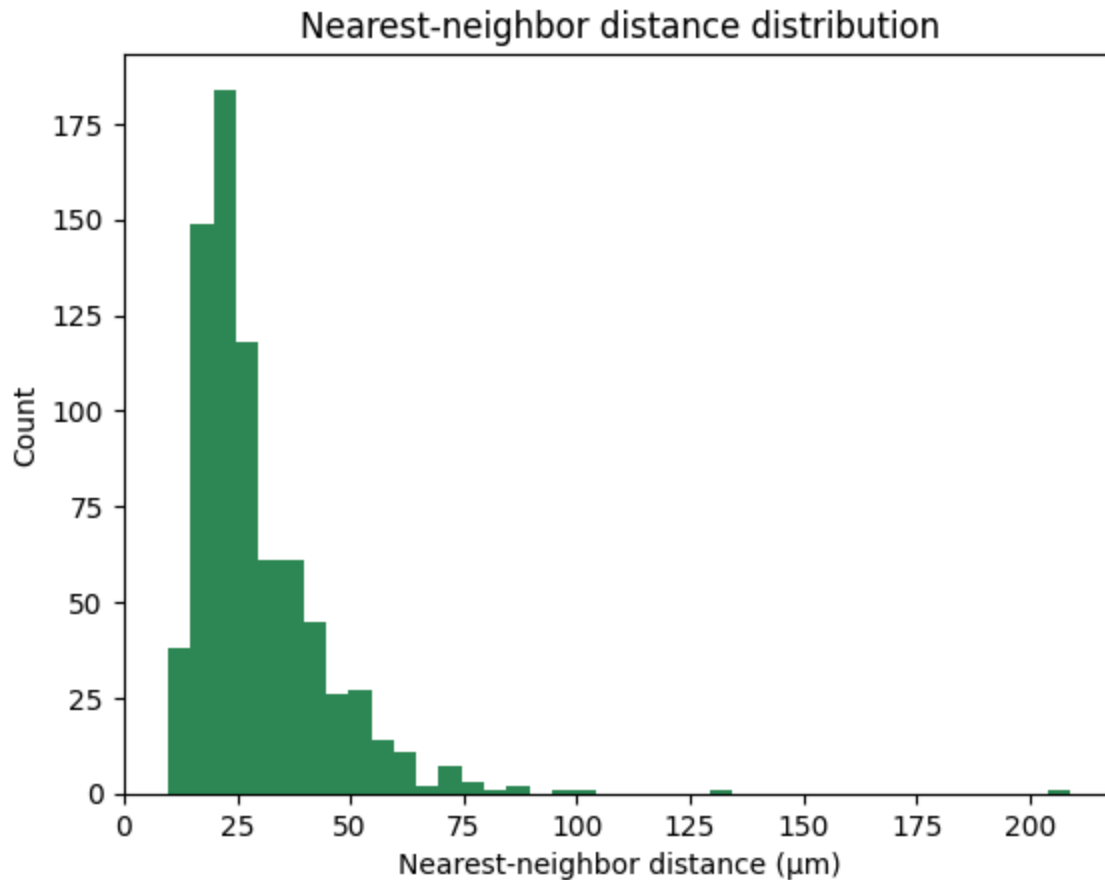
For each filtered region, `skimage.measure.regionprops` was used to extract mean intensity (over the max projection) and morphology features: area (μm^2), major and minor axis lengths (μm), eccentricity, and aspect ratio. Population distributions of signal intensity and cell area are shown in Figure 3. Mean intensity peaked around 280–300 (a.u.) with a long right tail of more strongly stained cells extending to ~ 650 , reflecting the wide range of TH expression levels across the population. The cell size distribution peaked at roughly 250–350 μm^2 , consistent with typical neuronal soma cross-sectional area, with a right-skewed tail extending to $\sim 1200 \mu\text{m}^2$. The tail corresponds to touching or overlapping cells that Cellpose merged into a single mask, consistent with the under-segmentation seen in Figure 2.



[Figure 3. Population distributions of mean signal intensity (left) and cell area in μm^2 (right) across detected TH-positive cells.]

5.3 Spatial organization

Centroids were converted from pixels to microns ($\times 2.34 \mu\text{m}/\text{px}$) and a KD-tree was used to find each cell's nearest neighbor ($k=2$, dropping the self-match). The distribution is shown in Figure 4 and is computed on the unfiltered Cellpose output ($n=753$ cells), prior to the size filter described in 5.1. Across the population, the mean nearest-neighbor distance was $29.7 \mu\text{m}$ (median $24.9 \mu\text{m}$), with a strong peak around $15\text{--}25 \mu\text{m}$ and a long right tail extending to $\sim 200 \mu\text{m}$. The peak indicates that most TH-positive cells are densely packed, with neighbors only one or two cell diameters away, while the tail reflects the smaller population of isolated cells in lower-density peripheral regions. This distribution provides a quantitative baseline for comparing clustering patterns across regions or experimental conditions.



