

Cellpose SAM

Installation Settings Overview

Workflow

- **Input Data**
- **Training a Model***
- **Data Analysis**

Installation:

Create & Activate Environment:

1. conda create -n cellpose-sam python=3.10 -y
2. conda activate cellpose-sam

Install Cellpose:

1. python -m pip install 'cellpose[gui]'

Install Cellpose SAM pre-trained model:

1. conda activate cellpose-sam
2. python

At the >>> prompt type:

2. from cellpose import models
- model = models.CellposeModel(pretrained_model='cpsam')
- print("Loaded model:", model.pretrained_model)

Once downloading finishes type:

3. exit()

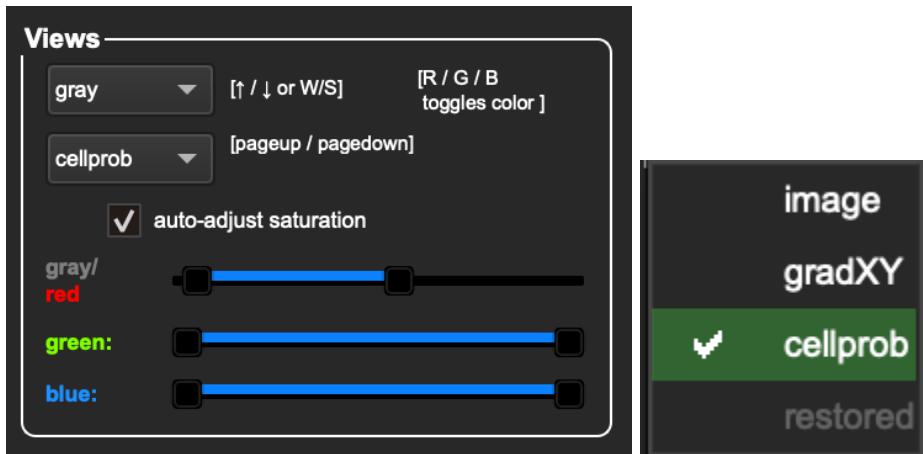
Launch GUI to begin:

conda activate cellpose-sam

python -m cellpose

Settings Overview:

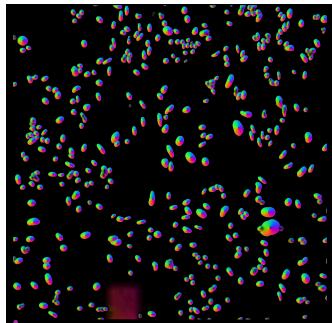
Views:



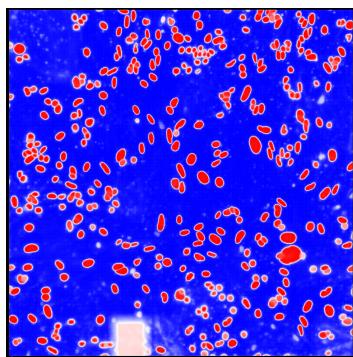
The setting I like to use is gray and then adjusting the saturation bar of gray/red to what it looks like in the screenshot above. This made the brightfield image way more visible.

image: Just shows the normal original image.

