Introduction to Nuclei Segmentation with StarDist

Martin Weigert, Olivier Burri, Siân Culley, Uwe Schmidt

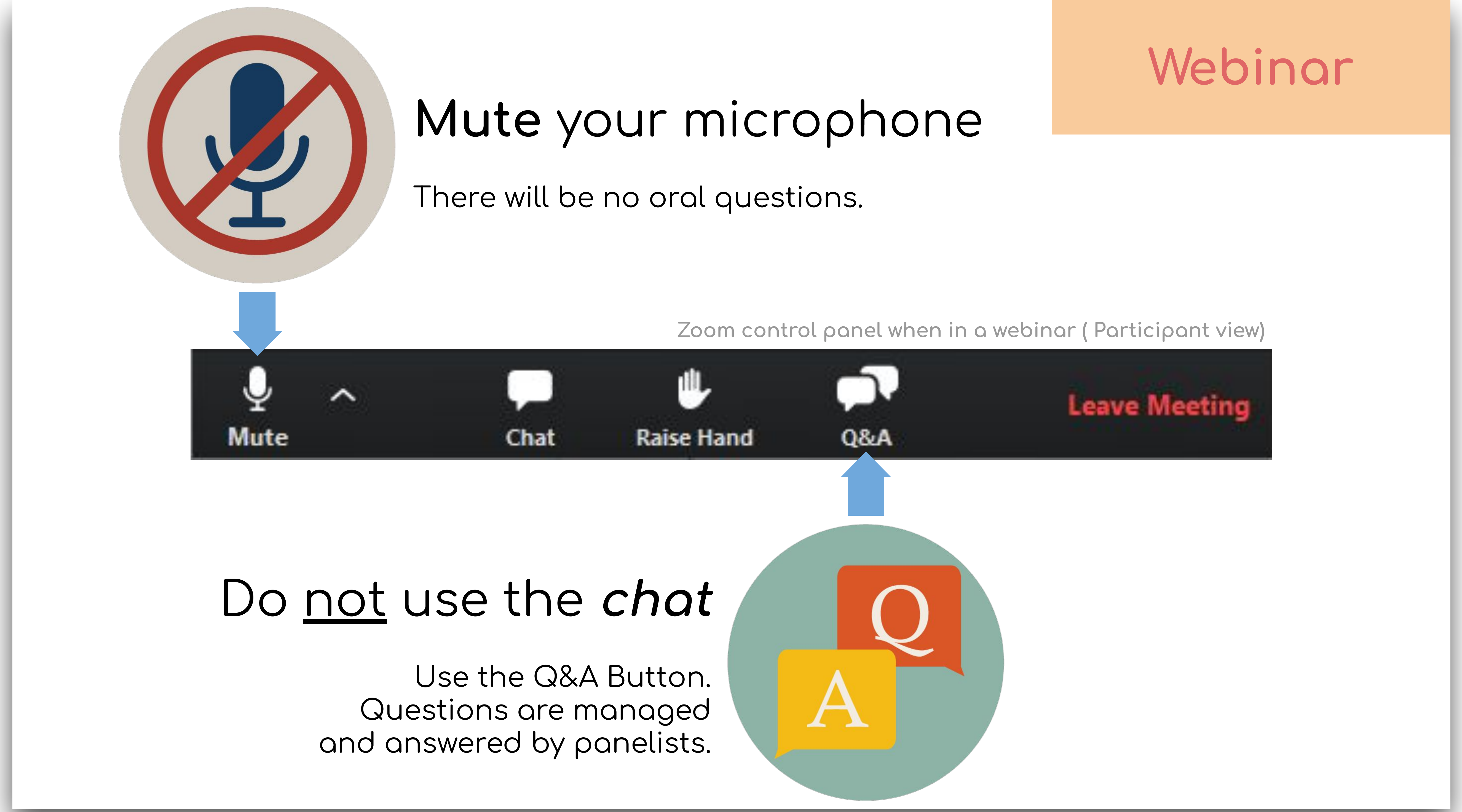
Neubias Academy @ Home, April 28th, 2020

The webinar will start 15:40 CEST

If you are already here:

1. Please follow the instructions to the right

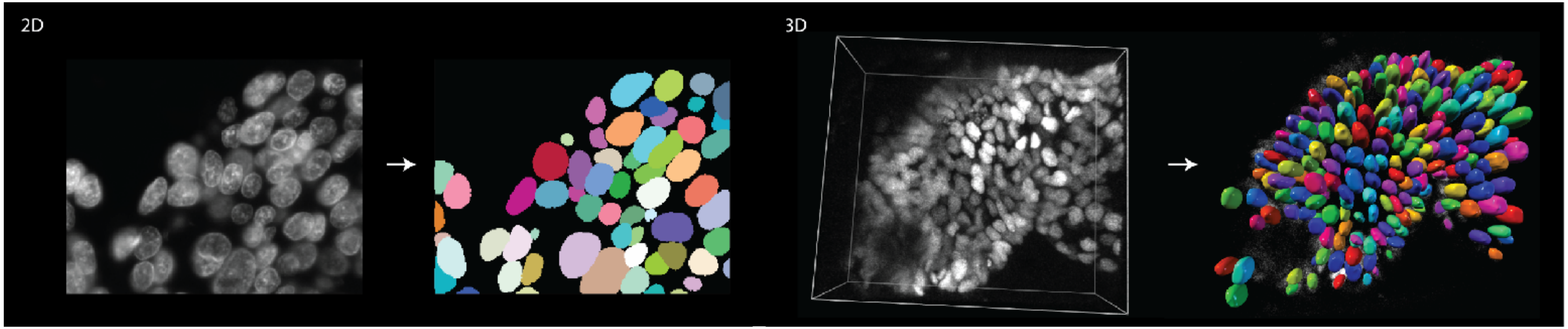
2. Please answer the short questions in the poll
(which should popup on your zoom screen)



Introduction to Nuclei Segmentation with StarDist

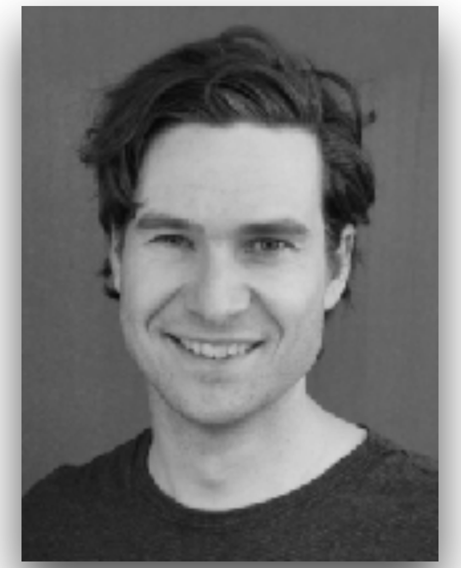
Martin Weigert, Olivier Burri, Siân Culley, Uwe Schmidt

Neubias Academy @ Home, April 28th, 2020



Webinar Material: https://github.com/maweigert/neubias_academy_stardist

Speakers/Moderators:



Martin Weigert
EPFL, Lausanne
@martweig

Uwe Schmidt
MPI-CBG, Dresden
@uschmidt83

Olivier Burri
EPFL, Lausanne
@ChiguireKun

Siân Culley
UCL, London
@SuperResoluSian

1. Introduction to nuclei segmentation and StarDist
2. Questions & Answers 1
3. How to use StarDist
4. StarDist in a core facility
5. Questions & Answers 2

