VIETNAM NATIONAL UNIVERSITY
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

COMPUTER VISION (CO3057)

CC01 - Assignment Report

Cellpose-SAM - superhuman generalization for cellular segmentation

Supervisor: PhD Nguyen Duc Dung, CSE-HCMUT
Students: Le Thi Phuong Thao - 2252757
Nguyen Thuy Tien - 2252806

Ho Chi Minh City, December 2025

Contents

Abstract

Instance segmentation of biological cells remains challenging due to overlapping structures, weak boundaries, and large morphological variability. Traditional pixel-wise segmentation approaches often fail in densely packed scenes. Cellpose addresses this issue through a flow-field formulation, where each pixel predicts a vector pointing toward its instance center, enabling robust separation via geometric dynamics. Recently, foundation models such as the Segment Anything Model (SAM) have demonstrated strong generalization across visual domains, but are not designed for geometry-based instance separation.

In this work, we study and reimplement the technique based on the Cellpose-SAM framework. The first part will be the cellpose core flow-field segmentation algorithm, implement entirely from scratch, including flow generation, gradient-based pixel dynamics, and clustering. We further analyze the integration of SAM as a feature encoder for predicting Cellpose-style flow fields, following recent research. Our experiments demonstrate that while SAM provides rich representations, effective integration requires training a dedicated flow prediction head. The results highlight the complementary roles of representation learning and geometric reasoning in modern cell segmentation pipelines.

The source code for this implementation is available here: Computer Vision Assignment

1 Introduction

This work investigates whether Cellpose's flow-field formulation can be reproduced from first principles and how it may be integrated with SAM features to combine generalization and geometric robustness, following the methodology proposed by Pachitariu et al. [1]

1.1 Problem statement

Accurate instance segmentation of biological cells is difficult due to frequent contact between neighboring cells, heterogeneous morphology, and limited availability of annotated training data. While Cellpose addresses these challenges using a geometry-driven flow-field formulation, its performance depends on task-specific training and may not generalize well across imaging modalities. Conversely, foundation models such as SAM offer strong generalization but lack mechanisms for instance-level geometric separation. This work investigates whether Cellpose's flow-field formulation can be reproduced from first principles and how it may be integrated with SAM features to combine generalization and geometric robustness.

1.2 Cell Morphology

Cells are the fundamental units of biological organization, serving as the basic building blocks of all living organisms. Cell morphology, which describes the shape, size and structure of cells is crucial for understanding the types and states of cells and how cells respond to their micro-environments. The ability to identify and analyze individual cells is essential for understanding biological processes, disease mechanisms and responses to therapeutic interventions.

1.3 Cell Segmentation

Cell segmentation is a computational process used to identify and separate individual cells within biomedical images. Given a raw microscopy image containing multiple cells, the goal is to assign each pixel to either background or a specific cell instance, where each cell receives a unique label. This is fundamentally an instance segmentation problem in the biological imaging domain, requiring the algorithm not only to distinguish cells from background but also to separate each cell as a distinct entity.

1.4 Applications of Cell Segmentation

Cell segmentation enables researchers to perform critical analyze including:

- Count cell populations to assess drug responses or growth rates;
- Measure morphological features such as size, shape and structural characteristics of individual cells;
- Track cells across time-lapse sequences to study migration and development;
- Quantify protein or gene expression at the single-cell level;
- Classify cells into different types for diagnostic or research purposes.

Thus, an accurate computational method for cell segmentation is of paramount importance for appropriately handling biomedical images and extracting insights from the data.

1.5 Challenges in Instance Segmentation of Cells

Cell segmentation presents several significant challenges that make it a difficult computational problem:

- **Cell overlap and contact:** Cells frequently touch or overlap in biological samples, making it difficult to determine clear boundaries between individual cells.
- **Variability across imaging modalities:** Microscopy images differ dramatically depending on whether they are acquired using fluorescence microscopy, histopathology staining, phase-contrast imaging, or other technologies.
- **Image quality limitations:** Real-world microscopy data often contain noise, blur, low contrast, and various imaging artifacts.
- **Variation across tissue types and experiments:** The performance of segmentation methods can change substantially across different tissues and imaging technologies.

- **Diverse cell morphologies:** Cells may be round, elongated, or highly irregular depending on cell type, organism, and experimental conditions. Algorithms often perform better on larger or rounder cells, while smaller or irregular cells remain challenging.
- **Common segmentation failure modes:** Underperforming methods may exhibit oversegmentation (splitting one cell into multiple segments), undersegmentation (merging multiple cells into one), missing cells (failing to detect cells), or false positives (identifying non-cellular structures as cells).

These challenges highlight the need for robust computational methods that can generalize across diverse imaging conditions and cell types.

2 Survey of existing techniques

2.1 U-Net and CNN-based Cell Segmentation

Since its introduction, the U-Net architecture [2] has established itself as the baseline for biomedical image segmentation. As summarized in Table 1, U-Net is a Fully Convolutional Network (FCN) based on an encoder-decoder architecture with skip connections, designed to recover high-resolution spatial details from compressed semantic features.

Despite its widespread adoption, standard U-Net implementations perform *semantic segmentation* rather than instance segmentation. This pixel-wise classification approach assigns a binary label (foreground or background) to each pixel, which presents fundamental limitations in dense microscopy environments:

- **The Touching Cell Problem:** Traditional semantic segmentation predicts only binary masks. Consequently, when two distinct cells share a boundary, the pixels between them are classified uniformly as foreground, causing the model to merge multiple instances into a single object. This inability to distinguish boundaries between touching cells is a primary failure mode in crowded tissues.
- **Dependency on Post-Processing:** To mitigate the merging issue, U-Net outputs typically require heuristic post-processing algorithms (e.g., Watershed) to separate instances. These methods often rely on parameter tuning and can lead to errors such as over-segmentation or under-segmentation.
- **Limited Generalization:** Unlike modern foundation models, standard U-Net backbones are often trained from scratch on specific datasets. With limited parameter capacity (approximately 30M) and a lack of large-scale pretraining, they often struggle to generalize to unseen imaging modalities without retraining.

These limitations necessitate the shift toward geometry-based approaches like Cellpose, which replaces binary classification with flow-field prediction to explicitly model instance separation.

2.2 SAM - Segment Anything Model Family

The Segment Anything Model (SAM) series represents Meta's progressive development of foundation models for visual segmentation.

SAM 1 (2023) established the foundation for **promptable image segmentation**, enabling users to segment any object via geometric prompts (points, boxes, masks) using a ViT-MAE encoder trained on the SA-1B dataset (11M images, 1.1B masks). However, it was limited to static images with no temporal understanding.

SAM 2 (2024) extended these capabilities to *video segmentation* by introducing a streaming memory architecture with a Hiera backbone, memory encoder, and memory bank, enabling real-time object tracking across frames. Despite these advances, both models required per-object visual prompts and could not automatically segment all instances of a semantic concept.

2.2.1 Core Architecture and Design Philosophy

The Segment Anything Model (SAM) follows a modular architecture composed of three main components: a powerful image encoder based on the Vision Transformer (ViT), a flexible prompt encoder that processes various input types (e.g., points, bounding boxes, masks, or text), and a lightweight mask decoder that generates segmentation outputs. This design enables *promptable segmentation*, allowing users to specify segmentation targets through intuitive interactions.

The image encoder, typically instantiated as a ViT-Large or ViT-Huge variant, serves as the backbone for extracting rich visual representations. Pretrained using large-scale self-supervised techniques such as Masked Autoencoders (MAE), the encoder learns generalizable features that transfer effectively across diverse visual domains. This component accounts for the majority of the model parameters (approximately 300–600 million, depending on the variant) as well as most of the computational cost.

2.2.2 Training Data and Generalization Capability

SAM models are trained on massive datasets comprising millions of images with billions of mask annotations. Notably, the SA-1B dataset contains over 11 million images annotated with approximately 1.1 billion masks, collected using a sophisticated data engine that combines manual annotation with model-assisted labeling. The unprecedented scale of this dataset enables strong zero-shot generalization across a wide range of visual domains, from natural images to specialized fields such as medical imaging.

The SAM family has progressively expanded its capabilities, evolving from static image segmentation to video object tracking through temporal memory mechanisms, and more recently to semantic concept detection

using text or exemplar prompts. Each generation addresses limitations of its predecessors while maintaining backward compatibility with existing prompting paradigms.

2.2.3 Strengths of the SAM Family

The SAM family exhibits several key strengths:

- **Rich visual representations**: Large-scale pretraining yields features that capture both low-level textures and high-level semantics, facilitating transfer to specialized domains.
- **Zero-shot generalization**: The models can segment objects in previously unseen domains without fine-tuning, reducing reliance on domain-specific annotated data.
- **Flexible prompting**: Support for multiple prompt types (points, boxes, masks, and text) enables diverse interaction paradigms tailored to different applications.
- **Modular architecture**: The clear separation between the encoder, prompt processor, and decoder allows selective replacement or fine-tuning of components for task-specific adaptations.

2.2.4 Limitations for Cellular Segmentation

Despite their strong general-purpose performance, SAM models face several challenges when applied to cellular segmentation:

- **Prompt dependency**: Standard SAM requires per-instance prompts, making it impractical for images containing hundreds or thousands of cells that require fully automatic segmentation.
- **Instance separation**: SAM lacks explicit geometry-based mechanisms for instance separation. When cells touch or overlap, the model struggles to delineate boundaries without individual prompts.
- **Domain gap**: Although pretrained on diverse natural images, SAM's training data contains limited microscopy imagery. Fine-grained cellular structures and specialized imaging modalities (e.g., fluorescence or phase-contrast microscopy) are therefore underrepresented.
- **Lack of topological guarantees**: SAM's mask predictions do not enforce topological consistency, which can lead to implausible mask shapes or unintended merging of adjacent cells.

2.3 Cellpose

Many biological applications require accurate segmentation of cell bodies, membranes, and nuclei from microscopy images. While deep learning has enabled major advances in this area, most existing methods are specialized and rely on large, domain-specific training datasets. **Cellpose** addresses this limitation by providing a generalist, deep learning–based segmentation algorithm capable of segmenting a wide variety of image types *out of the box*, without the need for retraining or parameter tuning.

Quantitative cell biology often depends on simultaneous measurement of cellular properties such as shape, spatial organization, RNA expression, and protein expression. Assigning these properties to individual cells requires segmentation of the imaged volume into distinct cell bodies, commonly using a cytoplasmic or membrane marker. However, this task becomes especially challenging when cells are packed densely within tissues.