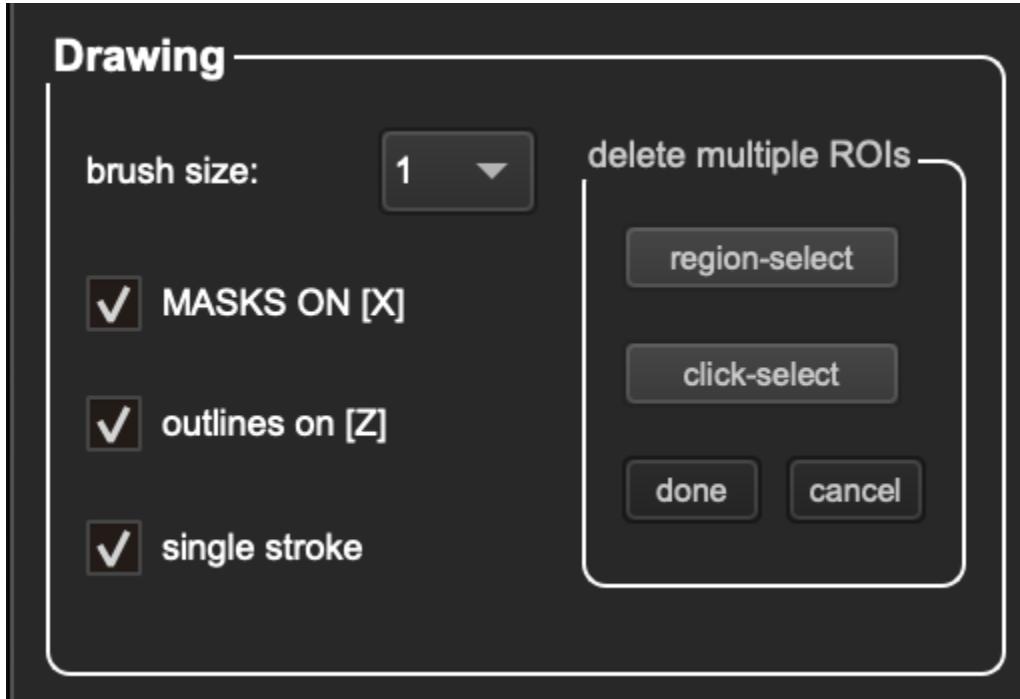
gradXY: This shows the vector field of the cells (flow field). Don't worry about this too much, just know that if the flow field is clean then there should a nice rainbow gradient that converges into a black or dark center (where the flow magnitude is 0) in the middle of each cell.



cellprob: Probability that an individual pixel is inside a cell. The more bright and hot red it is; the higher the probability. Lower intensity means not likely to be inside the cell. I would double check any medium/dim areas to see if they are actually cells/pixels within cells or not.



Drawing:



Brush size: Adjusts the size of the brush for drawing ROIs.

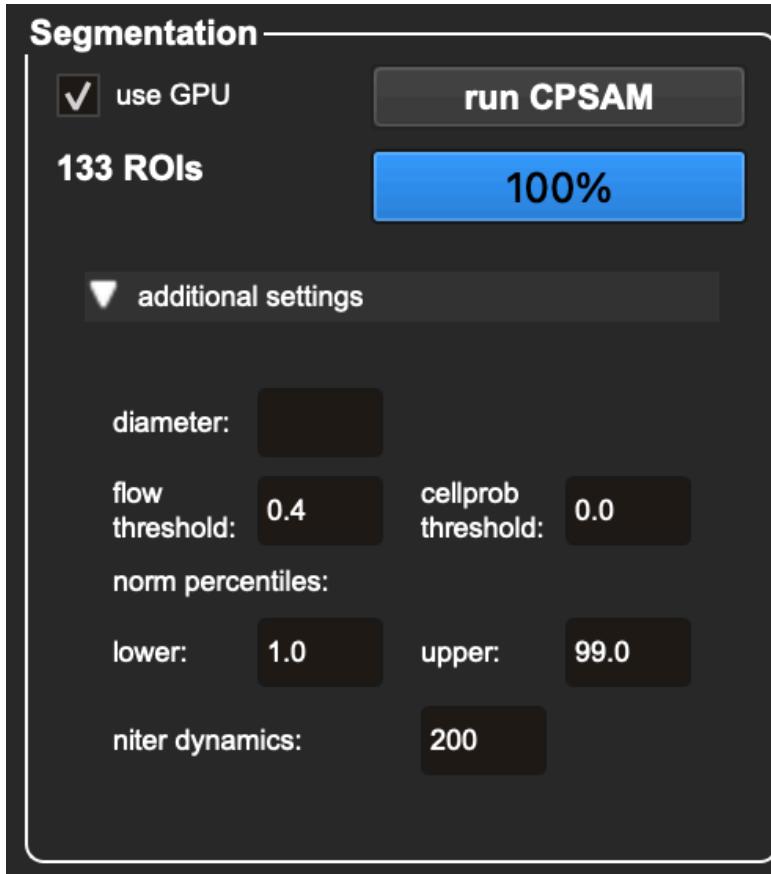
Masks: The colored mask over an ROI. I like click X a lot to take away and bring back the mask to check if the ROIs it circled are correct

Outlines: outlines of roi

Single stroke: keep this check marked in order to draw ROIs efficiently

Deleting ROIs: You can delete ROIs one at a time with click-select or a lot at the same time with region-select.