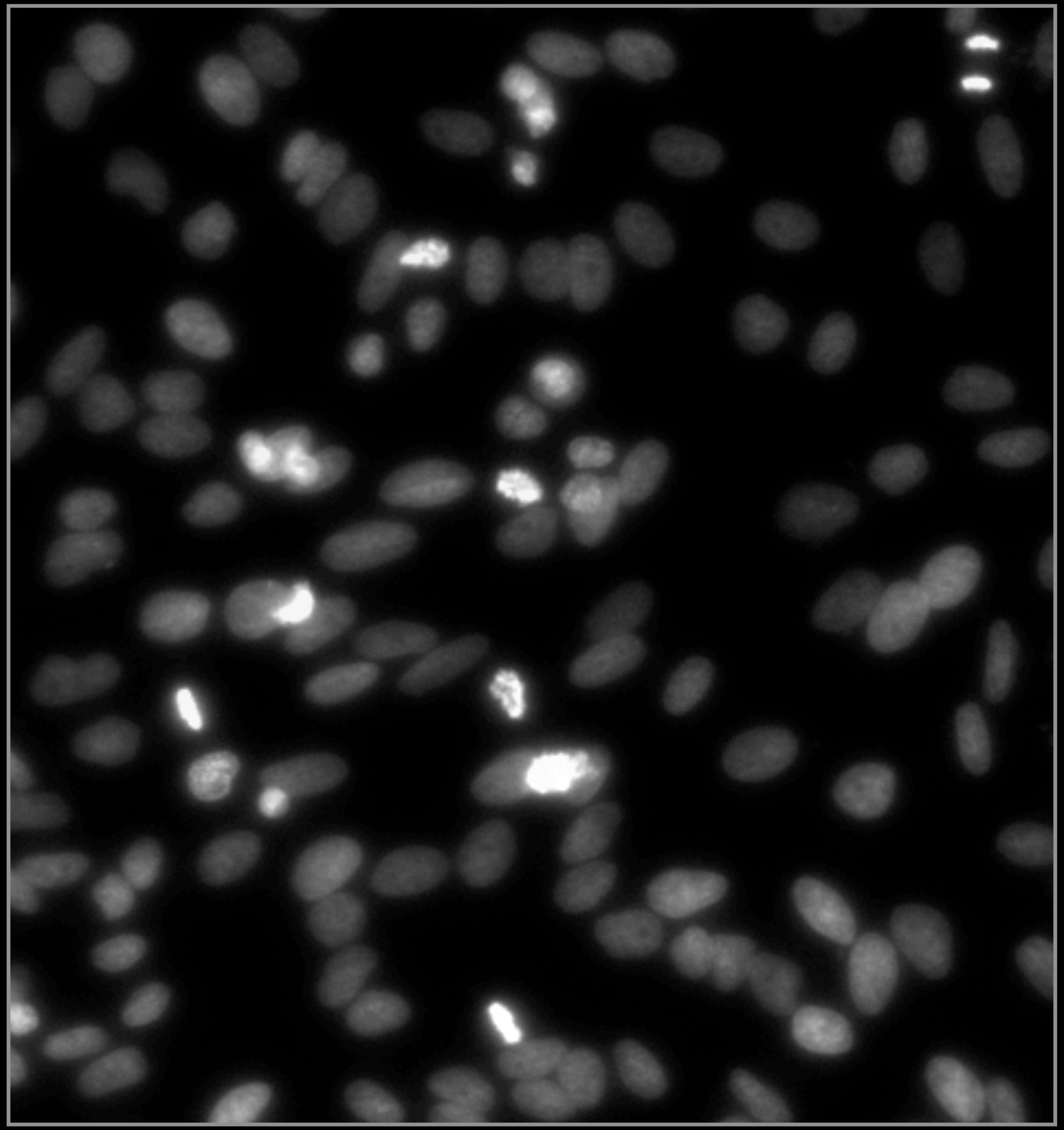
Outline

1. Introduction to nuclei segmentation and StarDist
2. Questions & Answers 1
3. How to use StarDist
4. StarDist in a core facility
5. Questions & Answers 2

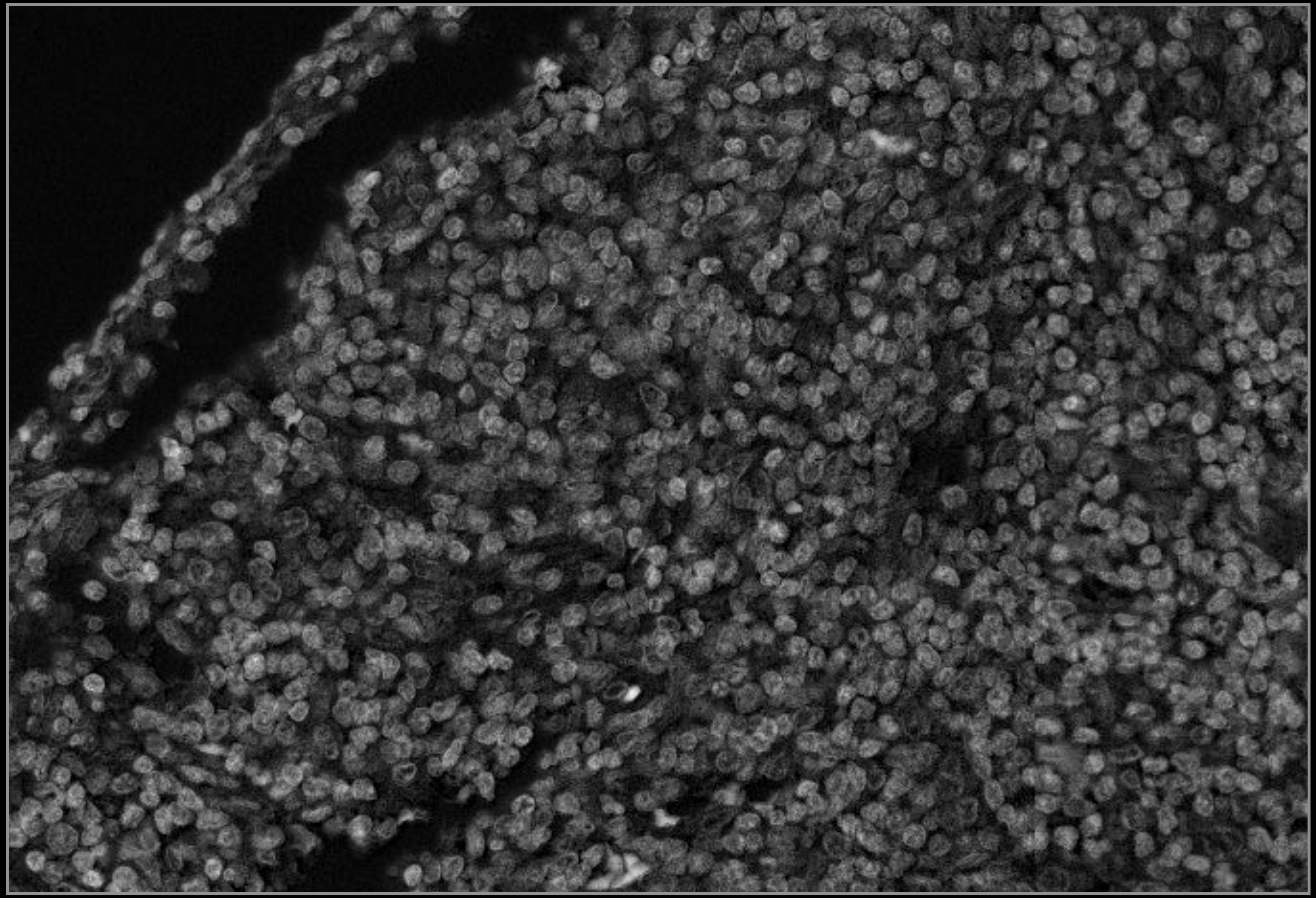
Introduction to nuclei segmentation and StarDist

Cell/Nuclei Segmentation in Microscopy

2D - Fluorescence



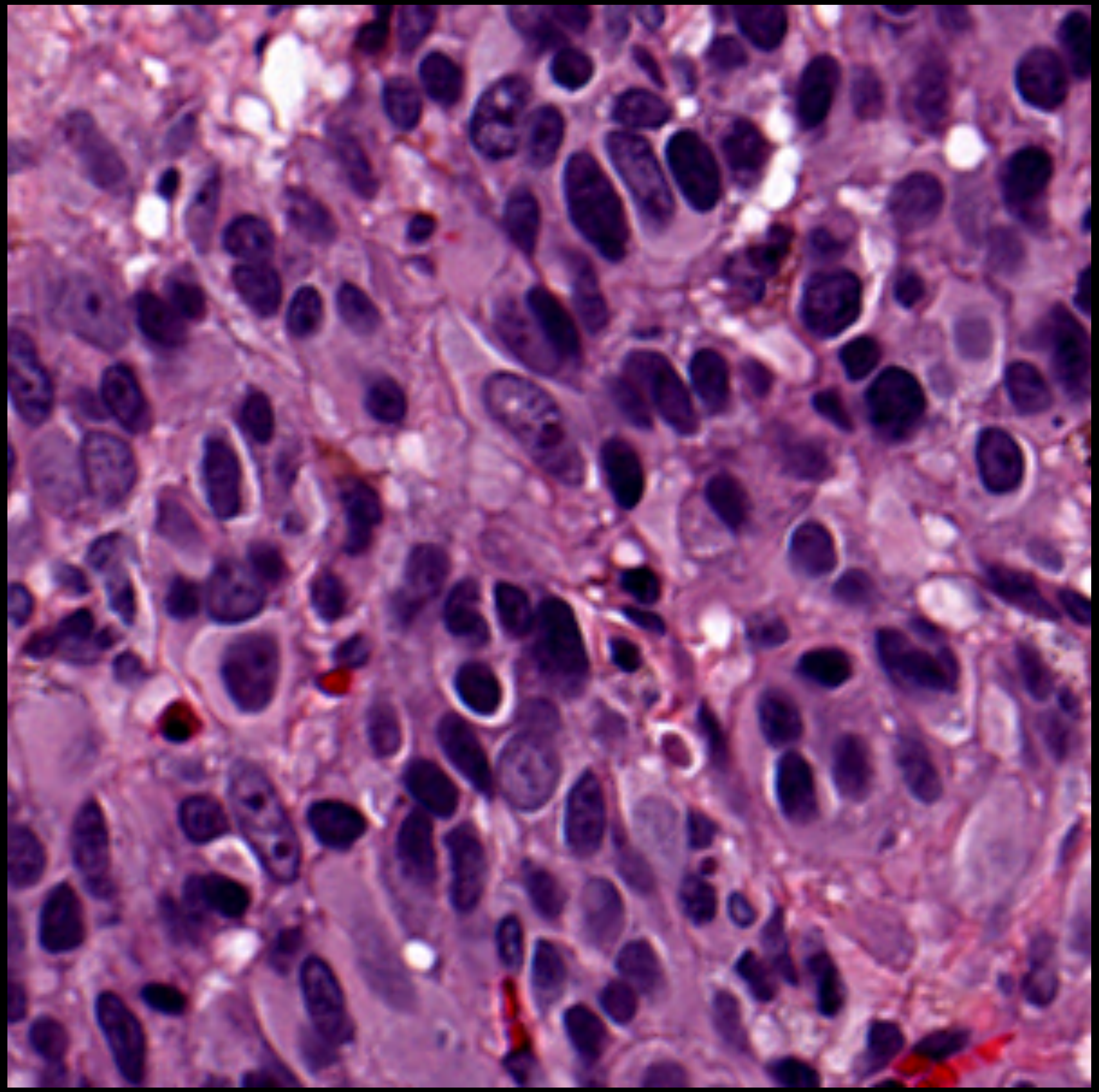
Data from Caicedo et al, 2019



Data from Anna Maria Tsakiroglou (Manchester)