2.3.1 Core Architecture: Flow-Field Representation for Instance Segmentation

The key innovation of *Cellpose* is its vector flow representation, which converts cell masks into a continuous field that can be directly predicted by a neural network. This auxiliary representation is constructed by generating an energy function for each mask via the equilibrium solution of a heat-diffusion simulation, with a heat source placed at the mask's center. For a broad family of shapes, this energy function exhibits a single global maximum at the cell center.

Starting at any pixel inside a mask, the spatial gradient of this energy function points "uphill" toward the center. Running gradient ascent from all pixels therefore produces a mapping in which pixels converging to the same fixed point belong to the same cell. The horizontal and vertical gradients of the energy function constitute a reversible representation that the network can learn to predict.

2.3.2 Limitations of Cellpose

- **Limited Generalization:** Cellpose struggles with images visually distinct from training data. Performance drops significantly on "non-microscopy" and "microscopy:other" categories, scoring only 0.73 compared to 0.79 on familiar image types.
- **Small Training Dataset:** The model was trained on only ∼70,000 segmented objects from 616 images. This limited diversity constrains the model's ability to generalize to new cell types, tissues, and imaging modalities.
- **U-Net Backbone Constraints:** The U-Net architecture has limited capacity (∼30M parameters) and was trained from scratch on biological images only, without benefit of large-scale pretraining. This limits the "world knowledge" the model can encode.
- **Cell Size Estimation Dependency:** Cellpose requires estimating average cell size to resize images before processing. Errors in size estimation propagate to segmentation errors. The style-based size prediction achieves only 0.93–0.97 correlation with ground truth.
- **Sensitivity to Image Quality:** The original Cellpose lacks built-in robustness to noise, blur, varying contrast, and different channel orders. Later versions required separate image restoration models to handle degraded images.
- **Cannot Handle Overlapping Cells:** The flow field representation assumes each pixel belongs to only one cell. Truly overlapping or occluded cells cannot be segmented correctly.
- **Computational Overhead:** Achieving best performance requires multiple test-time enhancements: model ensembling (averaging 4 models), image tiling, and augmentation, significantly increasing computational cost.

Table 1: Comparison of Representative Cell Segmentation Methods

Method	**Core Idea**	**Segmentation Type**	**Handling Touching Cells**	**Generalization Capability**
U-Net	Encoder–decoder CNN with skip connections	Semantic (pixel-wise)	• Limited capability • Requires post-processing	• Limited capability • Task-specific training
Cellpose	Flow-field prediction toward instance centers	Instance segmentation	• Strong performance • Geometry-driven separation	• Moderate capability • Depends on training data
SAM	Foundation model with promptable region masks	Region-based segmentation	• Moderate performance • Struggles with dense instances	• Strong capability • Pretrained across domains

Table 1 summarizes the strengths and limitations of representative approaches to cell segmentation. Classical convolutional models such as U-Net perform well for semantic segmentation but struggle to separate touching instances. Cellpose addresses this challenge through a flow-field formulation that enables geometry-driven instance separation, yet its performance depends on task-specific training. Conversely, foundation models such as SAM demonstrate strong generalization across visual domains but lack mechanisms for precise instance-level separation in dense cellular scenes. Motivated by these observations, we focus on the flow-field representation underlying Cellpose and investigate its integration with SAM features to combine robust geometric reasoning with general-purpose visual representations.

3 Proposed Approach

"Modern algorithms for biological segmentation can match inter-human agreement in annotation quality. This however is not a performance bound: a hypothetical human-consensus segmentation could reduce error rates in half."

- **Previous ceiling**: If two human experts agree only 85% of the time, we thought 85% accuracy = perfect AI.
- **New insight**: A consensus of multiple humans would achieve 93% accuracy, because different humans make different mistakes that cancel out.
- **Goal**: Create a model that approaches this "human-consensus bound" rather than just matching individual human performance.

3.1 Flow Field Representation

The **flow field** is the key innovation of Cellpose, addressing the problem of representing cell masks in a form that neural networks can predict. The idea is that each pixel inside a cell has a vector pointing toward the cell's center.

To construct this flow field, Cellpose performs a heat-diffusion simulation with a heat source at the cell center, producing an energy function with a single maximum at that point. Horizontal and vertical gradients of this energy function define a reversible representation learnable by the network.

For simple shapes (e.g., circles or ovals), pixels point directly to the center. For more complex shapes, pixels at extreme positions first point toward intermediate pixels, which then point toward the center. Pixels outside cells are assigned zero gradients.

3.2 Why Flow Field Solves the Touching Cells Problem ?

Traditional semantic segmentation predicts only binary masks, which cannot distinguish boundaries between touching cells, leading to merged objects.

Flow fields solve this by preserving directional information: pixels of cell A point toward center A, and pixels of cell B point toward center B. Even when cells touch, gradient tracking causes pixels to converge to different centers, forming distinct clusters—each corresponding to one cell instance.

3.3 Gradient Tracking Algorithm

The conversion from flow predictions to instance masks proceeds as follows:

1. Threshold the cell probability map at 0.5 to identify cell pixels.
2. For each such pixel, iteratively update its position for 200 steps by moving it along the vector field (dX, dY).
3. Pixels that converge to the same fixed point are clustered together.

Each cluster corresponds to one cell instance.

3.4 Why do we need Cellpose-SAM ?

- The original `Cellpose` model uses a U-Net backbone with ∼30M parameters trained on only 70,000 biological objects. This limited dataset restricts its "world knowledge," leading to poor generalization to unseen image types, high sensitivity to imaging variations, and reliance on accurate cell-size estimation.
- `SAM` was pretrained on 11M images with over 1B masks, giving it highly generalizable visual features. However, it requires user prompts (e.g., points or boxes) to identify objects, making it unsuitable for automatic cell instance segmentation.
- `Cellpose-SAM` combines the strengths of both models by using SAM's ViT-L encoder for rich pretrained visual features while integrating Cellpose's flow-field decoder to enable automatic segmentation without prompts.
- The large pretrained encoder helps the model recognize consistent cell-boundary patterns and ignore inconsistent annotation noise. It can revert to pretrained inductive biases to learn robust and generalizable structural features.

- The flow-field representation enforces topological consistency, preventing errors such as merging nearby cells or producing impossible mask shapes.
- This combination allows `Cellpose-SAM` to achieve superhuman generalization, surpassing inter-human agreement and approaching the human-consensus limit of annotation quality.

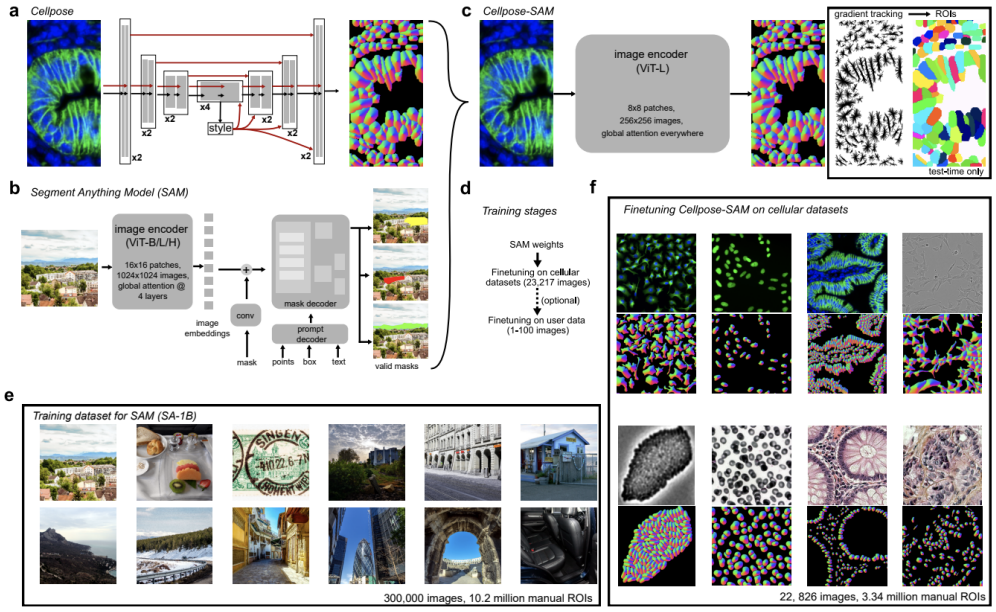


Figure 1: Defining and training the Cellpose-SAM model

3.5 Cellpose-SAM Architecture

Overview Cellpose-SAM integrates complementary strengths from two established segmentation frameworks: the pretrained image encoder of the Segment Anything Model (SAM) and the flow-based segmentation formulation of Cellpose. The central design principle is to leverage SAM's strong image representation learning capability while relying on Cellpose's proven mechanism for producing dense and coherent segmentation masks. While SAM is highly effective at learning general-purpose visual features, its original architecture is not optimized for dense instance segmentation in microscopy images. In contrast, Cellpose excels at transforming image features into accurate cell masks through flow field prediction.

Integration of SAM and Cellpose The Cellpose-SAM architecture selectively incorporates components from both models.

From SAM, the Vision Transformer Large (ViT-L) image encoder is adopted. This encoder consists of 24 transformer blocks with an embedding dimension of 1024 and accounts for approximately 305 million parameters of SAM's total 312 million parameters. The encoder is initialized using pretrained weights from the SA-1B dataset, which comprises approximately 300,000 natural images annotated with 10.2 million regions of interest. This large-scale pretraining provides strong, generalizable visual representations.

Conversely, several SAM components are omitted. Specifically, the mask decoder and prompt decoder modules—which process point, box, or text prompts—are removed, along with the associated sequential mask generation mechanism. These components are unnecessary for fully automatic dense segmentation and would introduce significant computational overhead.

From Cellpose, the flow-based segmentation paradigm is retained. The model predicts spatial flow fields that indicate pixel-wise directions toward object centers, which are subsequently converted into instance masks via gradient-tracking post-processing. The original Cellpose training objectives are also preserved, including a mean squared error loss for flow prediction and a binary cross-entropy loss for cell probability estimation.

Architectural Modifications To adapt the SAM encoder for cellular microscopy data, several architectural modifications are introduced. The input image resolution is reduced from 1024×1024 pixels to 256×256 pixels, reflecting the typical scale of cell images and reducing computational cost. The patch size is correspondingly decreased from 16×16 to 8×8 pixels, enabling finer-grained spatial modeling.