Segmentation: (Depends on the dataset in question)



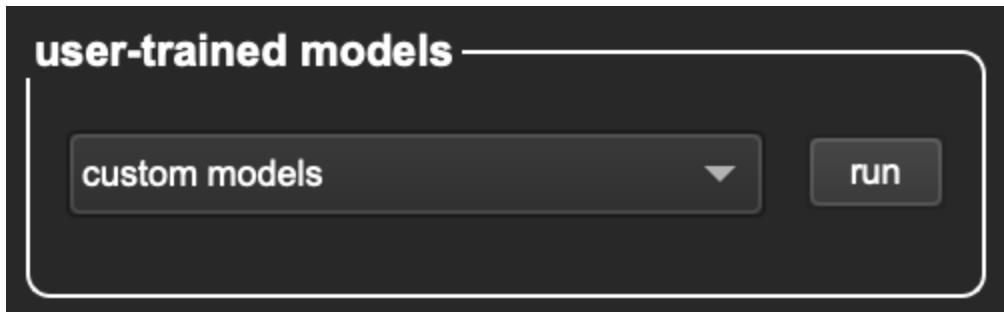
Diameter: you can set the diameter of each cell by counting the pixels. cellpose SAM is invariant to cell size so you don't have to do this in this case.

Flow threshold: Lower this if there are too many weird masks created. Raise this if a lot of real cells are discarded.

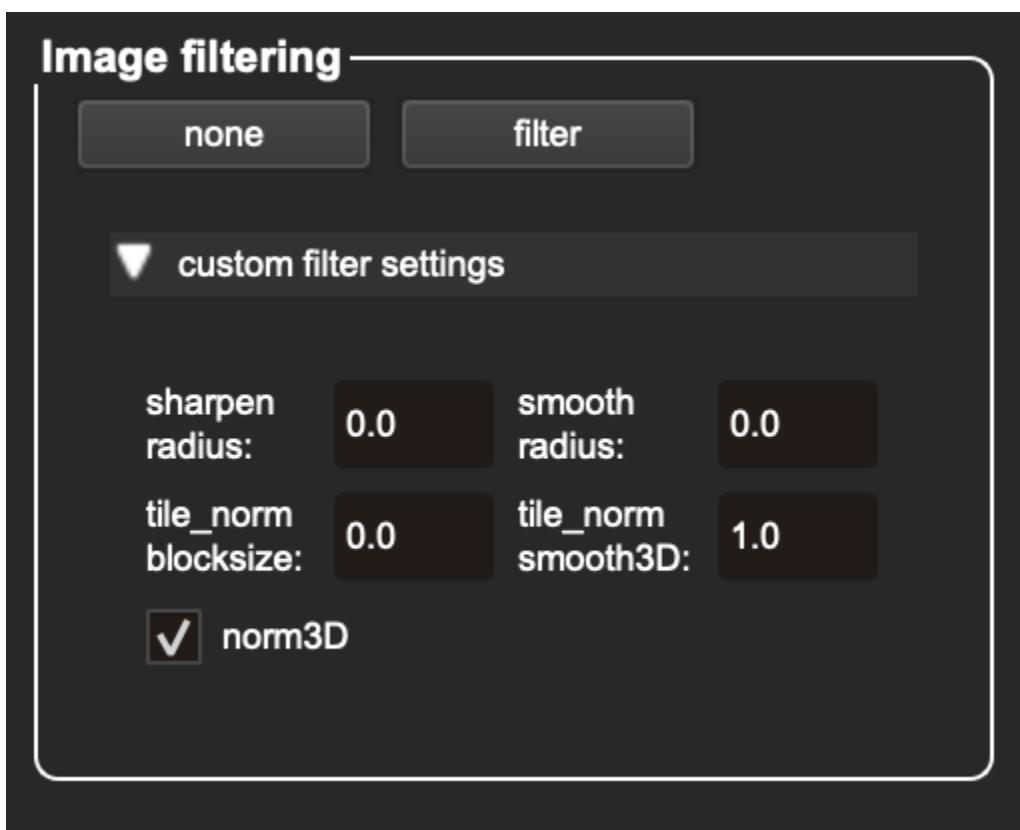
Cellprob threshold: If missing faint cells lower this value. If background blobs that aren't cells are kept then raise this value.

norm/percentile: don't change this

Niter dynamics: don't change this



This is where the custom model that you trained on your dataset will appear.