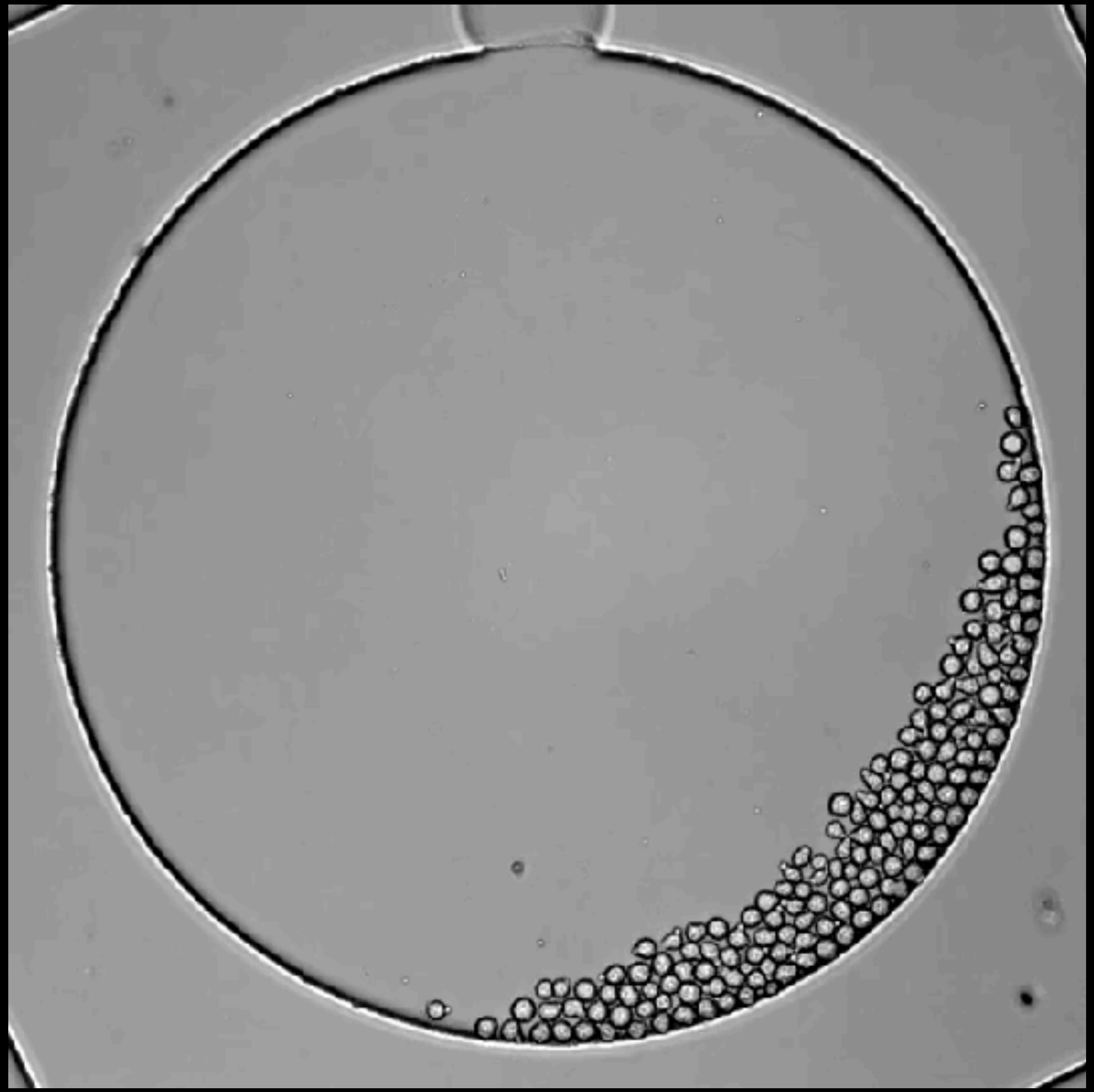
Cell/Nuclei Segmentation in Microscopy

2D - RGB/Histopathology



H&E stain, data from CancerImagingArchive

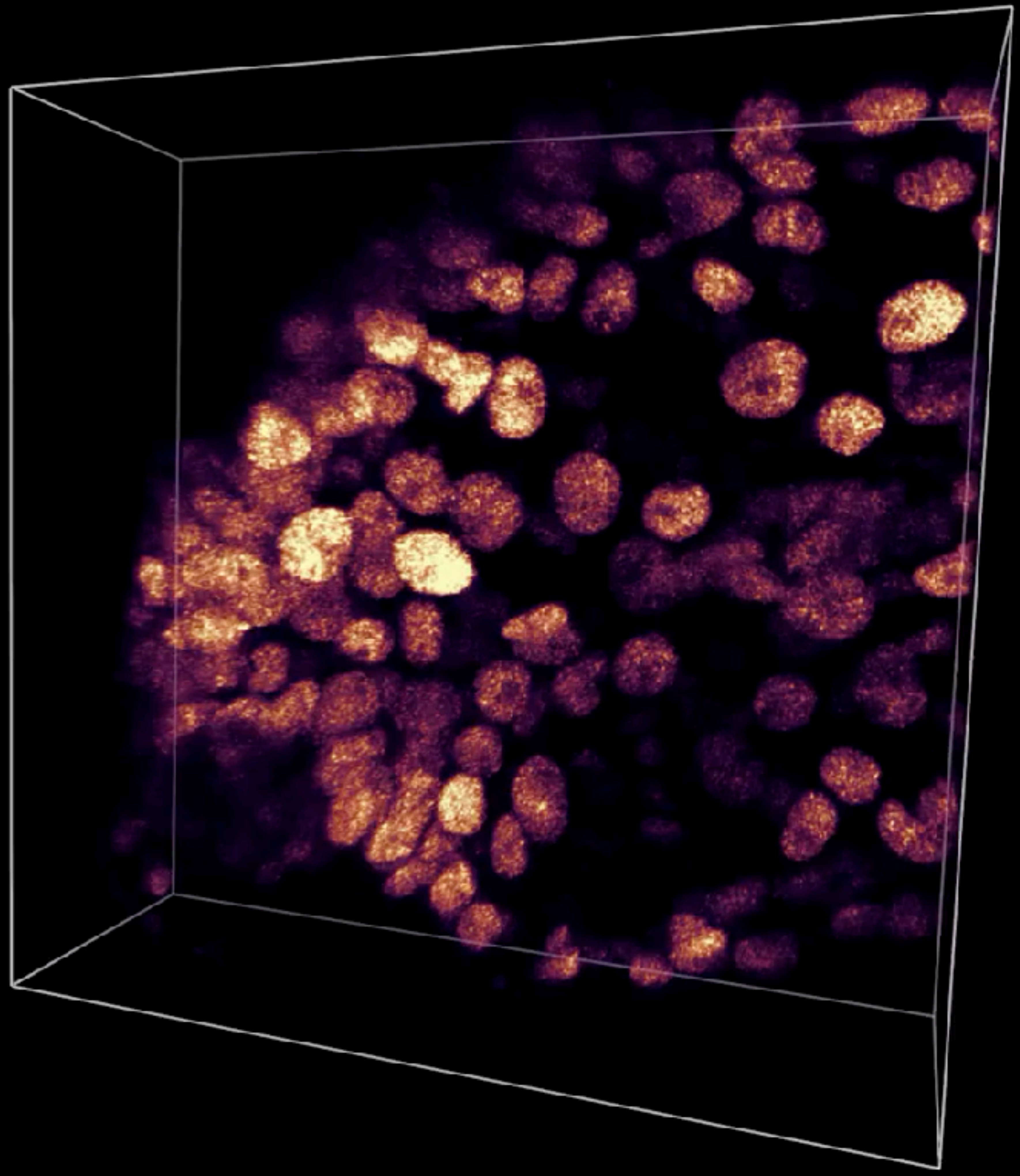
2D + time (Brightfield)



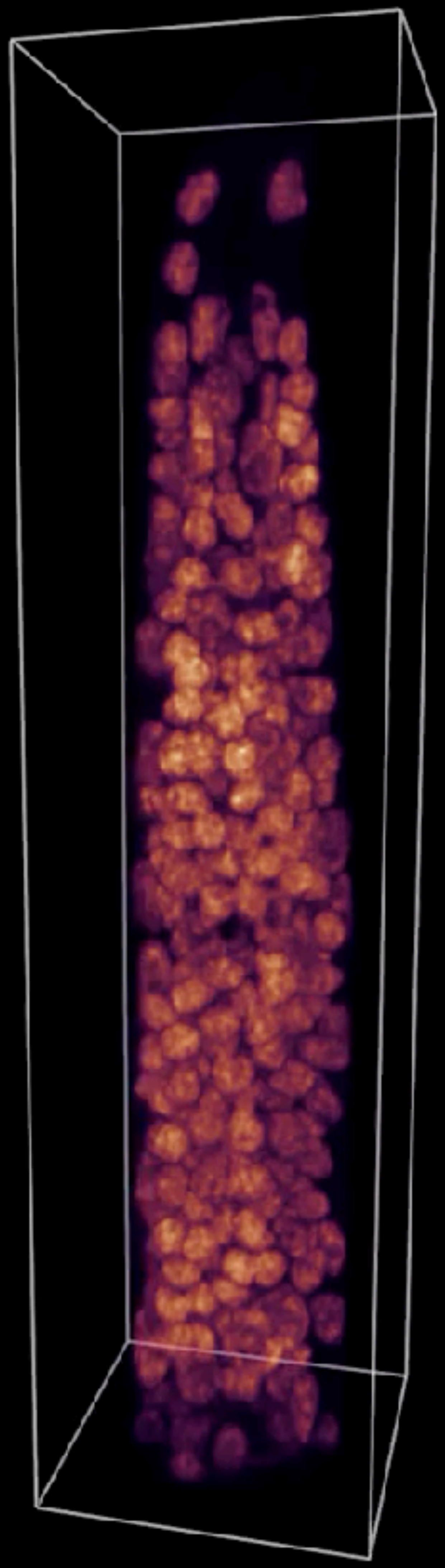
Mouse stem cells, data from cell tracking challenge

Cell/Nuclei Segmentation in Microscopy

3D (+time)