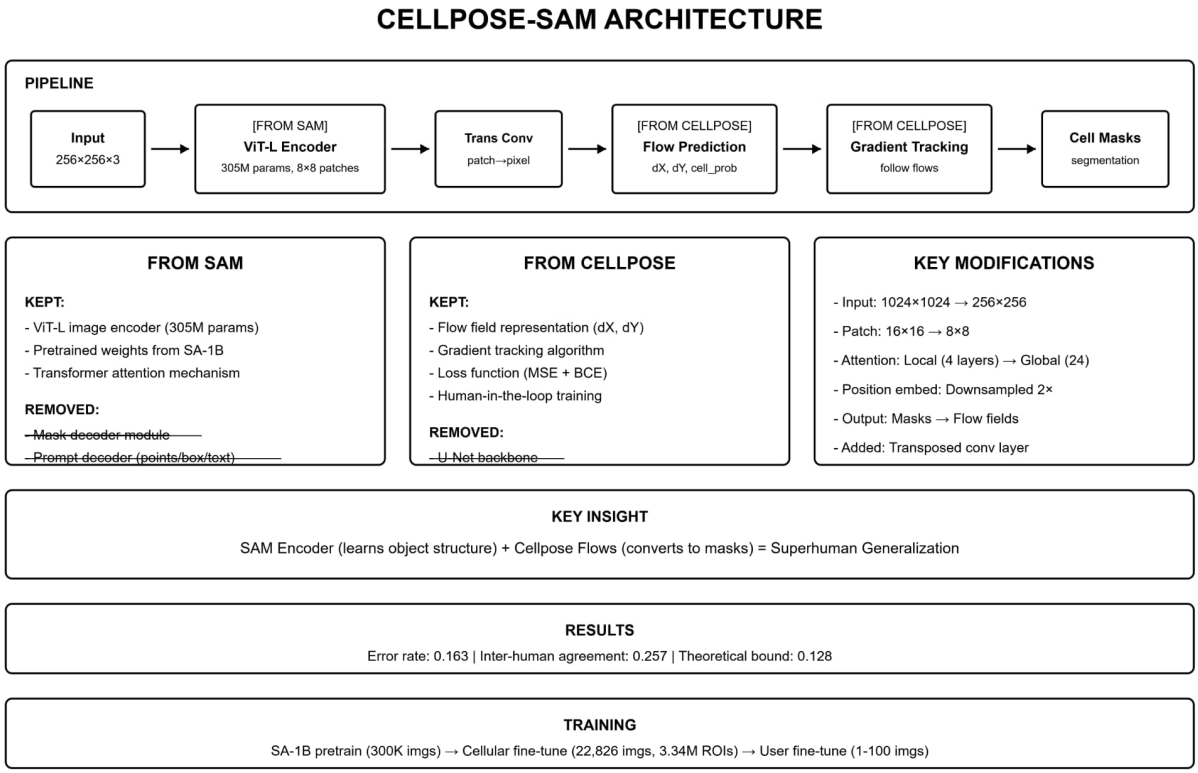


Figure 2: Cellpose-SAM Architecture

The attention mechanism is further modified by replacing the original hybrid attention scheme—where global attention is applied only at selected layers—with global attention across all transformer layers. This change improves long-range spatial reasoning, which is critical for capturing cell morphology. Positional embeddings are downsampled by a factor of two to remain consistent with the reduced input resolution. Finally, a transposed convolution layer is appended to the encoder output to project patch-level embeddings back into pixel space, enabling dense flow field prediction.

Summary Cellpose-SAM integrates the complementary strengths of two paradigms—SAM's large-scale pretrained visual representations and Cellpose's geometry-driven flow-field formulation—to achieve robust, fully automatic cellular instance segmentation. The key contributions of this integration are summarized as follows:

- **Combination of representation and geometry**: The framework leverages SAM's powerful image encoder, trained on large-scale data, together with Cellpose's flow-based instance separation, which is explicitly designed to handle touching and overlapping cells.
- **Removal of prompt dependency**: By replacing SAM's prompt-dependent mask decoder with a flow prediction head and gradient-tracking algorithm, Cellpose-SAM enables dense instance segmentation without requiring per-object prompts.
- **Architecture adaptations for microscopy**: Modifications such as reduced input resolution, smaller patch size, global attention, and transposed convolution for dense outputs ensure compatibility with cellular microscopy while preserving the representational power of the pretrained encoder.
- **Mitigation of parent-model limitations**: The proposed design overcomes SAM's inability to reliably separate touching instances and alleviates Cellpose's limited generalization arising from small-scale training.
- **Practical performance**: The resulting architecture delivers accurate instance segmentation across diverse 2D and 3D microscopy modalities while maintaining computational efficiency suitable for real-world analysis pipelines.

4 Experiments and Results

Since `Cellpose-SAM` is computationally heavy (1.23 GB model, requiring a GPU with at least 8 GB VRAM), we adopt a two-part approach:

- **Re-implement the Flow Field Algorithm from scratch**
  Demonstrate the core mechanism responsible for separating touching cells
- **Explain the Cellpose - SAM architecture and its integration**
  Show how SAM's encoder replaces the U-Net backbone in the combined `Cellpose-SAM` model

4.1 Re-implement the Flow Field Algorithm from scratch

4.1.1 Implementation Details

To realize the flow-field pipeline, we implemented three core functions corresponding to the stages of flow generation, pixel dynamics, and instance recovery.

- `labels_to_flows(label_mask)`
  - **Purpose:** Replaces the neural network prediction step for this experiment. It mathematically derives the "ideal" flow field from ground-truth masks by calculating normalized vectors pointing to the centroid of each cell.
  - **Input:** A binary or integer mask ($H \times W$) where each cell has a unique ID.
  - **Output:** A vector field ($H \times W \times 2$) containing dY, dX components and a cell probability map.
- `follow_flows(flow, n_iters=200)`
  - **Purpose:** Implements the Euler integration (gradient tracking) algorithm. It iteratively updates pixel positions based on the local flow vectors, simulating the "movement" of pixels toward cell centers.
  - **Input:** The vector field ($H \times W \times 2$) and number of iterations.
  - **Output:** An array of final coordinates ($H \times W \times 2$) representing where each pixel "landed" after the dynamics simulation.
- `cluster_final_positions(final_coords)`
  - **Purpose:** Recovers instance masks from the converged pixel locations. We use DBSCAN (Density-Based Spatial Clustering of Applications with Noise) to group pixels that converged to the same sink point.
  - **Input:** The final coordinate map from the dynamics step.
  - **Output:** The final predicted instance mask ($H \times W$).

Table 2: Core Functions Implemented for Flow-Field Algorithm

Function Name	**Purpose**	**Input**	**Output**
`labels_to_flows`	Derives ideal flow vectors (dX, dY) from ground truth to simulate network predictions.	Integer Label Mask ($H \times W$)	Flow Field ($H \times W \times 2$)
`follow_flows`	Simulates pixel dynamics via Euler integration (gradient tracking).	Flow Field	Final Pixel Coordinates
`cluster_positions`	Groups converged pixels into instances using DBSCAN.	Final Pixel Coordinates	Predicted Instance Mask

Altogether, those functions will be uses as in the procedure below:

Algorithm 1 Gradient Tracking and Mask Recovery

Require: Flow field $F \in R^{H \times W \times 2}$, Cell Probability $P \in R^{H \times W}$, Iterations N, Threshold τ
Ensure: Instance Mask $M \in Z^{H \times W}$
Initialize pixel positions: $X_0 \leftarrow$ grid coordinates (x, y)
Define active pixels: $S \leftarrow \{(x, y) \mid P(x, y) > \tau\}$
for $t = 1$ to N **do**
    Sample flow at current positions: $\vec{v} \leftarrow \text{BilinearInterpolate}(F, X_{t-1})$
    Update positions: $X_t \leftarrow X_{t-1} + \vec{v}$
    Clip X_t to image boundaries $[0, H] \times [0, W]$
end for
$X_{\text{final}} \leftarrow \text{Round}(X_N)$ restricted to set S
Cluster X_{final} using DBSCAN to assign instance IDs
Map IDs back to original grid to form M
return M

Where each symbol represents:

Table 3: Mathematical Notations used in Gradient Tracking

Symbol	Description	Dimension/Type
H, W	Height and Width of the input image	Scalars
F	Predicted Flow Field containing flow vectors (dY, dX) for every pixel	$R^{H \times W \times 2}$
P	Cell Probability Map (foreground confidence)	$R^{H \times W}$
M	Final Instance Mask where each cell has a unique integer ID	$Z^{H \times W}$
X_t	Matrix of pixel coordinates at iteration t	$R^{H \times W \times 2}$
S	Set of "active" foreground pixels	Subset of grid
τ	Threshold for determining foreground (usually 0.5)	Scalar $\in [0, 1]$
N	Number of dynamics iterations (steps)	Scalar (e.g., 200)
$\vec{v}$	Flow vector at a specific sampled position	Vector $\in R^2$

4.1.2 Evaluation Metrics

To quantify the performance of the flow-field algorithm, we utilized two standard metrics commonly applied in cellular segmentation tasks **Stringer2020cellposeStringer2022cellpose2**.

Intersection over Union (IoU) The IoU measures the geometric accuracy of a segmented mask relative to the ground truth. For a predicted segment P and a ground truth segment G, it is defined as:

$$\text{IoU}(P, G) = \frac{|P \cap G|}{|P \cup G|} \tag{1}$$

A match is considered valid (True Positive) only if IoU ≥ 0.5.

Average Precision (AP) While IoU evaluates individual object quality, AP measures the detection accuracy across the entire image. Following the protocol in the Cellpose paper, we calculate AP at an IoU threshold of 0.5 using the Critical Success Index (CSI) formulation:

$$\text{AP}_{50} = \frac{TP}{TP + FP + FN} \tag{2}$$

Where:
- **True Positives (TP):** Predicted masks that match a ground truth mask with IoU ≥ 0.5.
- **False Positives (FP):** Predicted masks with no matching ground truth (spurious detections).
- **False Negatives (FN):** Ground truth masks that were not matched by any prediction (missed cells).

This metric strictly penalizes both over-segmentation (high FP) and under-segmentation (high FN).

4.1.3 Experimental Results on Synthetic Benchmarks

To evaluate the robustness of the flow-field algorithm, we conducted a sensitivity analysis across seven synthetic scenarios (Ex 1–7). These scenarios were designed to isolate specific segmentation challenges: cell density ($N = 6$ to 30), geometric irregularity (circular vs. highly elongated ellipses), and occlusion levels (overlap factors 0.1 to 0.95).