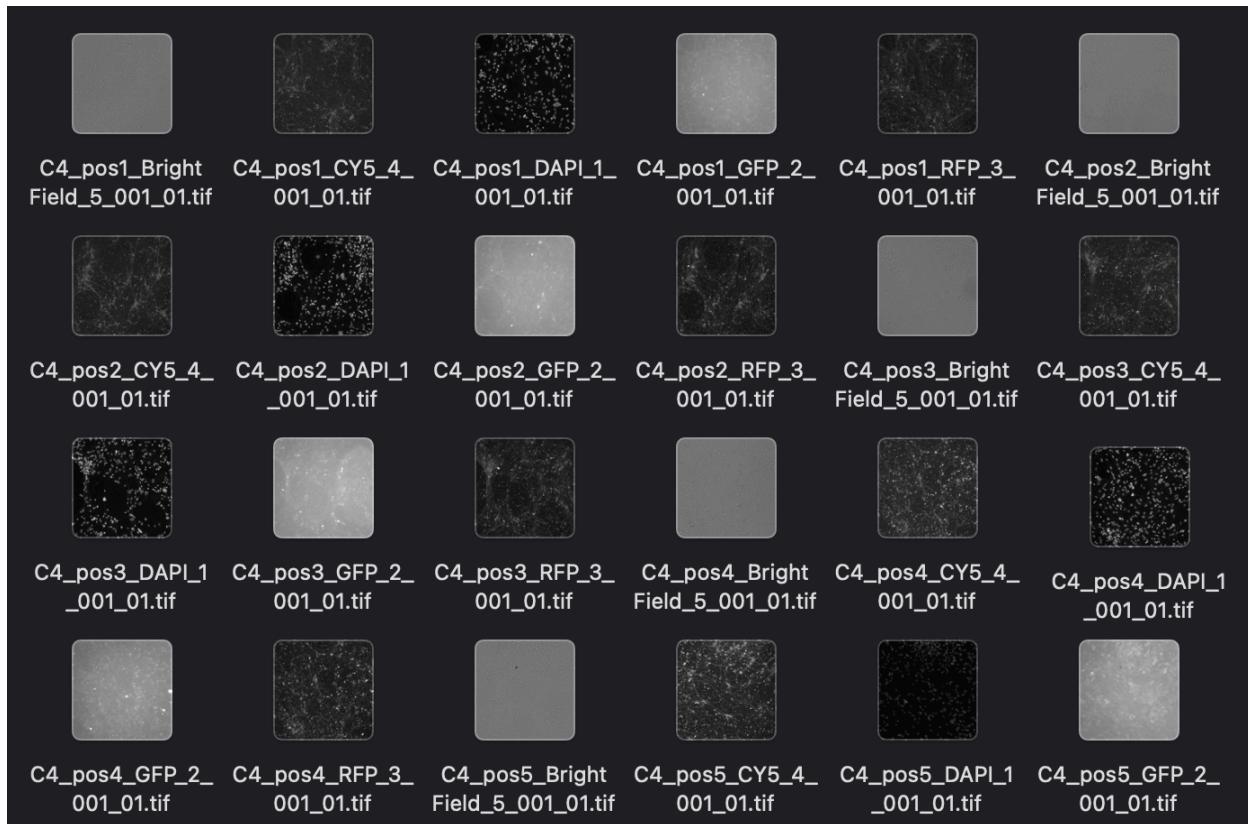


Only change these if your data is blurry, fuzzy, grainy, or has uneven illumination. This data seemed to run fine without it.

Workflow:

1) Input Data:

Your input data will look something like this:



C# is the specific well on the plate

pos# is the specific position of the field of view of the image on that well.