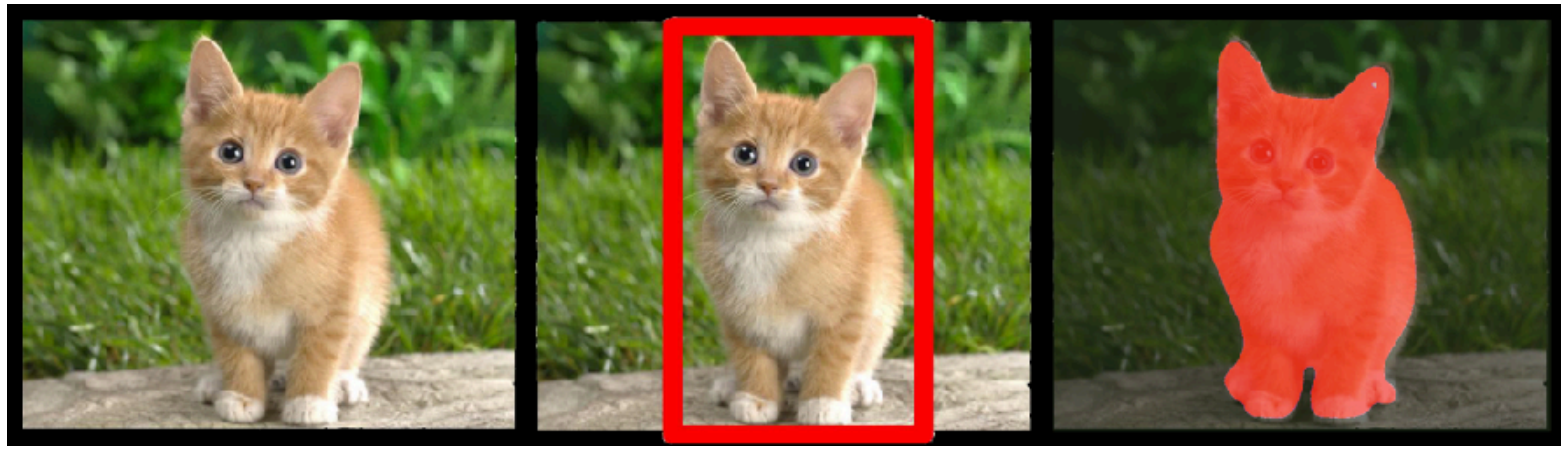
Paryhale Data from Ko Sugarawa

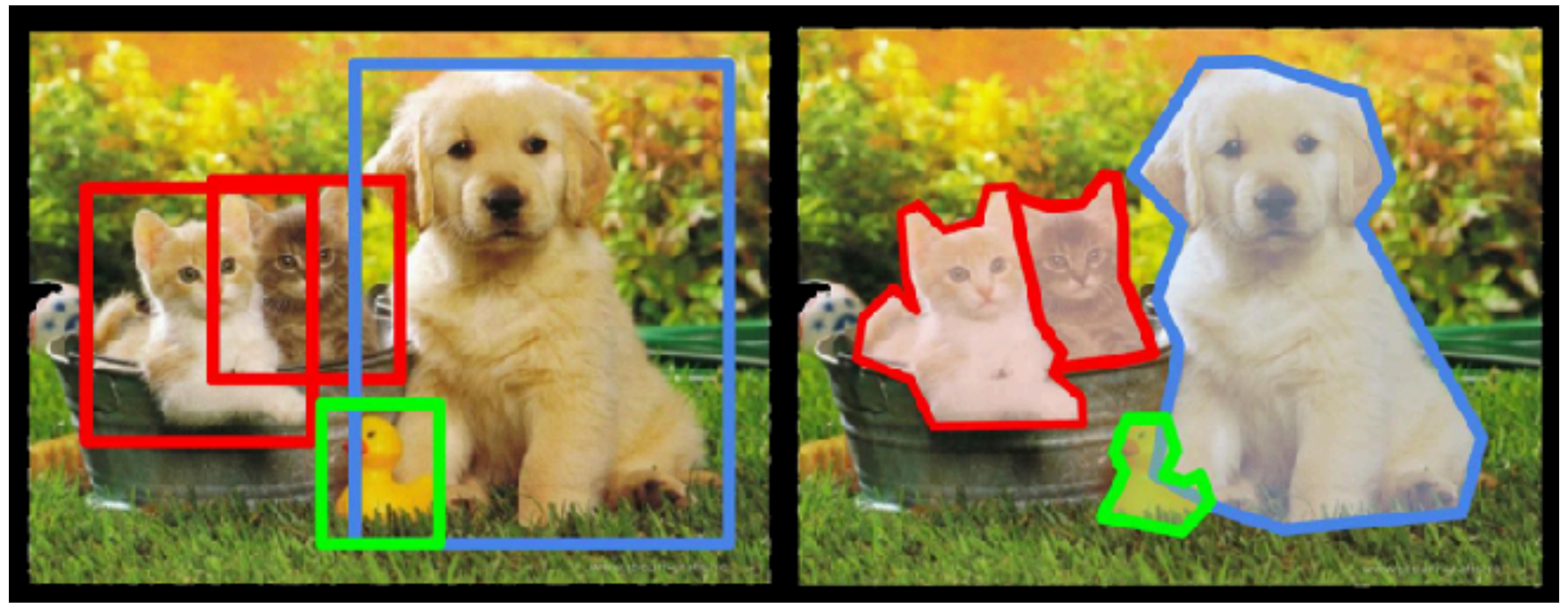


C elegans Data from Dagrmar Kainmüller

Computer Vision - Common Problems

Single Object			Multiple Objects	
Image Classification	Object Localization	Dense/Semantic Segmentation	Object Detection	Instance Segmentation

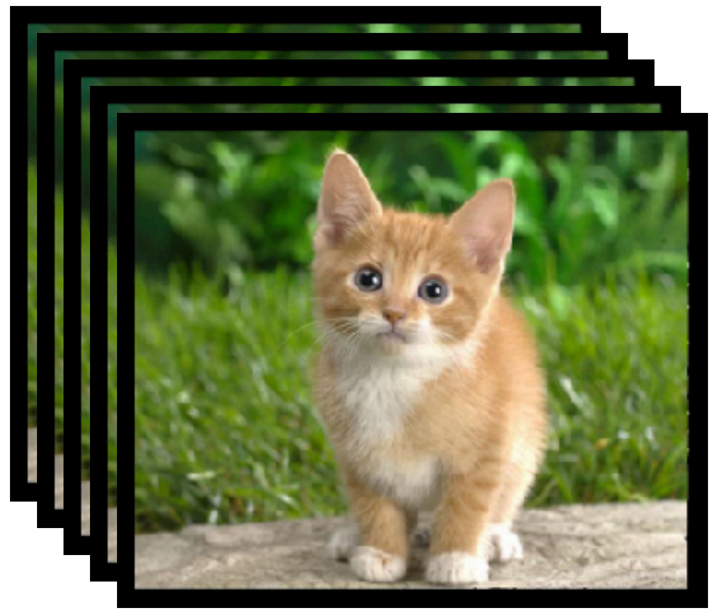




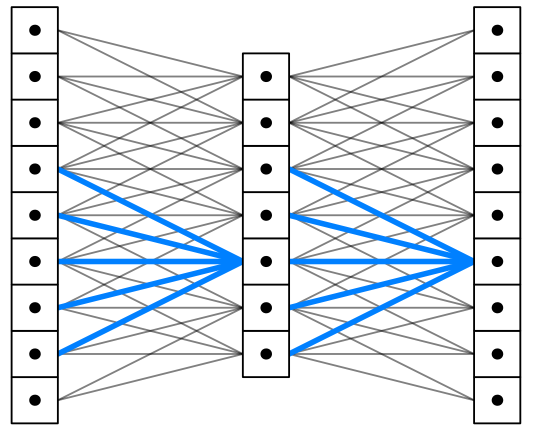
Cat	Cat	Cat	Cat, Dog, Duck	Cat, Dog, Duck

Currently the most successful paradigm: Supervised deep learning

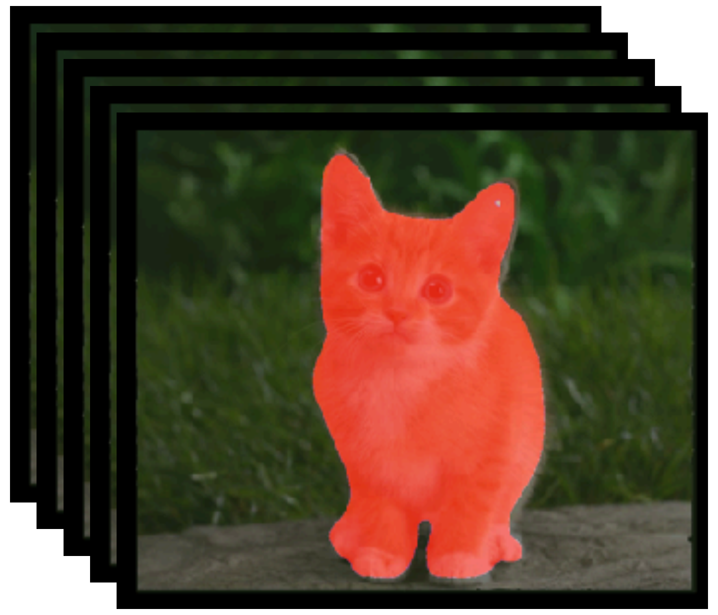
Input



Deep Neural Network



Output



Deep (Convolutional) Neural Networks trained on annotated training data (ground truth, GT)