[Figure 4. Distribution of nearest-neighbor distances (μm) among detected TH-positive cells. Centroids were converted to physical units ($\times 2.34 \mu\text{m}/\text{px}$) and the nearest other cell was found for each cell using a KD-tree query.]

5.4 – Validation against manual annotations

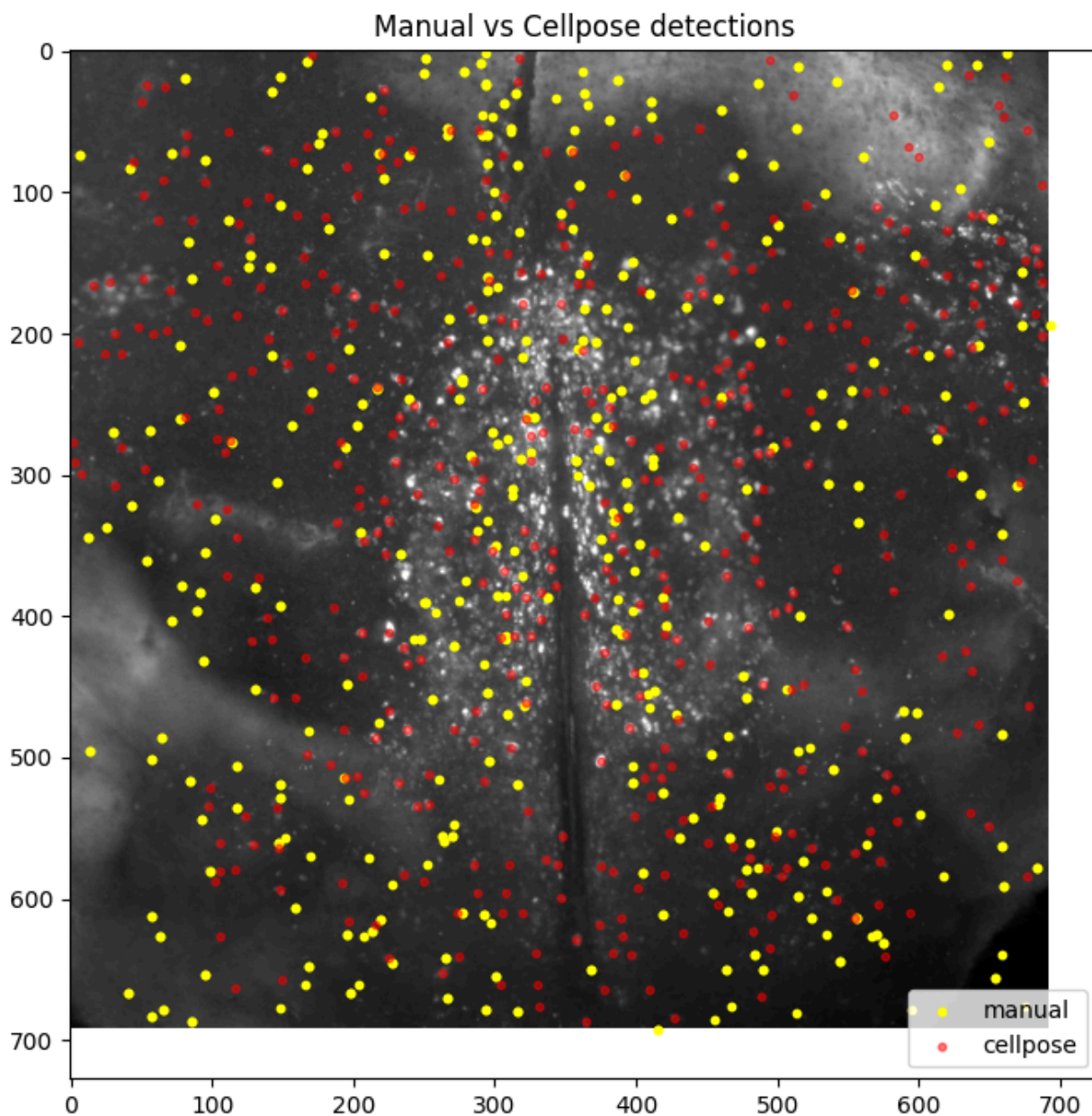
To evaluate detection accuracy, the 486 filtered Cellpose centroids were compared against 616 manual cell annotations made in Fiji on the same maximum-intensity projection used as the Cellpose input, ensuring an apples-to-apples comparison. A detection was counted as a true positive (TP) if it fell within 20 px of a manual annotation, with each annotation matched at most once via a greedy KD-tree assignment.