CY5/DAPI/GFP/RFP is the specific indicator captured in that position in that well on the plate. Each indicator is for a specific kind of cell or neuron. It can also be for all cells and neurons.

Dapi - all cells (soma and neurons)

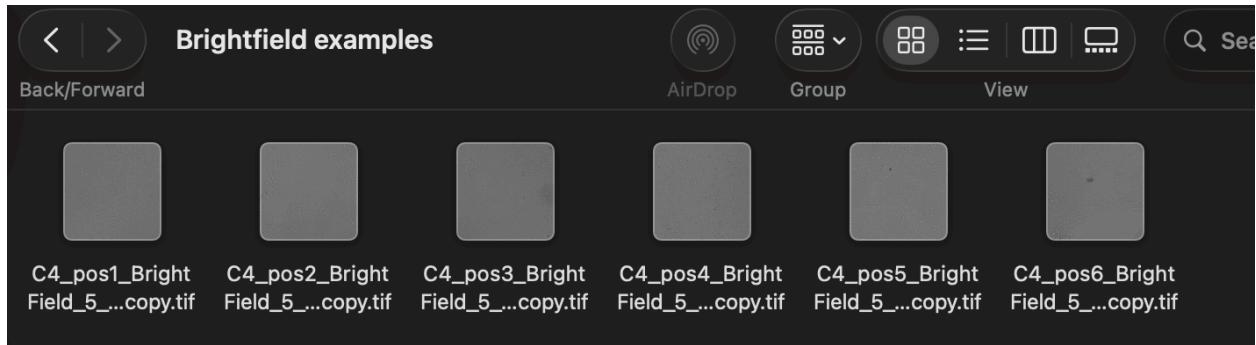
Cy5- all neurons MAP2 (marker for skeleton protein of neurons; recognizes all neurons)

Rfp- gaba neurons inhibitory

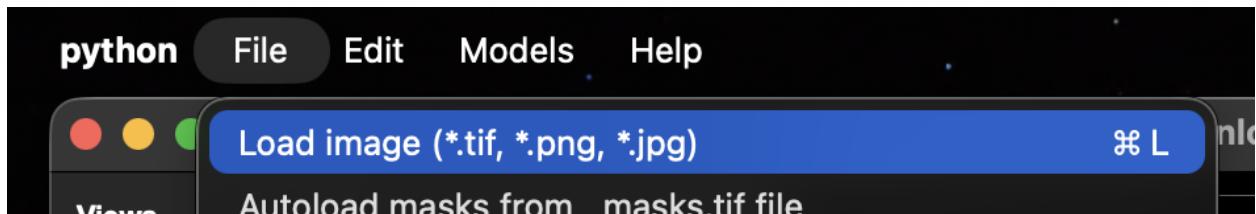
Your goal is to train a CPSAM model on each kind of indicator (aka cell type). So you will have a CPSAM model specifically tuned for detecting GABA neurons (GFP), a CPSAM model for detecting all cells (DAPI), etc.

2) Training a Model:

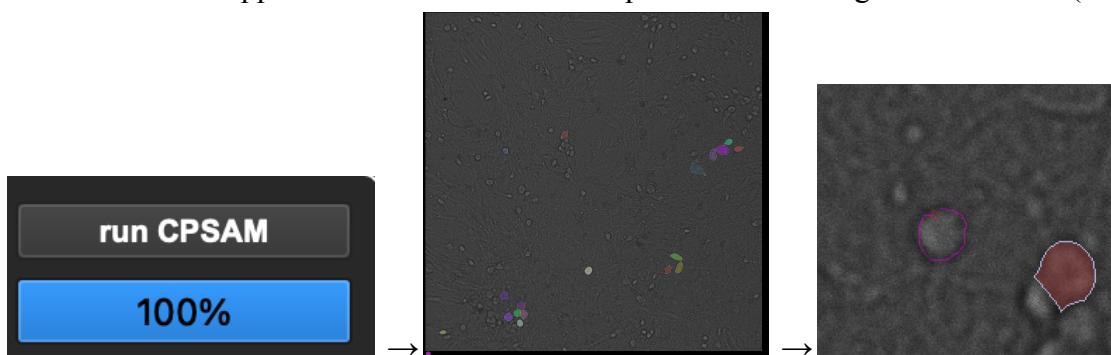
1. Put images you want it to train on in the folder. They should be of the same modality (all brightfield in this case). From my experience it gets pretty good at 4-6 images of training data. I would suggest you put **at least 10 images in the folder**. 4-6 to train on and then the rest to test the model to see if it is good or not.