Computer Vision - Common Problems

Single Object

Image Classification | Object Localization | Dense/Semantic Segmentation

Cat | Cat | Cat

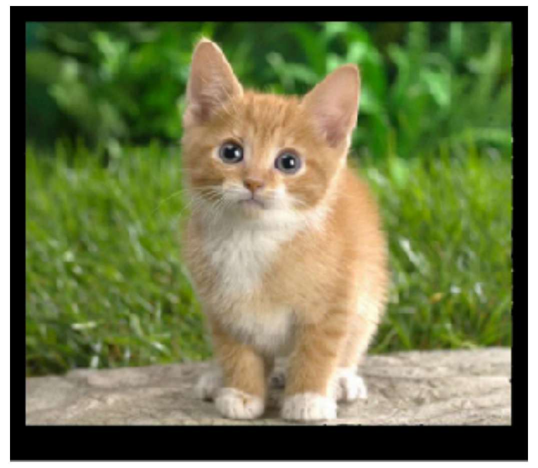
Multiple Objects

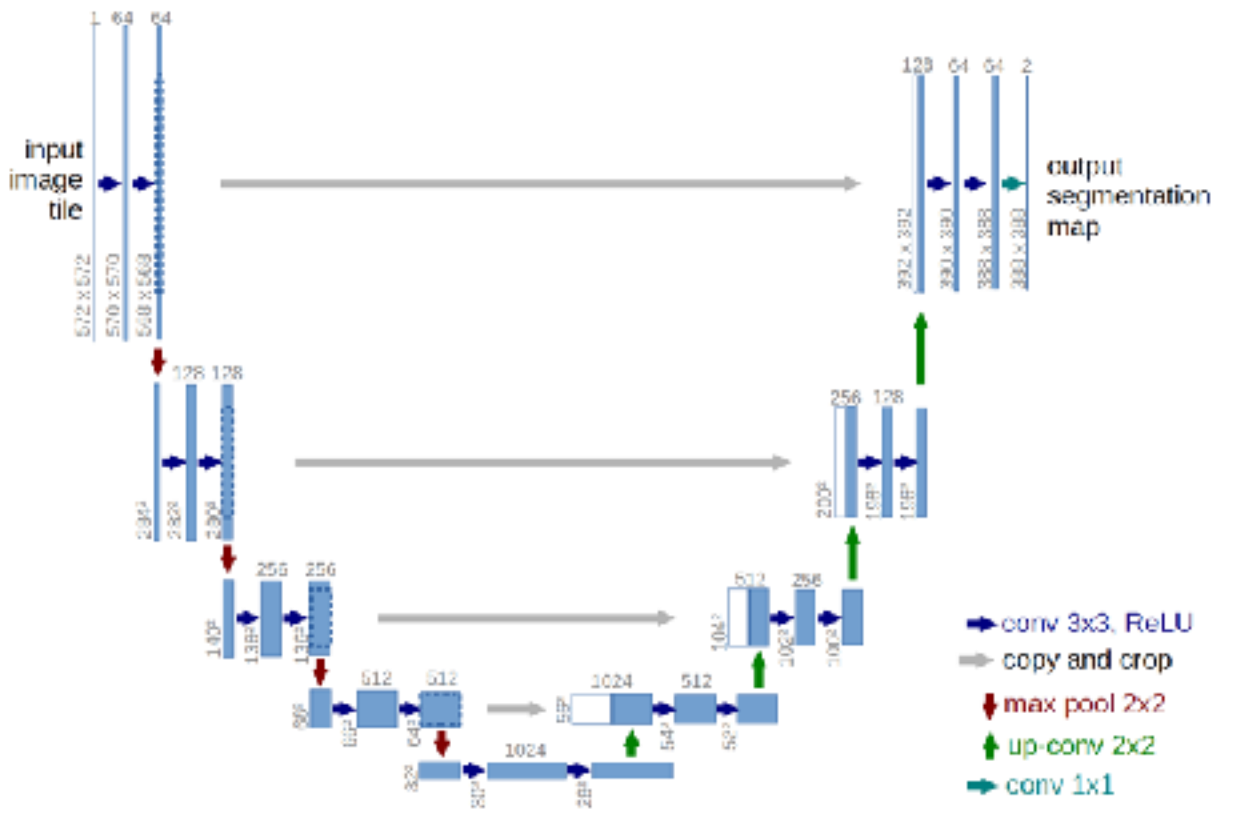
Object Detection | Instance Segmentation

Cat, Dog, Duck | Cat, Dog, Duck

U-Net

Ronneberger et al (2015)





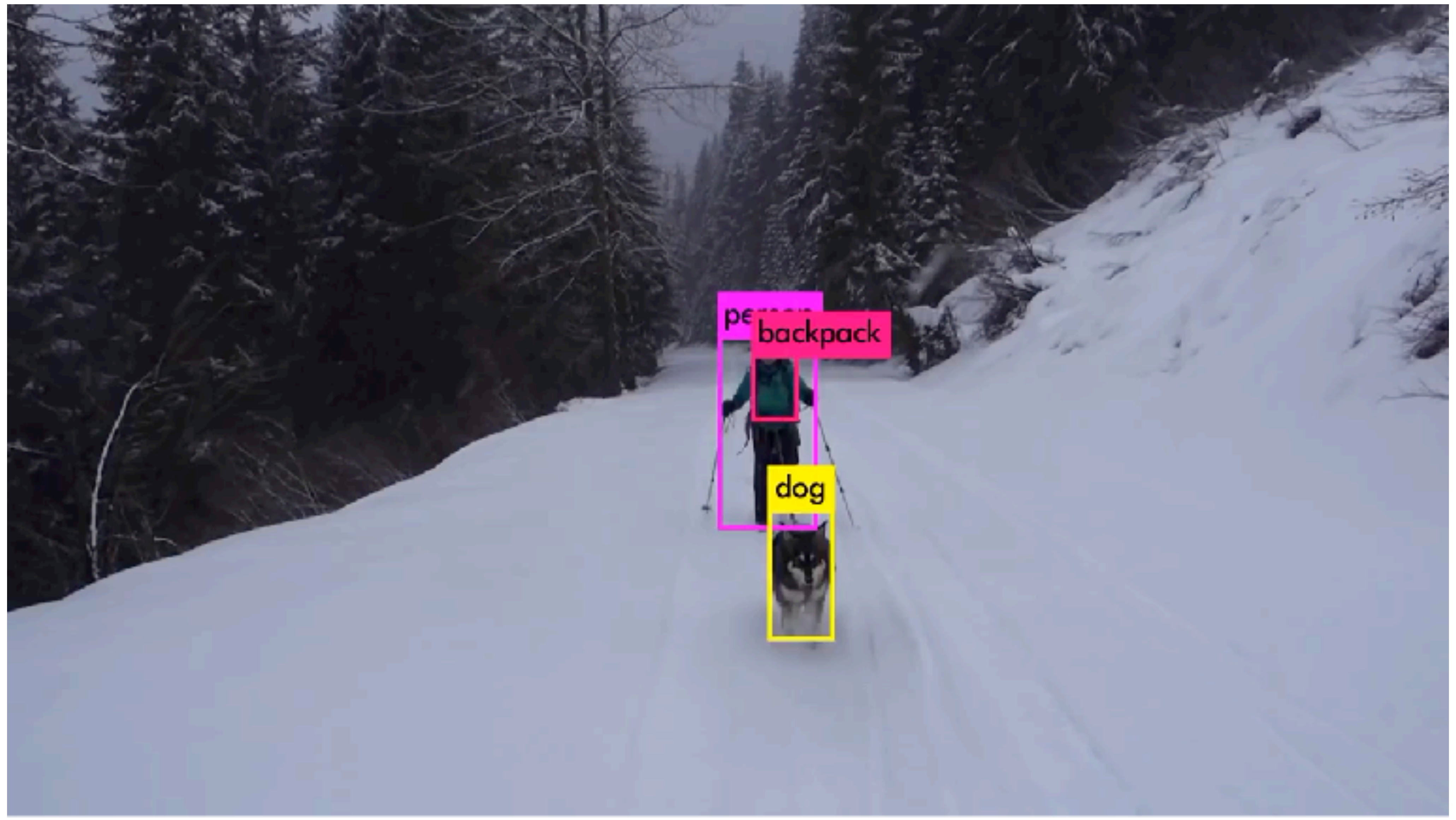
Computer Vision - Common Problems

Single Object			Multiple Objects	
Image Classification	Object Localization	Dense/Semantic Segmentation	Object Detection	Instance Segmentation

Cat	Cat	Cat	Cat, Dog, Duck	Cat, Dog, Duck

YOLO

Redmond et al (2016)