Quantitative Performance Table 4 summarizes the Intersection-over-Union (IoU) performance across all scenarios. The algorithm demonstrated remarkable stability, achieving a mean IoU above 0.97 in 5 out of 7 cases.

Table 4: Sensitivity Analysis of Flow-Field Segmentation on Synthetic Scenarios

ID	Scenario Description	Density	Shape	Overlap	Mean IoU
Ex 1	Baseline (Mixed shapes)	12 cells	Mixed	Mod (0.6)	0.994
Ex 2	High Overlap Ellipses	10 cells	Ellipse	High (0.8)	**1.000**
Ex 3	Long Thin Cells (AR 4–6)	8 cells	Ellipse	Low (0.2)	0.987
Ex 4	Highly Elongated (AR 5–8)	10 cells	Ellipse	Mod (0.3)	0.980
Ex 5	High Density Circles	30 cells	Circular	Mod (0.7)	0.834
Ex 6	Sparse Line-like (AR 8–12)	6 cells	Linear	Low (0.1)	0.891
Ex 7	Extreme Overlap	10 cells	Mixed	**Max (0.95)**	0.976

Analysis of Key Factors Based on the quantitative data and visual inspections (Figure 4), we observed the following behaviors:

- **Impact of Geometry (Challenge Case):** Shape irregularity posed a moderate challenge. While the model handled standard ellipses well (Ex 2–4), performance degraded slightly for "line-like" cells with aspect ratios > 8 (**Example 6**, IoU 0.891). In these cases, the flow vectors along the major axis become nearly parallel, making convergence to the center slower and less precise.

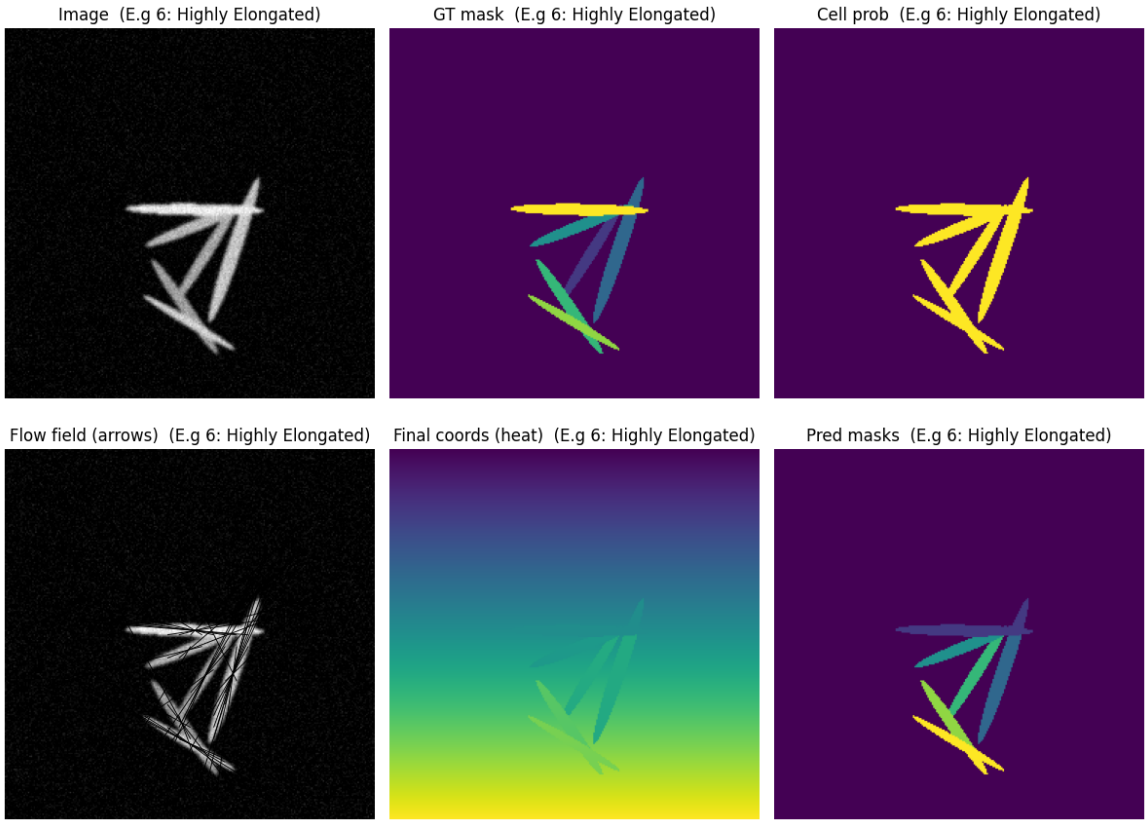


Figure 3: Result on line-like cells

- **Robustness to Overlap (Success Case):** Contrary to traditional watershed or binary segmentation methods, the flow-field approach excelled in high-occlusion scenarios. In **Example 7** (Top Row, Figure 4), despite an extreme overlap factor of 0.95, the algorithm achieved a near-perfect IoU of 0.976. The gradient vectors successfully directed pixels to their respective centroids even when cell boundaries were largely obscured.
- **Sensitivity to Density (Failure Case):** The primary failure mode was identified in high-density clusters rather than overlap. In **Example 5** (Bottom Row, Figure 4), the IoU dropped to 0.834 with 4 false negatives. This suggests that when centroids are packed too closely, the flow gradients interfere, causing the clustering step (DBSCAN) to merge adjacent instances.

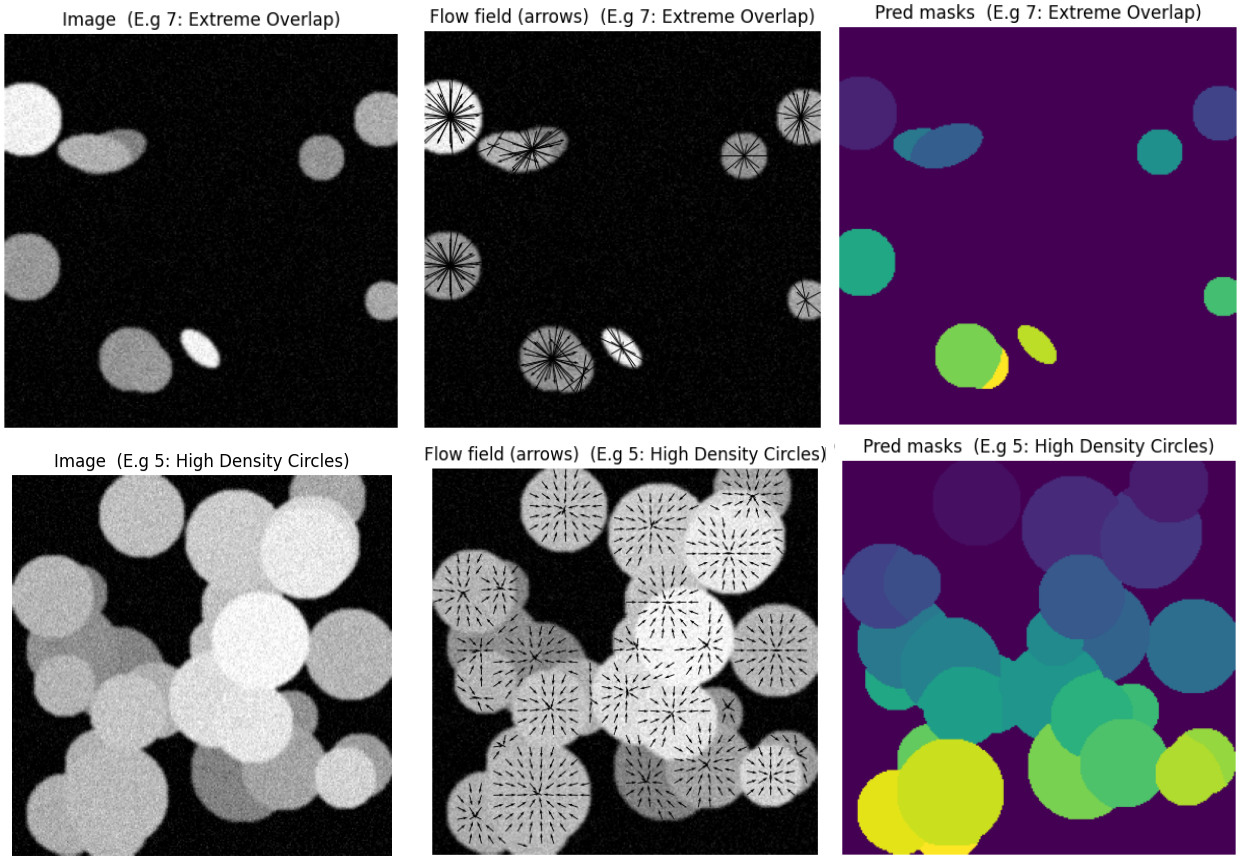


Figure 4: **Results on Synthetic Benchmarks. Top Row (Example 7):** The algorithm demonstrates robustness by successfully resolving 10 cells with extreme overlap (95%), achieving a Mean IoU of 0.976. **Bottom Row (Example 5):** A high-density scenario (30 cells) highlights limitations; the algorithm dropped 4 instances (AP=0.86) due to flow-field interference in crowded regions.