2. Load one image from the folder. You can rotate through the images in the folder (on the GUI) by pressing left and right arrow keys ← and →. Make sure you click on the GUI image screen before hitting arrow keys else it won't rotate through the images.

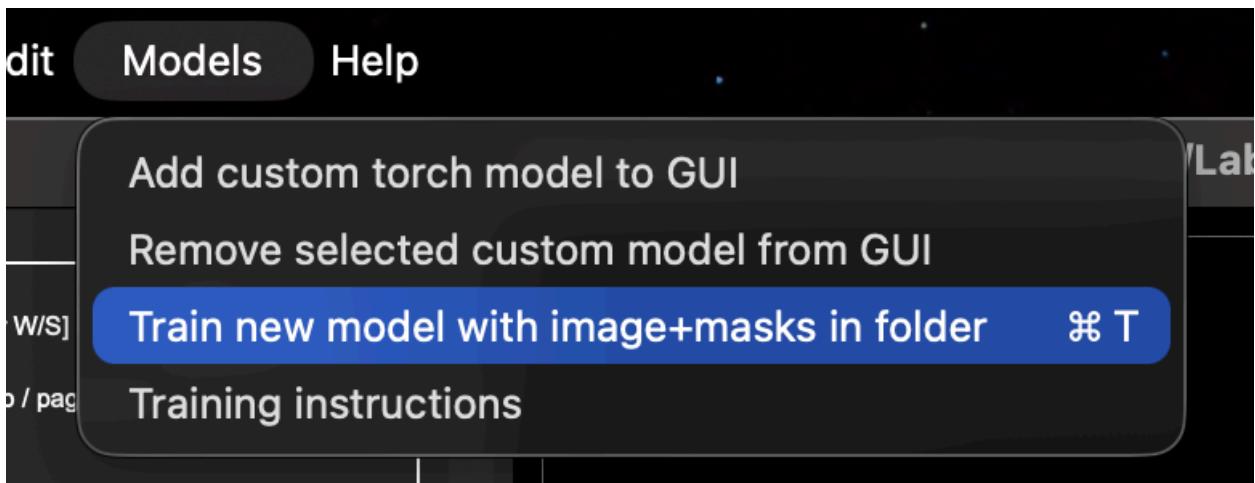


3. Run cellpose SAM. Then do command+click to delete wrong ROIs. To create new ROIs you right click on the edge of the ROI you want to create a mask for. You should see a red circle appear. Trace the ROI and complete the circle to get a ROI mask. (seen bellow)