Computer Vision - Common Problems

Single Object

Image Classification	Object Localization	Dense/Semantic Segmentation

Cat	Cat	Cat

Multiple Objects

Object Detection	Instance Segmentation

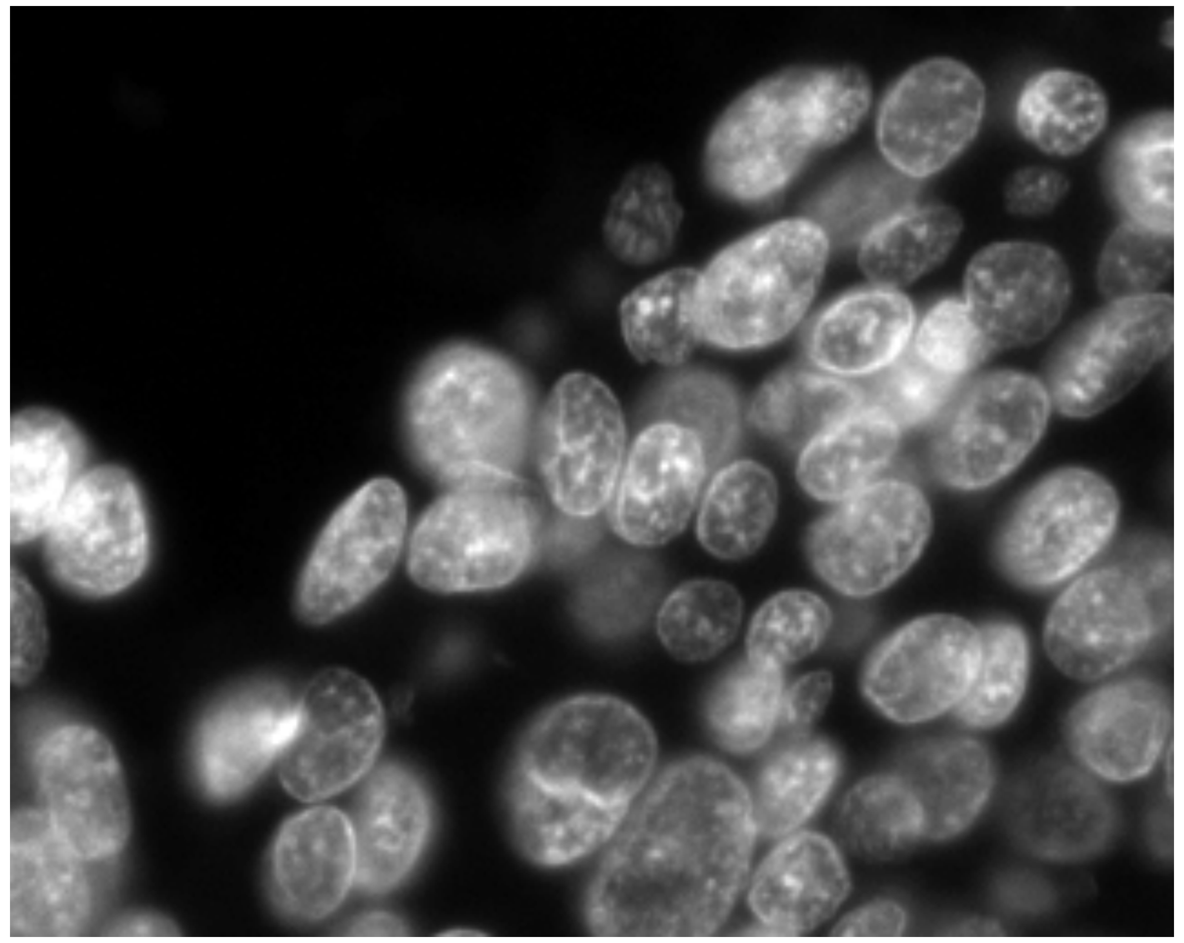
Cat, Dog, Duck	Cat, Dog, Duck

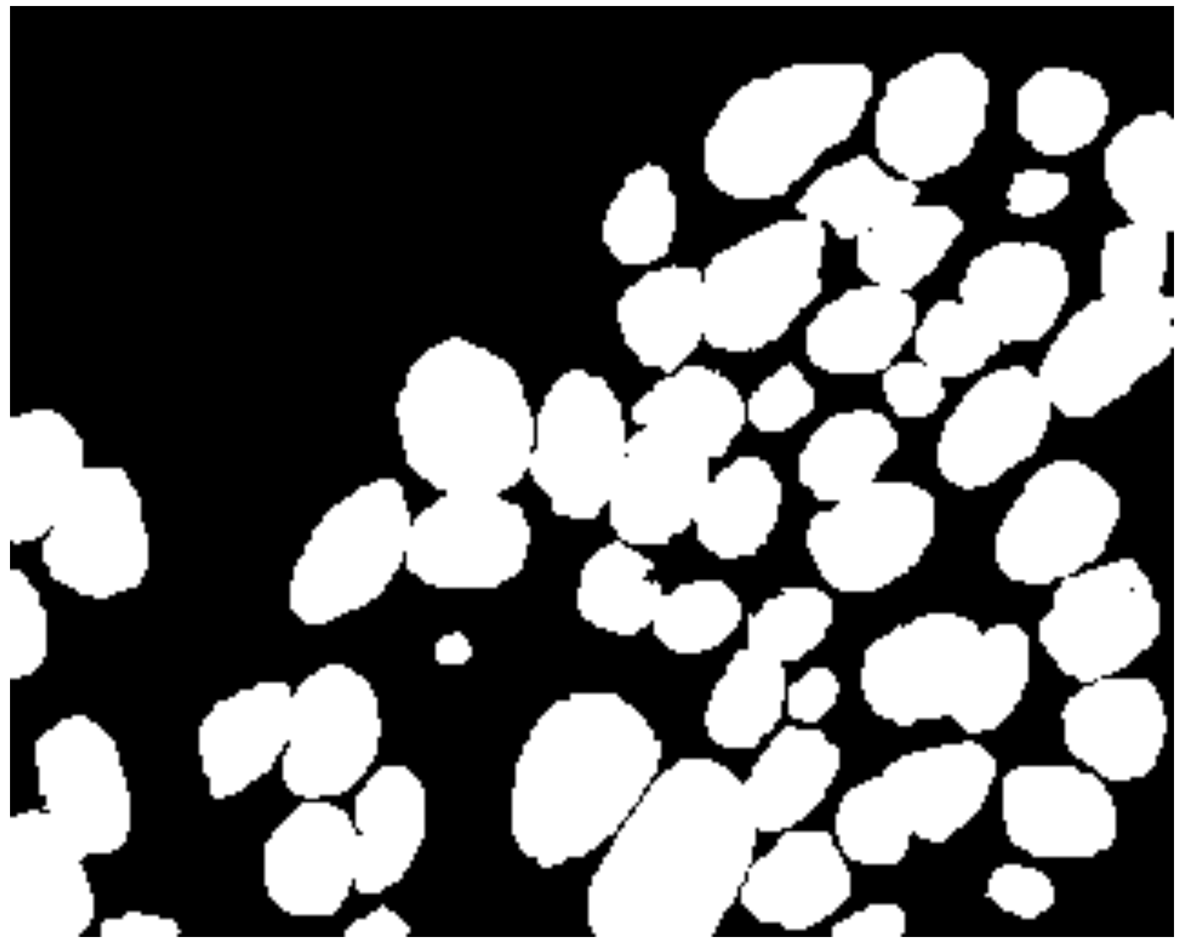
Mask-RCNN

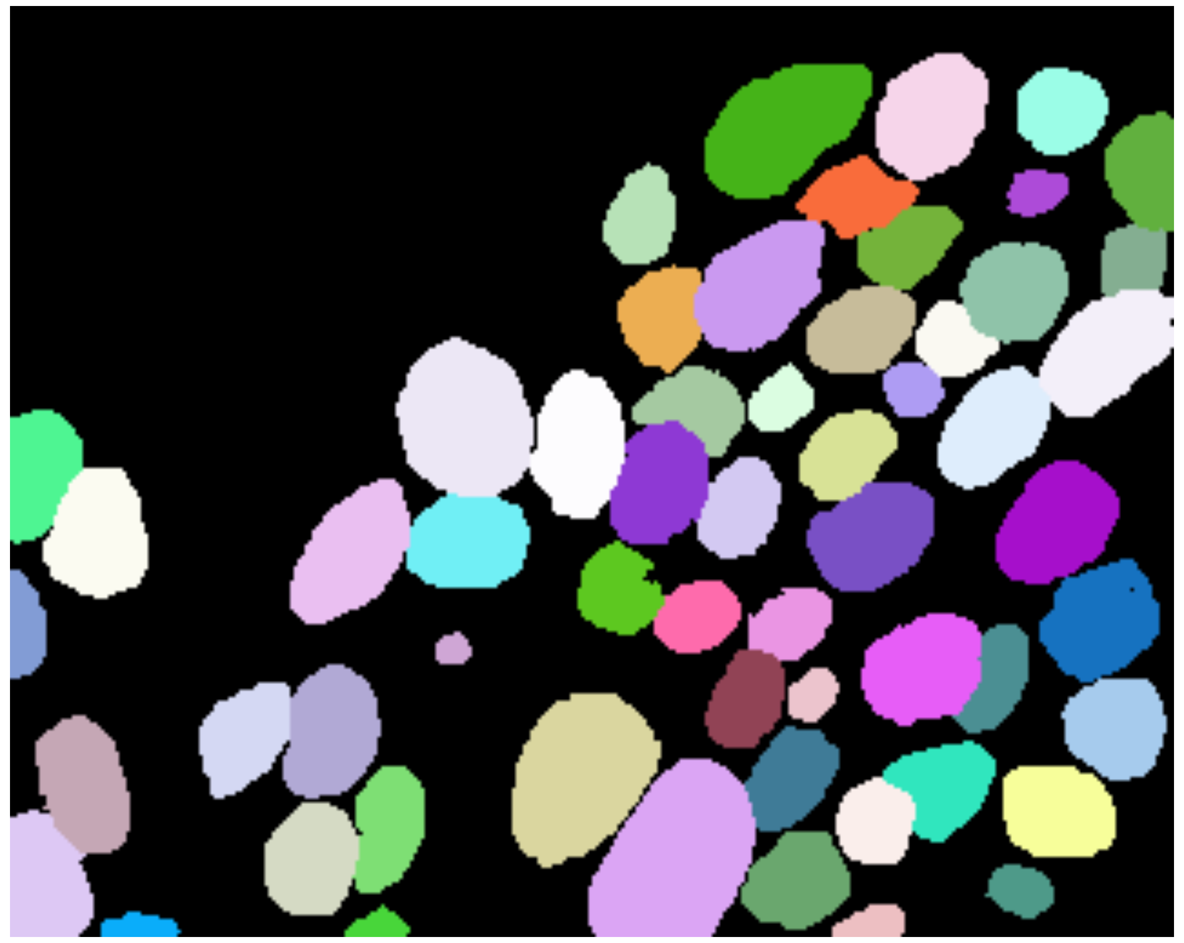
He et al (2017)

Our Problem: Nuclei Segmentation

Input Image → Dense Segmentation (foreground/background) → Instance Segmentation (individual cells)