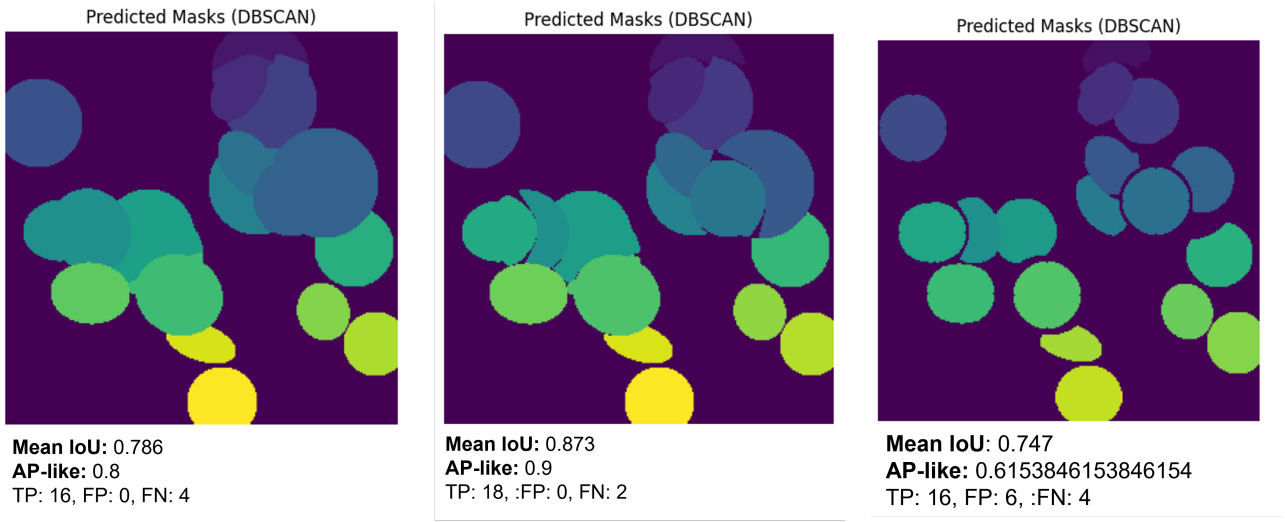
4.1.4 Parameter Sensitivity and Convergence Analysis

To understand the stability of the flow-field representation, we investigated the relationship between the temporal parameter (Euler iterations N) and the spatial clustering parameter (DBSCAN ϵ).

Impact of Clustering Parameters under Ambiguity

The DBSCAN parameter ϵ defines the maximum distance for two pixels to be considered neighbors. To isolate its effect, we conducted a stress test on a high-overlap scenario (20 cells) using significantly reduced flow iterations ($N = 20$). This setup prevented full pixel convergence, leaving the embeddings as diffuse "clouds" rather than tight points.

As visualized in Figure 5, under these ambiguous conditions, the segmentation quality is highly sensitive to ϵ.



a) High $\epsilon = 5.0$
(Under-segmentation)

b) Optimal $\epsilon = 1.5$
(Correct Separation)

c) Low $\epsilon = 0.5$
(Over-segmentation)

Figure 5: **Qualitative Impact of DBSCAN Epsilon.** We deliberately reduced flow iterations to $N = 20$ to create diffuse pixel clusters. This highlights the trade-off between merging cells (a) and fragmenting them (c) when flows have not fully converged.

Quantitative Validation Our metric analysis confirms the visual trade-offs observed in the stress test:

- **Under-segmentation ($\epsilon = 5.0$):** The loose threshold caused neighboring pixel clouds to merge. This resulted in **4 False Negatives (FN)**—where distinct ground truth cells were swallowed into larger super-pixels—dropping the Mean IoU to 0.786.
- **Over-segmentation ($\epsilon = 0.5$):** The strict threshold fractured single cells into multiple unconnected components. This caused a spike in **False Positives (FP=6)**, representing spurious artifacts, and yielded the lowest Mean IoU of 0.747.
- **Optimal Balance ($\epsilon = 1.5$):** This setting struck the correct balance, recovering 18 out of 20 cells (TP=18) and achieving the peak Mean IoU of 0.873.

Convergence and Parameter Decoupling
While ϵ controls spatial tolerance, the number of iterations (N) determines the spatial contraction of the embeddings. We analyzed the Mean IoU across a range of iterations (0 to 200) for varying ϵ values to test the system's robustness.

Figure 6 reveals a critical "decoupling effect":

- **Early Instability ($N < 50$):** At low iterations, the performance variance between different ϵ values is massive. A tight $\epsilon = 0.5$ fails completely (IoU ≈ 0.2), while looser thresholds perform better.
- **Convergence Stability ($N > 150$):** As the gradient tracking proceeds, pixels contract into extremely dense centroids. Crucially, at $N = 200$, the performance curves for $\epsilon \in [1.5, 5.0]$ all collapse onto the same maximum IoU line.

This confirms that sufficient computational depth (iterations) renders the hyperparameter ϵ largely irrelevant. This stability explains why the Cellpose framework generalizes well across different datasets without requiring the user to fine-tune clustering thresholds for every image.

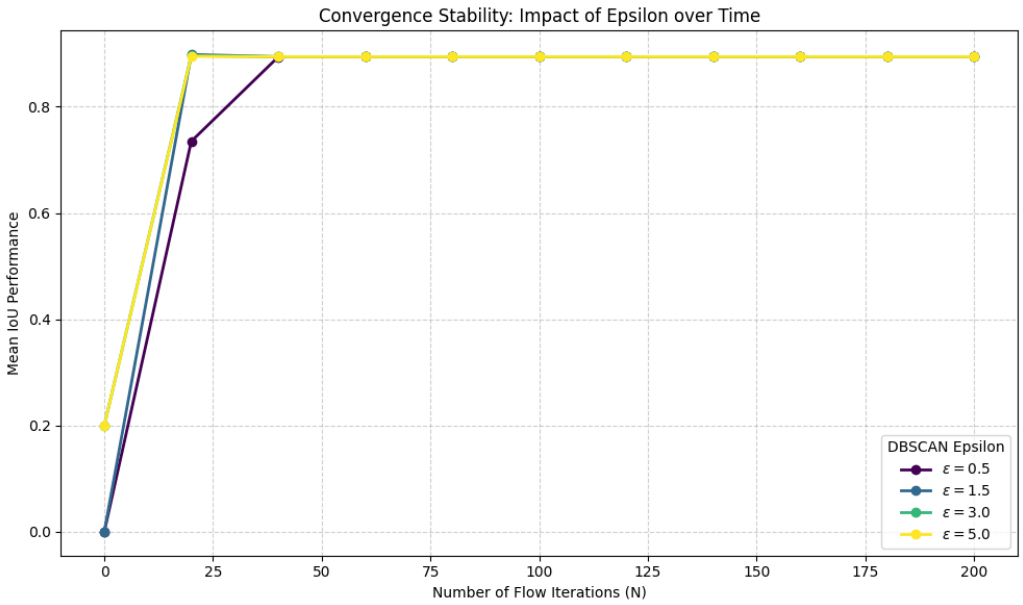


Figure 6: **Convergence Stability: Impact of Epsilon over Time.** At low iterations, segmentation quality is volatile and dependent on ϵ. However, as the flow field converges ($N > 150$), all ϵ curves (except the most extreme outliers) converge to optimal performance. This demonstrates that full gradient tracking provides robustness against hyperparameter variation.

4.2 Cellpose-SAM Integration

4.2.1 Feature Representation Mismatch Between SAM and Cellpose

Before introducing the fine-tuning strategy, we first analyze a direct integration of SAM and Cellpose without task-specific training. In this experiment, image embeddings extracted from a pretrained SAM are passed into a lightweight flow prediction head, followed by the standard Cellpose flow dynamics and clustering procedure. No alignment training between SAM features and Cellpose-style flow supervision is performed.
The motivation of this experiment is to evaluate whether SAM image embeddings, which are trained for generic segmentation tasks, can be directly reused to predict Cellpose-style flow fields without additional adaptation. The input images of this experiment are from real datasets, not from synthetic as above.

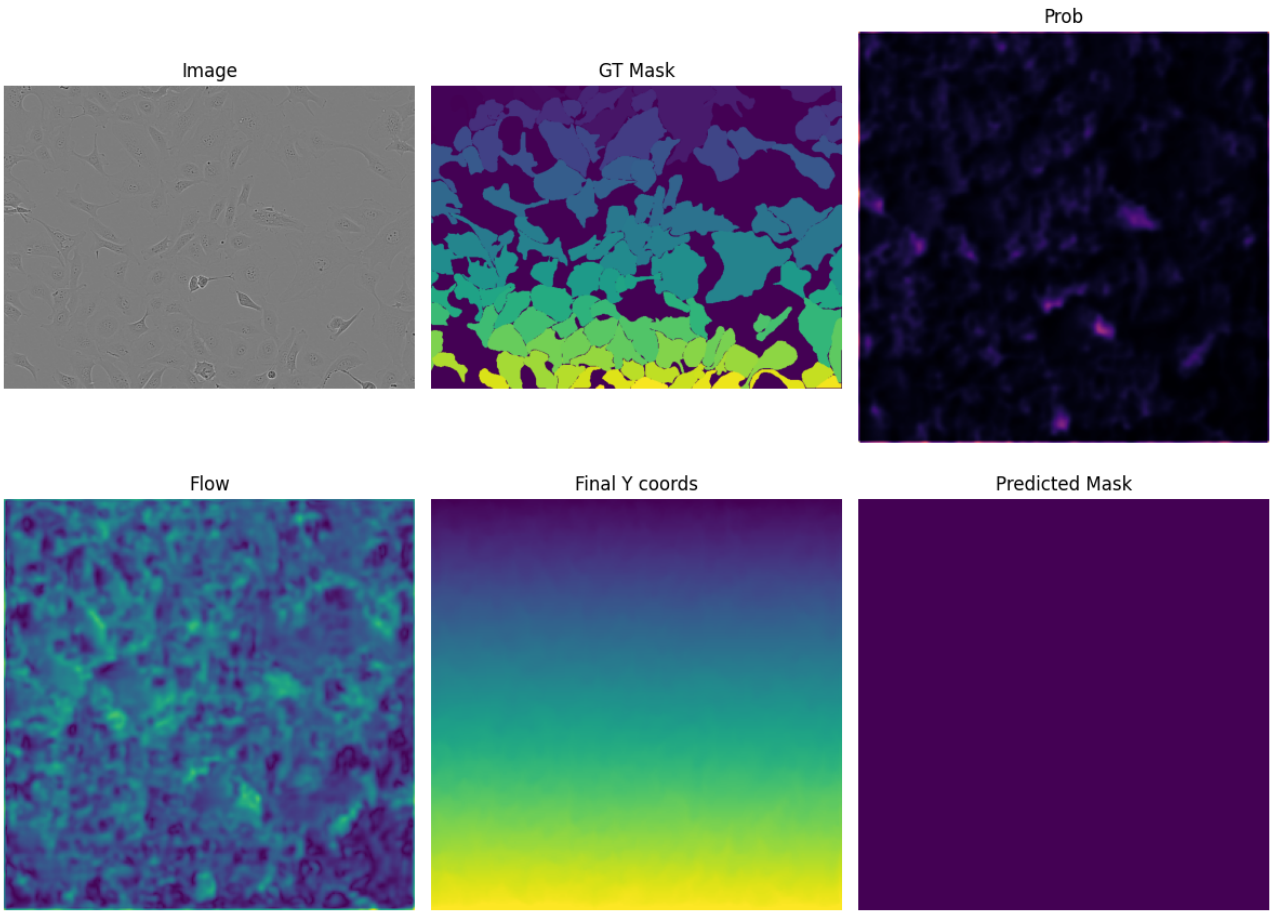


Figure 7: Cellpose-SAM pipeline result without alignment training

Observed Behavior

It is important to emphasize that the ground-truth masks visualized in this experiment originate from the Cellpose dataset and are not generated by SAM.