Manual annotations	Cellpose detections	TP	FP	FN	Precision	Recall	F1
616	486	194	292	422	0.399	0.315	0.352

Figure 5 overlays the manual annotations (yellow) and Cellpose detections (red) on the max-intensity projection. The pipeline correctly identifies a substantial subset of TH-positive cells, but recovers fewer than half of the manually annotated cells (recall 0.315), and roughly 60% of its detections do not match a manual annotation (precision 0.399). Inspection of the overlay reveals two recurring failure modes:

- **Merged neighbors (under-segmentation):** In the dense band of cells flanking the midline, touching cells are sometimes enclosed within a single contour, producing one detection where there should be two or three. This is a primary source of false negatives at high local cell density.
- **Missed cells in low-signal regions:** Cells with low staining intensity, or located in peripheral regions with diffuse background fluorescence, often fall below the Cellpose cellprob threshold and are not detected.

The relatively balanced precision and recall ($F1 = 0.352$) reflect the trade-off inherent in the current parameter choice: the moderate cellprob_threshold = -2.0 and the tight size filter [25, 125] px favor specificity at the cost of missing dim or merged cells. The validation provides a quantitative benchmark against which future parameter sweeps or model fine-tuning can be compared.



[Figure 5. Comparison of manual annotations and Cellpose detections on the maximum-intensity projection. Yellow points indicate the 616 manual cell annotations made in Fiji; red points indicate the 486 filtered Cellpose centroids. A detection was counted as a true positive if it fell within 20 px of a manual annotation.]

The full result from the notebook run is included as a separate PDF file named runRecord.

6. Summary and Discussion

Evaluate your code's performance, including strengths and limitations. Discuss results in terms of accuracy, efficiency, and scalability. Suggest possible improvements and future directions for the project.

This project produced an end-to-end image analysis pipeline that takes a fluorescence z-stack of TH-stained tissue and returns a per-cell quantitative description of the population in terms of staining intensity, morphology, and spatial organization. The pipeline is straightforward to run, requires no manual segmentation, and outputs measurements in physical units (μm , μm^2) suitable for downstream comparison across regions or experimental conditions.

Strengths. The pipeline is fully automated from raw stack to per-cell measurements, requires only a small set of user-tunable parameters, and runs in a few minutes per image on a Colab GPU. Outputs are reported in physical rather than pixel units, making results portable across datasets with different acquisition settings. The use of regionprops and a KD-tree provides access to a rich set of features (intensity, area, axis lengths, eccentricity, and nearest-neighbor distance) from a single segmentation pass. Validation against manual annotations on the same maximum-intensity projection provides an honest quantitative benchmark, and the failure modes uncovered by that validation are visually interpretable in Figure 5.

Limitations. Quantitative validation against 616 manual annotations yielded precision 0.399, recall 0.315, and F1 0.352, indicating that Cellpose segmentation is the dominant source of error in the pipeline. Two failure modes account for most of the gap. First, under-segmentation along the dense midline merges adjacent cells into single masks, inflating cell area and depressing the count where cells are most clustered. Second, dim cells in peripheral, low-signal regions fall below the cellprob threshold and are missed entirely. The current parameter choice (diameter=3, cellprob_threshold=-2.0, size filter [25, 125] px) reflects a tradeoff between these two failure modes: a more permissive threshold would recover more dim cells but increase false positives in the diffuse background, while a stricter threshold would further worsen recall. We also note that some Cellpose detections classified as false positives may correspond to real cells missed during manual annotation, so the reported precision should be interpreted as a lower bound.

A second limitation is that the segmentation operates on a maximum-intensity projection rather than the full z-stack. Collapsing the stack discards depth information, and causes cells at different z-planes to overlap in the projection, contributing directly to under-segmentation observed in dense regions. The size filter further trades precision for simplicity: the tight [25, 125] px window removes merged objects but also discards any genuinely large cells that exceed the upper bound.

Future directions. Several improvements would address the limitations above:

- **3D segmentation.** Running Cellpose directly on the z-stack (do_3D=True) rather than the projection would preserve z-resolution and reduce false negatives in densely packed

regions where cells overlap in the projection. This has been tested on a small slice of the image, however requires much longer GPU run time and memory usage.

- **Parameter sweep.** A systematic sweep over diameter, cellprob_threshold, flow_threshold, and the size filter bounds, scored against the manual annotations, would identify a more principled operating point than the current values, which were chosen empirically by inspecting segmentation quality.
- **Model fine-tuning.** The pretrained Cellpose model was trained on a broad set of cell types. Fine-tuning on a small set of manually annotated TH-stained images would likely improve both precision and recall on this specific staining modality.
- **Generalization.** The current pipeline is hard-coded to one input file. Refactoring the segmentation, measurement, and validation steps into reusable functions would make the pipeline easier to apply to additional datasets or to batch-process an entire experiment.

In its current form, the pipeline establishes a working baseline for quantitative analysis of TH-positive cell populations and provides a clear set of validated metrics against which future iterations can be measured.

7. References (if applicable)

Cite any relevant research papers, libraries, or resources used in your project. Acknowledge sources that contributed to your work.

This project has consulted heavily the lecture notes and relevant package documentation.