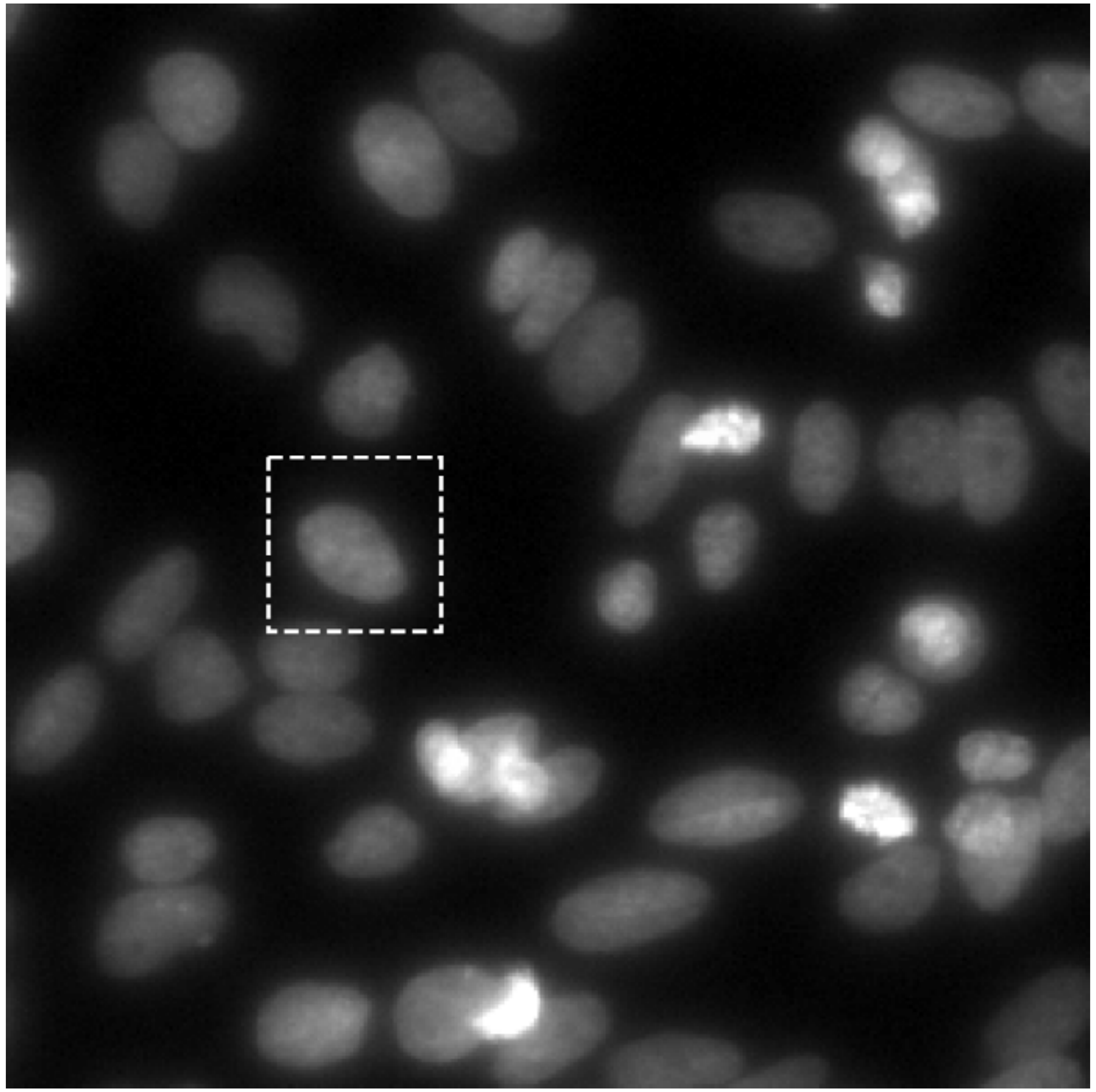


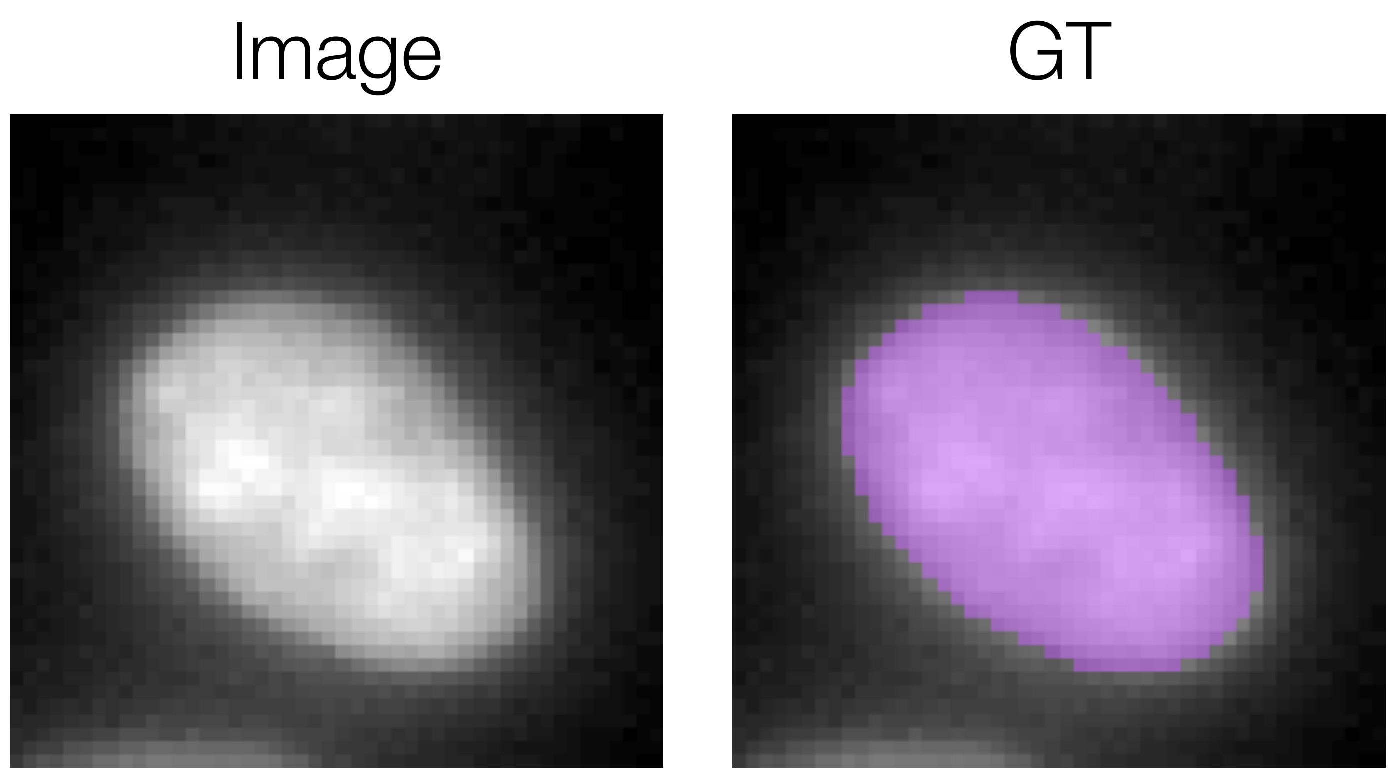




Challenges: many crowded objects, noisy images

Common (DL) approaches for nuclei segmentation



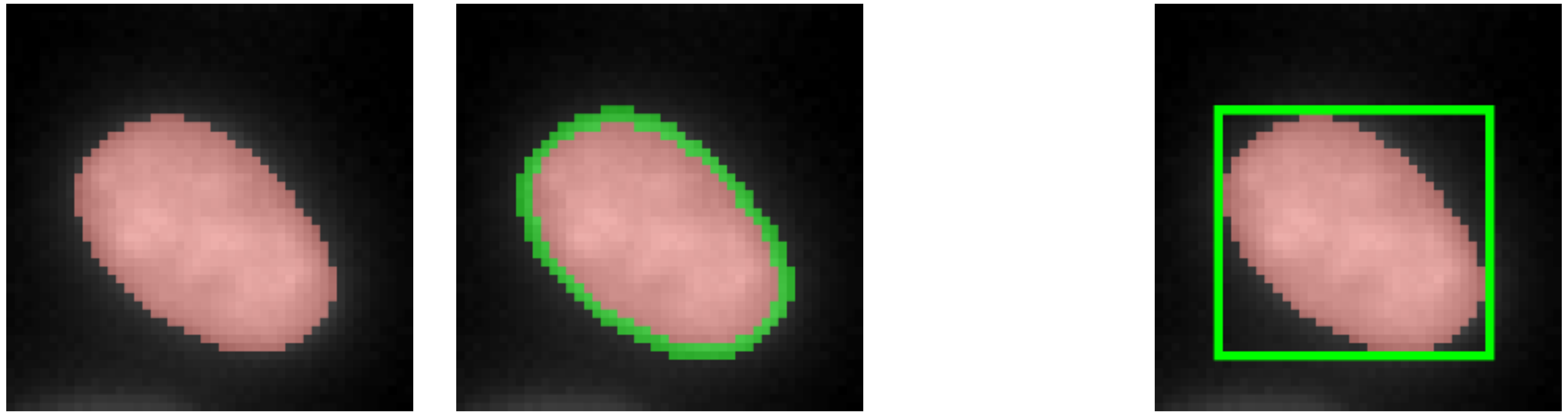


Bottom-Up

First segment, then localize

Top-Down

First localize, then segment



2 Class U-Net

Ronneberger et al. (2015)