Despite using high-quality microscopy images and accurate ground-truth instance masks from the Cellpose dataset, the resulting probability maps, flow fields, and final instance segmentations are largely uninformative. Specifically, the predicted cell probability values remain close to zero, the flow fields lack coherent radial structure, and the final clustering stage fails to produce meaningful instance masks.

This behavior is not due to an implementation error, but rather reflects a fundamental representation mismatch between SAM and Cellpose. SAM is trained to produce high-level semantic embeddings that support prompt-based mask decoding, whereas Cellpose relies on dense, geometry-aware flow fields that encode pixel-wise direction vectors toward cell centers. These flow fields require precise spatial and morphological cues that are not explicitly represented in SAM embeddings.

Implication

This experiment demonstrates that SAM embeddings alone are insufficient to drive Cellpose-style instance segmentation without explicit alignment. Consequently, a dedicated fine-tuning stage is required to adapt the pretrained SAM representation to the geometric flow prediction task central to Cellpose.

This observation directly motivates the fine-tuning strategy described in the following subsection.

4.2.2 Cellpose-SAM Fine-tuning Pipeline (Small Data, Human-in-the-Loop)

Human-in-the-Loop Dataset

The `human_in_the_loop` dataset is a small collection of cell images designed to demonstrate efficient model fine-tuning under a human-in-the-loop paradigm. This dataset follows the methodology introduced in *Cellpose 2.0*, where instead of annotating images from scratch, users iteratively correct the segmentation errors produced by a pretrained model.

The dataset consists of a limited number of training and testing samples, summarized as follows:

- **Training set:** 5 images with manually corrected segmentation masks stored in `_seg.npy` format.
- **Test set:** 3 images, including samples such as *breast_vectra* from the TissueNet dataset, which contains multiplex immunofluorescence tissue images widely used in pathology research.
- **Image resolution:** All images are 256×256 pixels with 3 color channels.

The annotation workflow is iterative: the pretrained model is first applied to an image, its errors are corrected using the graphical user interface (GUI), the model is retrained, and the process is repeated on subsequent images. As training progresses, each iteration requires fewer manual corrections as the model adapts to the specific characteristics of the data. Prior studies have shown that this strategy reduces the annotation effort from approximately 500–1000 cells to only 100–200 cells while maintaining comparable segmentation accuracy.

Despite using only five carefully corrected training images, this dataset achieves an Average Precision of 76.1% on unseen test data. These results highlight the effectiveness of the human-in-the-loop approach and demonstrate that deep learning-based cell segmentation can be made accessible even in settings with very limited annotated data.

- **Overview:** This experiment fine-tunes a pretrained Cellpose-SAM model using a very small, human-annotated dataset (human-in-the-loop). The model is adapted using five manually labeled training images and evaluated on three held-out test images. After approximately 25 minutes of GPU training, the model achieves an Average Precision (AP@IoU=0.5) of 76.1%.
- **Environment Setup:** GPU availability is verified prior to training, as Cellpose-SAM contains approximately 100 MB of parameters. Training is conducted using PyTorch 2.7 with CUDA support, reducing training time from hours to minutes and enabling iterative human-in-the-loop experimentation.
- **Pretrained Model Initialization:** The Cellpose-SAM model is initialized from official pretrained weights trained on approximately 300,000 natural images and 23,000 cellular images spanning diverse microscopy modalities and cell morphologies. These pretrained weights provide strong general segmentation capabilities and serve as the foundation for fine-tuning.
- **Human-in-the-Loop Data Preparation:** The dataset consists of five training images with manually curated segmentation masks and three test images reserved exclusively for evaluation. Rather than learning segmentation masks directly, Cellpose converts annotations into flow fields using a heat diffusion process centered on each cell, improving robustness to irregular and concave cell shapes.
- **Fine-tuning with Small Data:** Fine-tuning is performed for 100 epochs using the AdamW optimizer with a learning rate of 10^{-5}. A weight decay of 0.1 is applied to mitigate overfitting, which is critical when training on only five images. The training loss decreases from 1.1469 to 0.9568 (approximately 17%) and stabilizes after epoch 50, indicating convergence.
- **Evaluation on Test Images:** The fine-tuned model is evaluated exclusively on three unseen test images. Performance is measured using Average Precision at an IoU threshold of 0.5, where predictions with IoU ≥ 0.5 are considered correct detections. The final result is AP = 0.761, corresponding to 76.1% of cells being correctly segmented.
- **Interpretation:** The results demonstrate that pretrained Cellpose-SAM can be effectively adapted using extremely limited, human-labeled data. The strong test performance confirms that transfer learning enables meaningful generalization in low-data regimes, making this approach suitable for rapid, human-in-the-loop customization in practical microscopy workflows.

Stage	Description
Pretrained Weights	Cellpose-SAM (cpsam) model initialized from pretrained weights (∼100 MB).
Small Data	5 training images with corresponding human-annotated masks (`_seg.npy`).
Fine-tuning	Training for 100 epochs using the AdamW optimizer with learning rate 10^{-5}, taking approximately 25 minutes on GPU.
Fine-tuned Model	The fine-tuned model is saved to `models/new_model`.
Evaluation	Performance evaluated on 3 test images using Average Precision at IoU = 0.5, achieving AP = 0.761 (76.1% of cells correctly segmented).

Table 5: Cellpose-SAM fine-tuning pipeline using pretrained weights and small human-annotated data

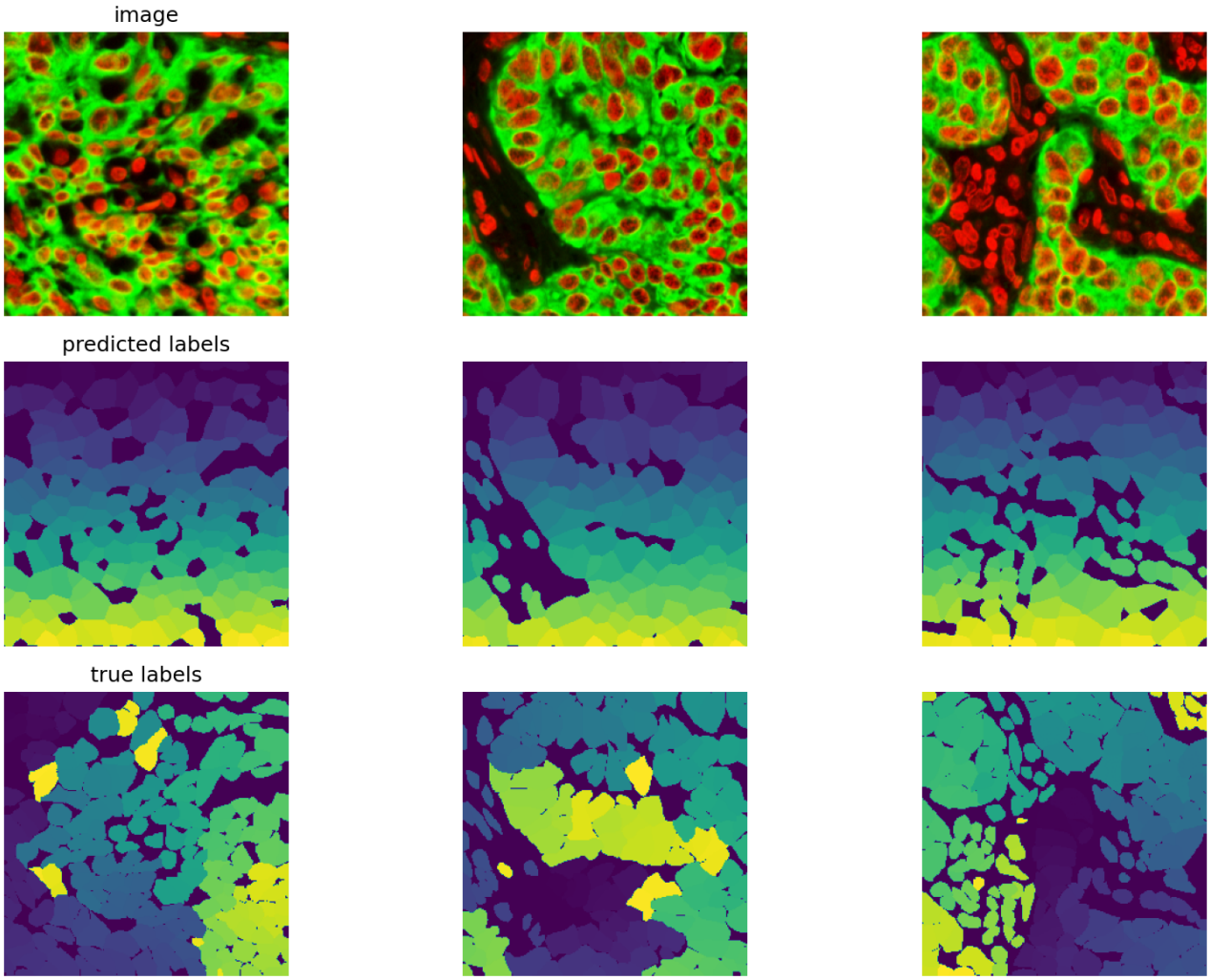


Figure 8: Cell Segmentation Output (Human_in_the_loop)

Metric	Value
Training images	5
Test images	3
Training epochs	100
Training time	∼25 minutes
Initial loss	1.1469
Final loss	0.9568
Loss reduction	16.6%
Average Precision @ IoU=0.5	0.761 (76.1%)