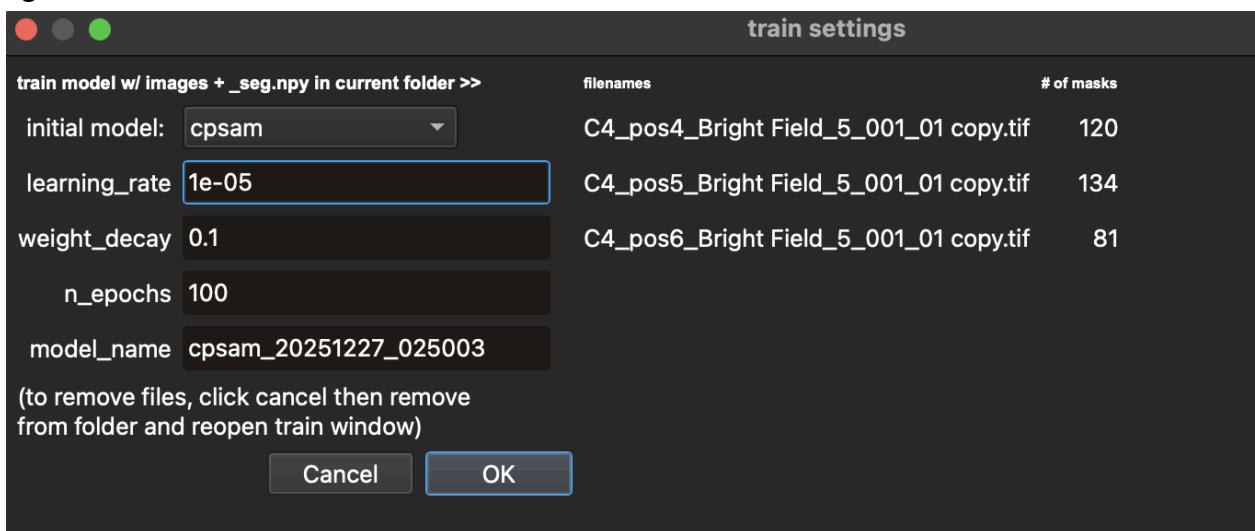


You want to trace all of the missed ROIs. Then hit the arrow key to go to the next image and do the same thing (run CPSAM → delete false ROIs & draw masks for missed ROIs). Do this for 4 images fully.

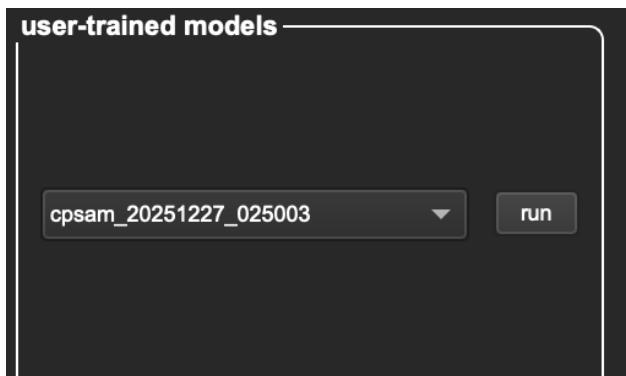
4. Train the model on these images that you created masks for.



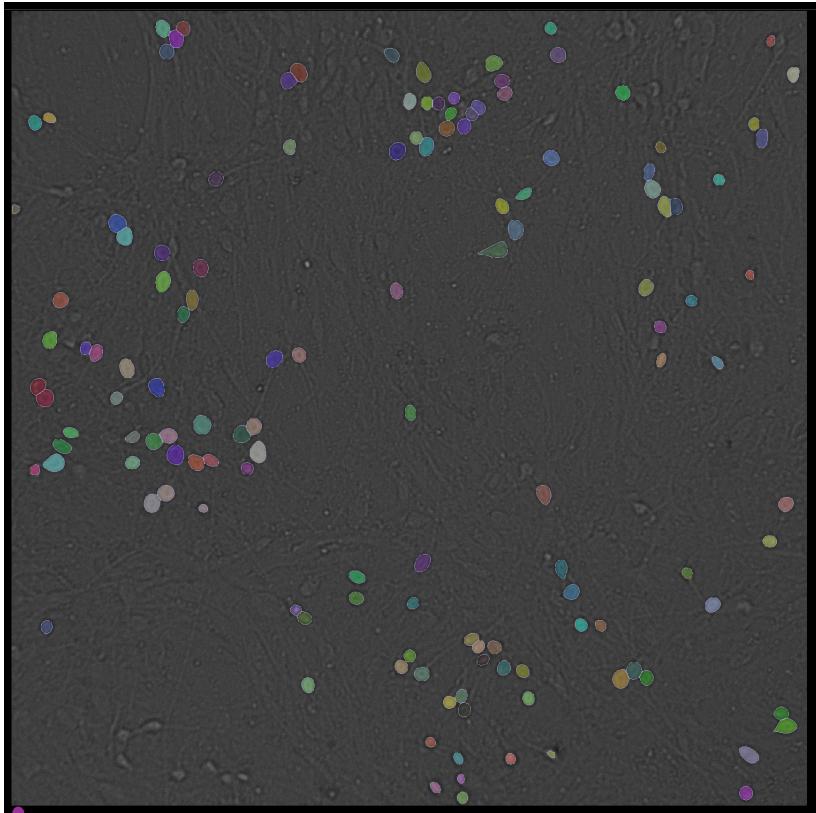
The default settings are perfectly fine. You will see the images that you created masks for on the right.