Table 6: Key Results Summary

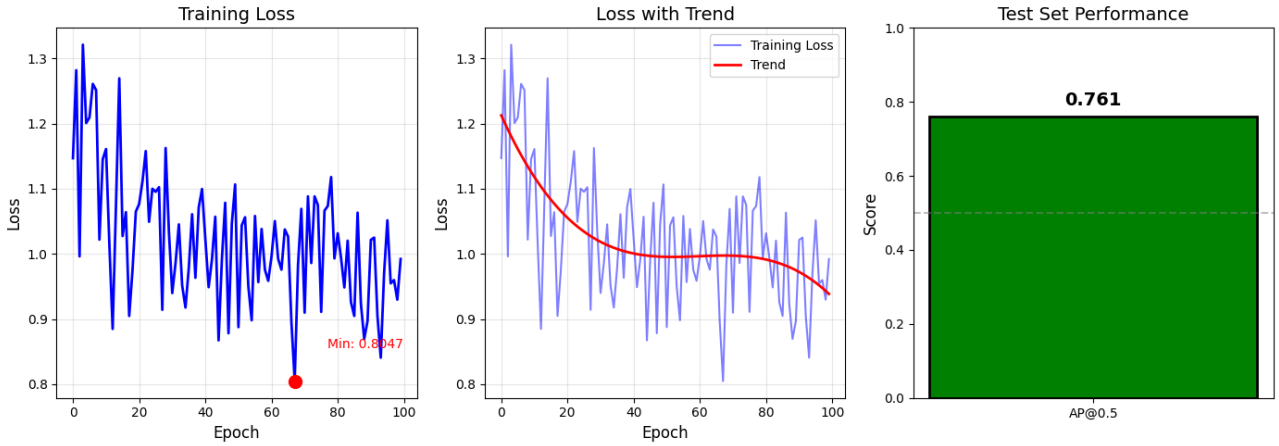


Figure 9: Cell Training Results Summary (Human$_i n_t he_l oop$)

4.2.3 Cellpose-SAM Demonstration Workflow

This section describes the execution workflow used to evaluate the Cellpose-SAM model on both two-dimensional (2D) and three-dimensional (3D) microscopy data. The workflow consists of installation, data preparation, segmentation, and result visualization stages.

Installation and Setup The Cellpose-SAM package was installed directly from the MouseLand GitHub repository. The model was initialized with GPU acceleration enabled, and the pretrained `cpsam` model (approximately 1.15 GB) was automatically downloaded to support inference.

Loading Example Data A set of example datasets was downloaded to evaluate the generalization capability of the model. This included 24 two-dimensional images collected from diverse sources such as Cellpose, Nuclei, TissueNet, Livecell, YeaZ, Omnipose, and DeepBacs. In addition, a three-dimensional RGB TIFF image was downloaded to facilitate testing on volumetric data.

Two-Dimensional Segmentation Two-dimensional segmentation was performed by applying the `model.eval()` function to all 24 images using 1000 inference iterations. The model produced predicted segmentation masks along with corresponding flow fields and style vectors for each image.

Visualization of 2D Results Segmentation performance was qualitatively assessed through visualization. Ground-truth masks were overlaid using solid purple contours, while predicted masks were indicated using dashed yellow contours, enabling direct visual comparison between reference annotations and model outputs.

Three-Dimensional Segmentation Three-dimensional segmentation was evaluated using two distinct approaches, as summarized in Table 7.

Table 7: Approaches for three-dimensional segmentation in Cellpose-SAM.

Method	Description
3D Flows (`do_3D=True`)	Computes flow fields from 2D slices across the YX, ZY, and ZX planes, combines them into a unified 3D flow representation, and subsequently generates volumetric segmentation masks.
Stitching (`stitch_threshold=0.5`)	Performs segmentation independently on each 2D slice and then stitches the resulting masks in three-dimensional space based on spatial overlap criteria.

Visualization of 3D Results The three-dimensional segmentation outputs were visualized by overlaying segmentation contours on selected Z-planes (0, 10, 20, . . . , 70), allowing qualitative assessment of mask consistency across depth.

Key Observations The experimental results indicate that Cellpose-SAM exhibits strong generalization performance across a wide range of microscopy image types, including both 2D and 3D data. While the stitching-based approach provides significantly faster inference (approximately 8 seconds compared to 35 seconds for 3D flows), the 3D flow-based method tends to produce more volumetrically consistent segmentations. Furthermore, the stitching strategy can be naturally extended to support cell tracking in time-lapse imaging scenarios.

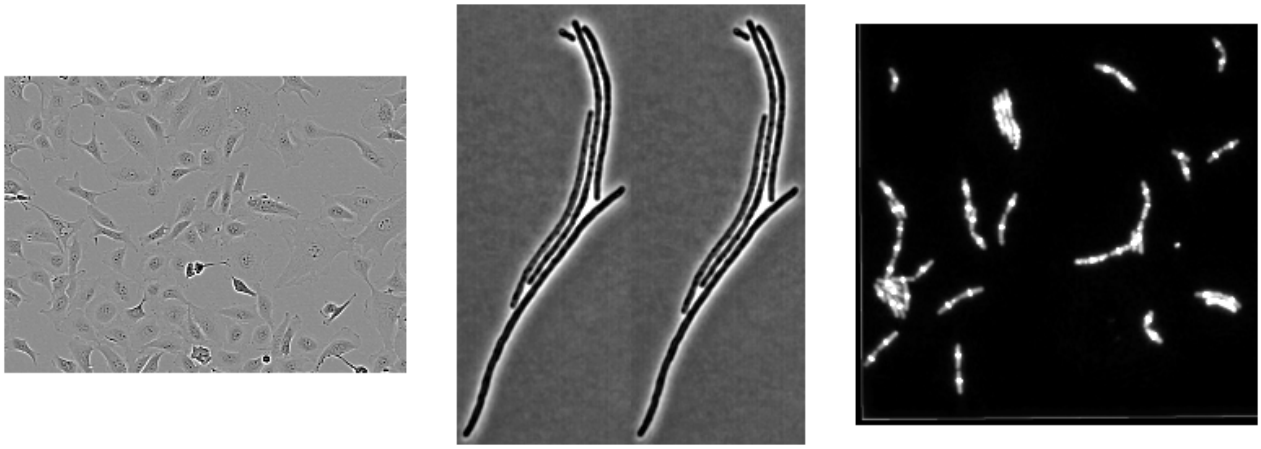
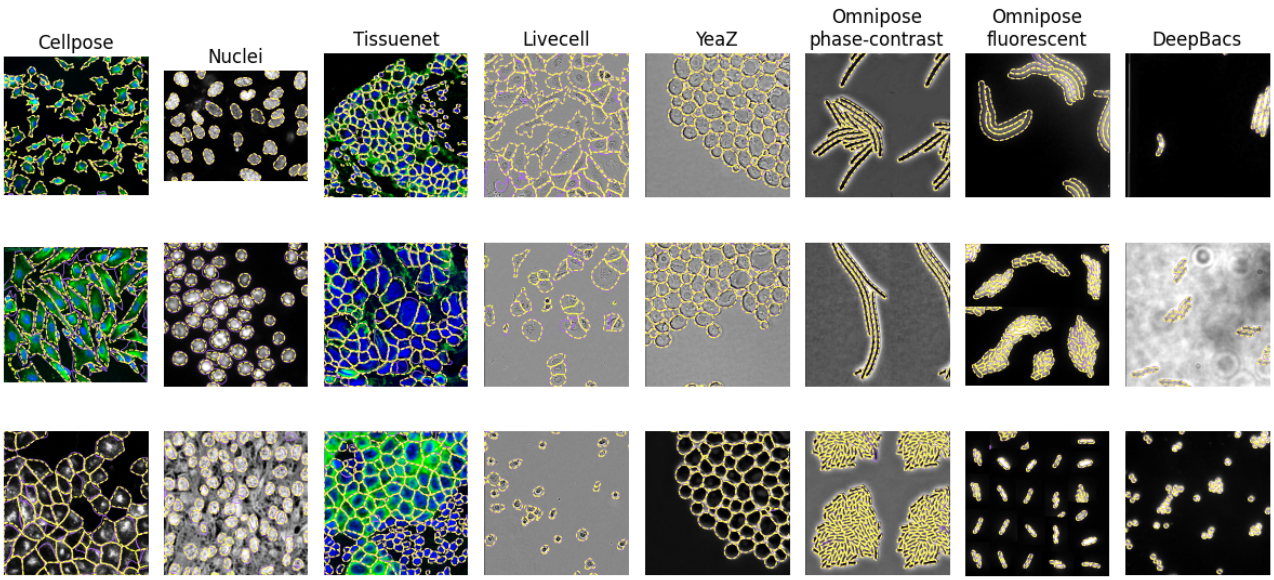
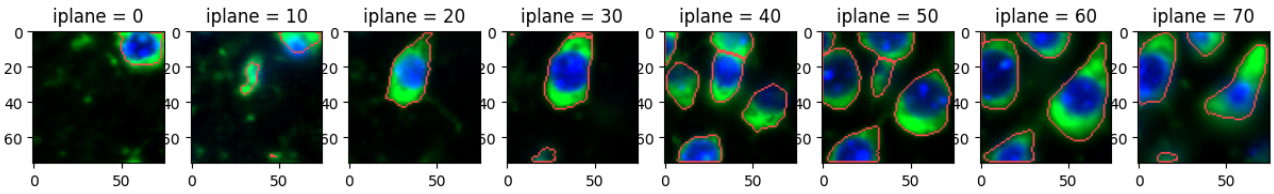


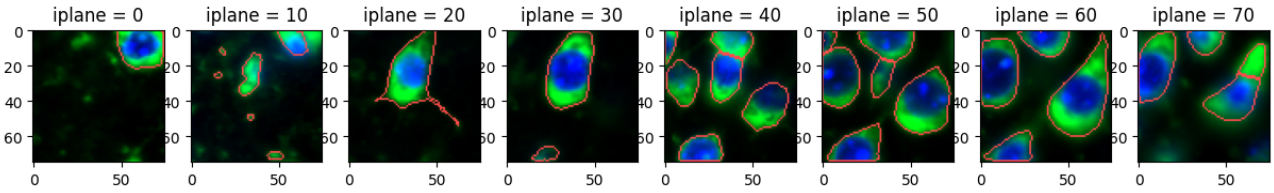
Figure 10: Sample Images



(a) 2D Cell Segmentation



(b) 3D Segmentation (Method 1)



(c) 3D Segmentation (Method 2)

Figure 11: Qualitative results for 2D and 3D cell segmentation.

5 Discussion

5.1 Summary of Experimental Findings

The experiments presented in Section 4 provide insights into both the algorithmic foundations of Cellpose and the practical effectiveness of integrating Cellpose with SAM through supervised fine-tuning.

The from-scratch reimplementation of the flow-field segmentation algorithm successfully reproduced the core behavior described in the Cellpose framework. Specifically, gradient-based pixel dynamics guided pixels toward instance centers, enabling effective separation of touching and overlapping cells on synthetic data. These results confirm that the flow-field representation and the associated gradient tracking algorithm constitute the essential mechanism underlying Cellpose's instance segmentation capability.