5. After training your model run it on the rest of the images in the folder (by cycling through the images with arrow keys ← & →)



6. Custom model result (using 3 initial annotated images):



This new model got 135 ROIs but missed around 15-20 from what I can tell. You just create new masks for those. Then move onto the next image and run the custom model and do the same thing by creating new masks for the missed ROIs.

7. After deleting false ROIs and creating new masks for the missed ROIs from your custom model, you can update the model by repeating step 4. (Essentially, you just update your model with new images until it doesn't miss any more masks)

train settings

filenames	# of masks
C4_pos1_Bright Field_5_001_01 copy.tif	153
C4_pos2_Bright Field_5_001_01 copy.tif	133
C4_pos4_Bright Field_5_001_01 copy.tif	120
C4_pos5_Bright Field_5_001_01 copy.tif	134
C4_pos6_Bright Field_5_001_01 copy.tif	81

train model w/ images + _seg.npy in current folder >>

initial model: cpsam

learning_rate: 1e-05

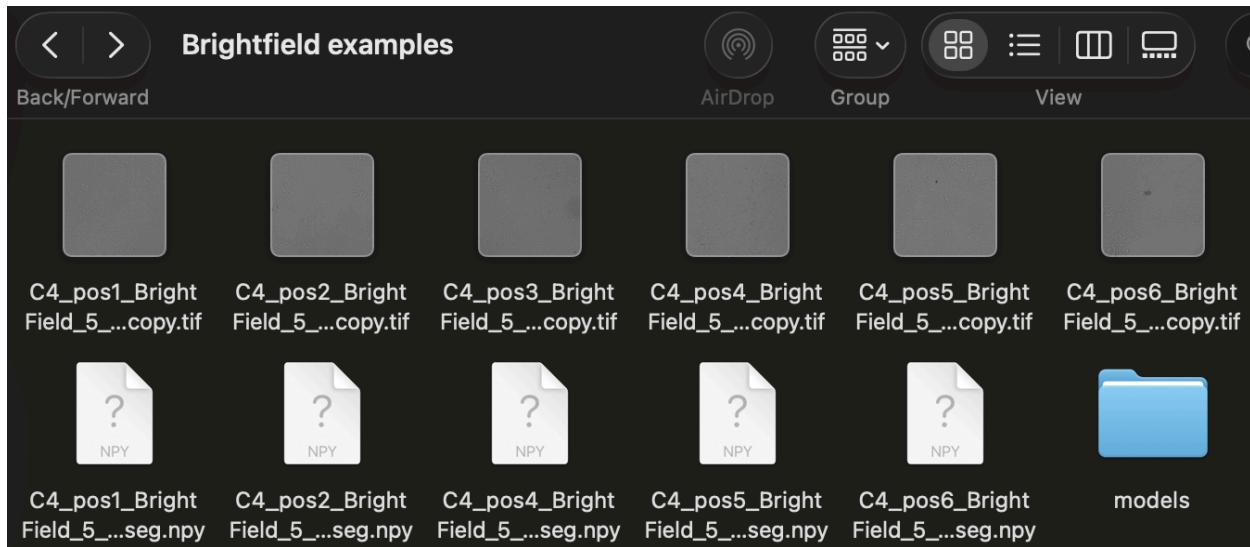
weight_decay: 0.1

n_epochs: 100

model_name: cpsam_20251227_025003

(to remove files, click cancel then remove from folder and reopen train window)

8. The model should be saved in the original folder that you had your data in. You can use this model anytime by selecting it in the GUI whenever you want to analyze data.



Repeat these steps to train a new CPSAM model for each indicator. You should have a specific CPSAM model for each indicator.

3) Data Analysis

1. Get data from a specific day
2. Run the respective model on the respective piece of data. (So GFP/GABA model on GFP/GABA indicator tiffs from the scope.)
3. Tally the total number of ROIs found by the model for that indicator on that specific day's data.
4. Look at trends over time. Did the number of ROIs increase or decrease for the ROIs of that specific indicator? EI ratio is one parameter you can get from all data.