In addition, the Cellpose-SAM experiments demonstrate that strong segmentation performance can be achieved when pretrained models are properly fine-tuned. Using only five human-corrected training images, the fine-tuned Cellpose-SAM model achieved an Average Precision of 76.1% on unseen test data, highlighting the effectiveness of transfer learning and human-in-the-loop optimization in low-data regimes.

5.2 Synergistic Integration of SAM and Cellpose

Cellpose-SAM combines the complementary strengths of SAM and Cellpose to overcome their respective limitations in cellular instance segmentation. SAM provides rich and transferable visual representations through large-scale pretraining but lacks automatic instance separation and topological guarantees.
In contrast, Cellpose offers a geometry-driven flow-field formulation that enables prompt-free, topologically consistent separation of touching cells but suffers from limited generalization due to small-scale training. By replacing Cellpose's U-Net with SAM's pretrained Vision Transformer encoder and introducing targeted architectural adaptations, Cellpose-SAM achieves robust generalization, accurate instance separation, and high sample efficiency. This synergy enables fully automatic, high-quality segmentation across diverse microscopy modalities while maintaining practical computational efficiency.

5.3 Role of SAM in the Cellpose-SAM Framework

The experimental results clarify the functional role of SAM within the Cellpose-SAM architecture. SAM does not directly generate instance-level segmentations suitable for dense cellular images, nor does it inherently produce flow-field representations. Instead, SAM serves as a powerful pretrained image encoder that provides rich semantic and contextual features across diverse visual domains.

In the Cellpose-SAM framework, these features are mapped to Cellpose-style flow fields through supervised learning. The successful fine-tuning experiment demonstrates that when this mapping is learned, SAM significantly improves generalization and robustness, particularly when adapting to new datasets with limited annotations. This explains why direct, untrained integration of SAM embeddings into the flow-field pipeline produces degenerate outputs, while supervised fine-tuning yields meaningful segmentation results.

5.4 Importance of Flow-Field Representation

Across all experiments, the flow-field representation emerges as the central component of successful cell segmentation. Unlike pixel-wise classification approaches, flow fields encode global geometric structure by guiding pixels toward instance centers, naturally resolving ambiguities caused by touching or overlapping cells.

The effectiveness of the fine-tuned Cellpose-SAM model further reinforces this observation: although the backbone architecture is enhanced with SAM, the final segmentation quality still depends on accurate flow prediction and subsequent gradient-based clustering. Thus, the flow-field formulation remains indispensable regardless of the choice of feature extractor.

- **Limitation of pixel-wise segmentation**: Binary or semantic pixel classification cannot reliably separate touching or overlapping cells, as adjacent instances share the same foreground label and lack explicit boundary signals.
- **Heuristic post-processing is fragile**: Methods such as watershed rely on assumptions that frequently fail in dense or highly crowded cellular environments.

- **Geometry-driven instance encoding**: Flow-field representation assigns each pixel a vector pointing toward its instance center, encoding instance identity through convergence behavior rather than boundary prediction.
- **Robustness to invisible boundaries**: Even when cell boundaries are ambiguous or absent due to extreme overlap, directional flow toward distinct centers enables reliable instance separation.
- **Reduced sensitivity to annotation noise**: Flow fields depend on centroids rather than precise boundaries, making them more robust to inter-annotator variability.
- **Topological guarantees**: Gradient tracking ensures that each foreground pixel converges to exactly one sink, producing connected, non-overlapping instance masks by construction.

5.5 Effectiveness of Human-in-the-Loop Fine-Tuning

The human-in-the-loop experiment demonstrates a practical and efficient strategy for adapting complex segmentation models to new domains. By iteratively correcting model predictions rather than annotating images from scratch, the annotation burden is substantially reduced while maintaining competitive performance.

Achieving an Average Precision of 76.1% using only five training images illustrates the strength of combining pretrained models with targeted human supervision. This approach is particularly well suited for biomedical imaging applications, where expert annotations are costly and datasets are often limited in size.

5.6 Limitations

Despite the promising results, several limitations should be acknowledged:

1. **Synthetic Data Evaluation**: The from-scratch implementation was evaluated exclusively on synthetic data, which may not fully capture the variability present in real microscopy images, including noise, blur, contrast variation, and imaging artifacts.
2. **Limited Test Set**: The fine-tuning experiments relied on a small test set consisting of only three images, limiting the statistical significance of the reported quantitative metrics. Evaluation on larger and more diverse benchmarks, such as TissueNet or LIVECell, would strengthen the conclusions.
3. **Computational Requirements**: Cellpose-SAM incurs substantial computational and memory overhead, with a model size of approximately 1.23 GB and a requirement of at least 8 GB of GPU VRAM. These requirements exceed those of classical segmentation models and may restrict accessibility in resource-constrained environments.
4. **Overlapping Cells**: The flow-field representation assumes that each pixel belongs to a single cell instance. As a result, truly overlapping or occluded cells, particularly in 3D projections, cannot be accurately segmented without additional modeling mechanisms.
5. **Cell Size Dependency**: Segmentation performance remains dependent on reasonable cell size estimation for image resizing. Although the pretrained model is more robust to estimation errors than the original Cellpose, inaccuracies in size estimation can still affect performance.

5.7 Implications and Future Work

The integration strategy demonstrated in this work—combining foundation model representations with domain-specific geometric mechanisms—suggests a general paradigm that extends beyond cellular segmentation. Foundation models contribute rich, transferable representations learned from massive and diverse datasets, while domain-specific algorithms encode task-relevant inductive biases grounded in expert knowledge. Individually, neither approach fully addresses the challenges of specialized biomedical imaging tasks; together, they enable capabilities that exceed what either component can achieve in isolation.

This complementary integration has the potential to advance a wide range of biomedical imaging applications, including vessel segmentation, organelle detection, and tissue classification, where effective domain-specific methods exist but are constrained by limited generalization. By leveraging pretrained representations while preserving domain structure, such hybrid frameworks can improve robustness across heterogeneous imaging conditions.

Beyond technical performance, the proposed framework democratizes access to advanced image analysis. By substantially reducing annotation requirements and enabling adaptation through intuitive correction-based workflows, Cellpose-SAM lowers the barrier to entry for laboratories lacking extensive computational resources

or large annotated datasets. This increased accessibility broadens the community of researchers who can apply state-of-the-art segmentation methods to biological discovery.

5.8 Future Research Directions

Several promising avenues for future research emerge from this work:

- **Expanded training data**: Incorporating larger and more diverse cellular datasets—spanning a wider range of morphologies, imaging modalities, and experimental conditions—could further improve generalization to the heterogeneous images encountered in biological research.
- **Efficient model design**: Developing lightweight architectures through techniques such as knowledge distillation or parameter-efficient design would reduce computational and memory requirements, enabling deployment on standard workstations without dedicated GPU resources.
- **Temporal extension**: Extending the framework to temporal sequences would support unified segmentation and tracking in time-lapse microscopy, using flow-field correspondence to maintain cell identities across frames and enabling analysis of dynamic cellular processes.
- **Multi-instance flow modeling**: Designing flow representations capable of handling genuinely overlapping cells in thick tissue projections or confluent cultures would address a core assumption limiting the current method's applicability.
- **Uncertainty quantification**: Incorporating uncertainty estimation would help identify ambiguous regions requiring human review, improving the efficiency of human-in-the-loop workflows and supporting automated quality control in high-throughput settings.
- **Hybrid interaction paradigms**: Developing interfaces that combine fully automatic segmentation with optional prompted refinement would enable flexible workflows, matching the level of user interaction to image difficulty—minimal intervention for routine cases and targeted guidance for challenging scenarios.

6 Conclusion

This work investigated the Cellpose-SAM framework and demonstrated that integrating foundation model representations with a geometry-driven flow-field formulation achieves instance segmentation performance approaching human-consensus accuracy while substantially reducing annotation requirements.

Our experiments show that flow-field representation is the critical mechanism enabling robust separation of touching and overlapping cells. By encoding instance identity through directional convergence rather than explicit boundary prediction, the method succeeds in scenarios where traditional boundary-based approaches fail. Integrating this formulation with SAM's pretrained encoder addresses the generalization limitations of prior methods, enabling effective domain adaptation with minimal training data through a human-in-the-loop paradigm.

The proposed framework resolves fundamental limitations of both parent models. SAM acquires the ability to perform automatic instance separation, which it inherently lacks, while Cellpose gains strong generalization capabilities that were previously constrained by limited training data. This synergy validates the broader principle that foundation models and domain-specific geometric priors play complementary roles, and that neither alone achieves the performance enabled by their careful integration.

These findings have practical implications for biological research. By lowering the need for large annotated datasets and specialized machine learning expertise, the human-in-the-loop workflow makes high-quality cell segmentation accessible to a wider range of laboratories. More broadly, as foundation models continue to evolve, their integration with task-specific mechanisms represents a promising and scalable direction for advancing biomedical image analysis.

Table 8: Contribution of Team Members

Name	**Student ID**	**Main Contributions**
Le Thi Phuong Thao	2252757	• Problem statement and cell segmentation background • Survey of existing techniques • Proposed approach • Cellpose-SAM integration
Nguyen Thuy Tien	2252806	• Survey of existing techniques • Flow-field representation and algorithm design • Cellpose-SAM integration • Discussion and interpretation of results

References

[1] M. Pachitariu, M. Rariden, and C. Stringer, "Cellpose-sam: Superhuman generalization for cellular segmentation," *bioRxiv*, pp. 2025–04, 2025.

[2] O. Ronneberger, P. Fischer, and T. Brox, *U-net: Convolutional networks for biomedical image segmentation*, 2015. arXiv: 1505.04597 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1505.04597.

[3] C. Stringer, T. Wang, M. Michaelos, and M. Pachitariu, "Cellpose: A generalist algorithm for cellular segmentation," *Nature methods*, vol. 18, no. 1, pp. 100–106, 2021.

[4] M. Pachitariu and C. Stringer, "Cellpose 2.0: How to train your own model," *Nature methods*, vol. 19, no. 12, pp. 1634–1641, 2022.

[5] Y. Wang et al., "A systematic evaluation of computational methods for cell segmentation," *Briefings in Bioinformatics*, vol. 25, no. 5, 2024.

[6] A. Kirillov et al., "Segment anything," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 4015–4026.