3 Class U-Net

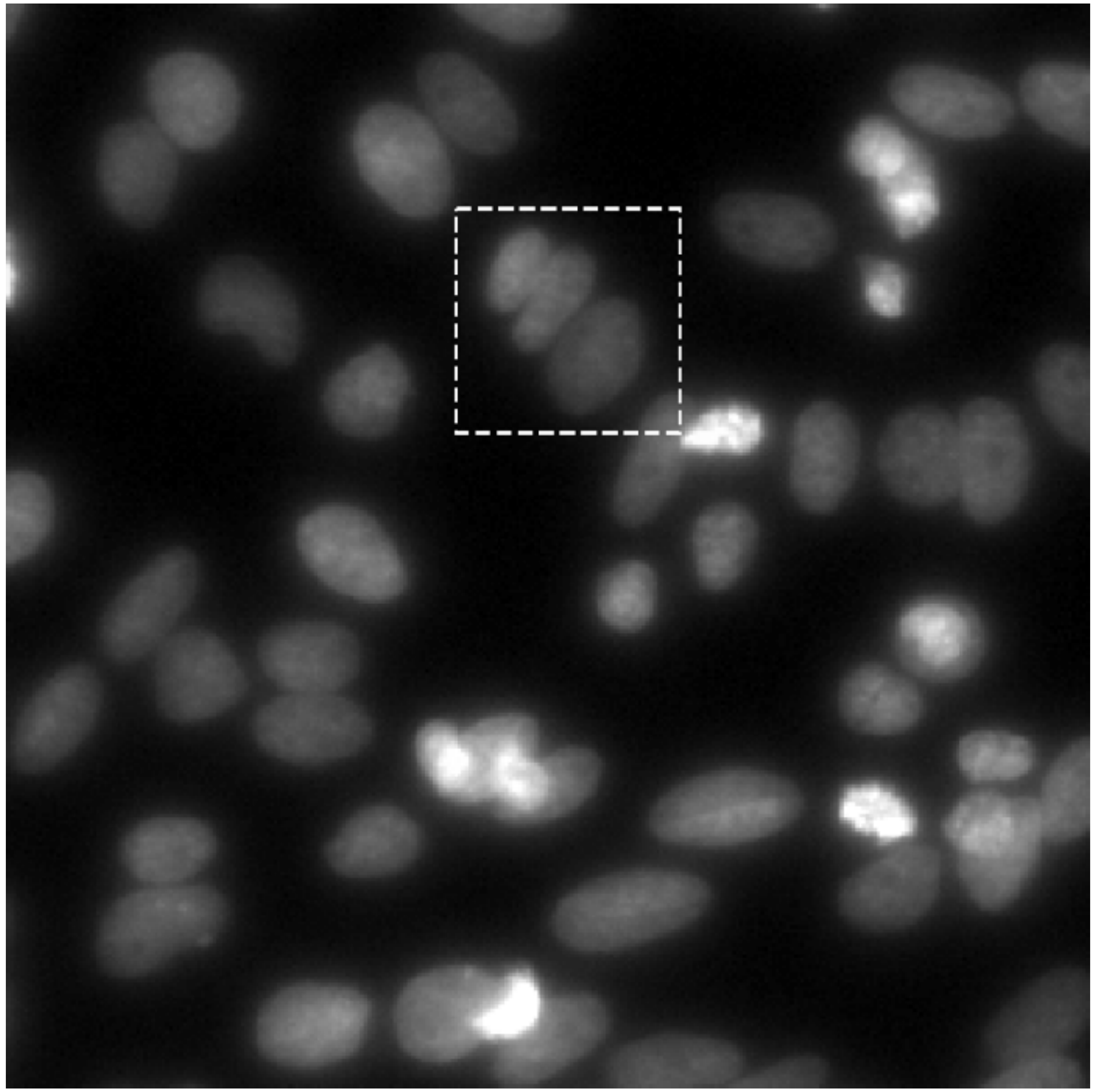
Caicedo et al.(2019)

Mask-RCNN

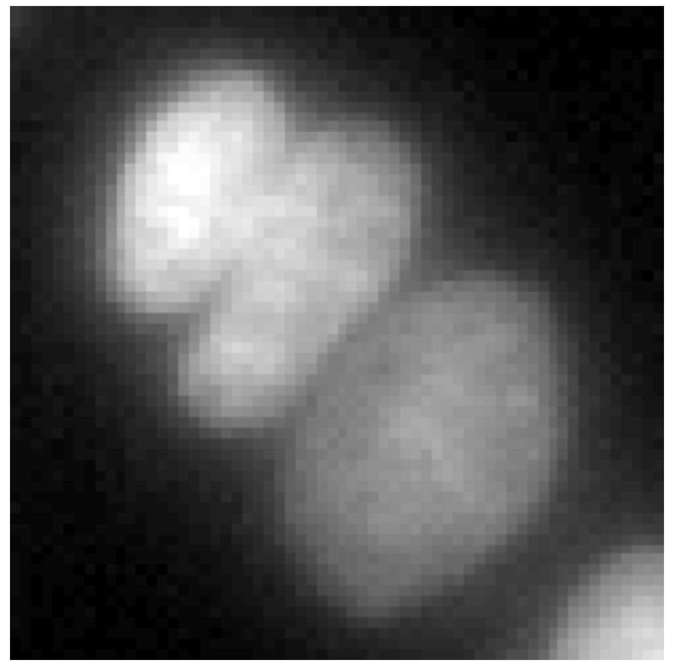
He et al (2017)

- Affinity based methods Wolf, Pape et al. (2018), Hirsch et al. (2020)
- Embedding based methods Neven et al. (2019), Stringer et al. (2020)

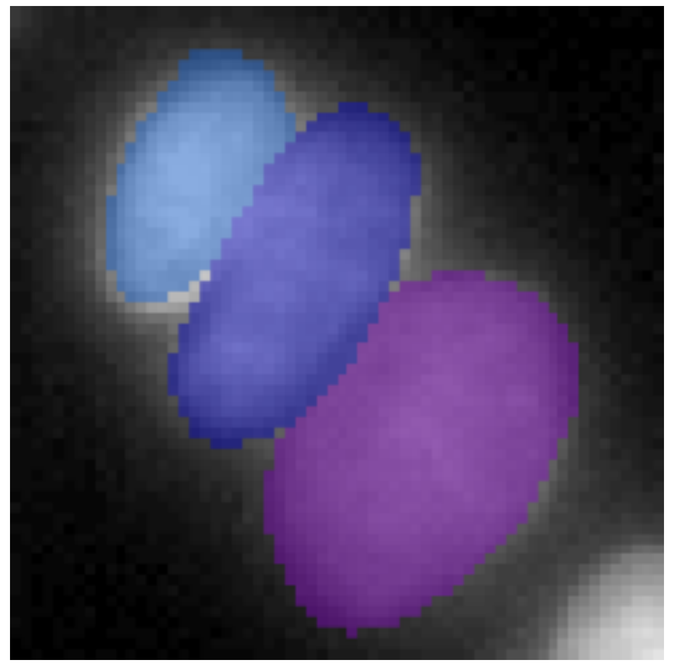
Problems for crowded objects



Image

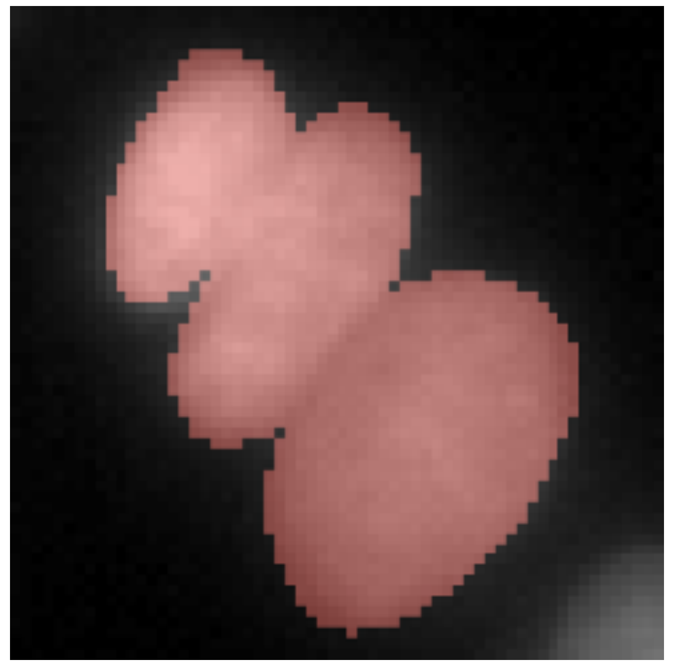


GT

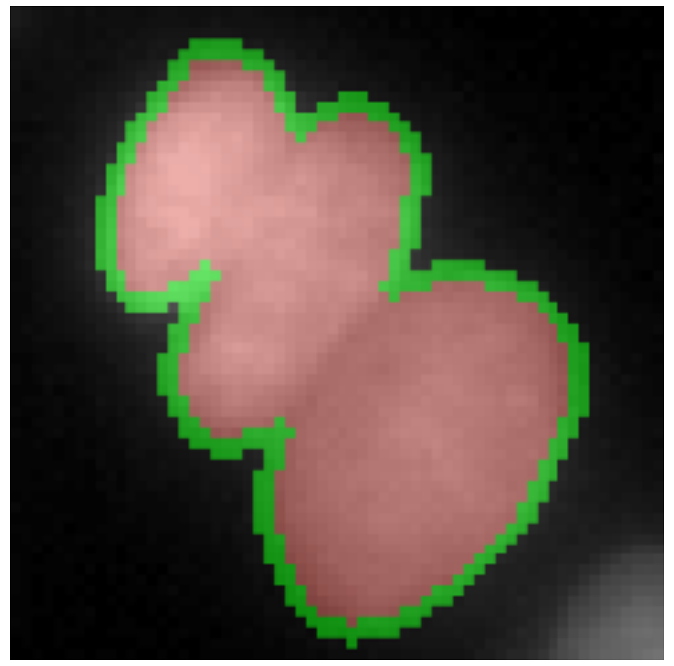


Bottom-Up

First segment, then localize



2 Class U-Net



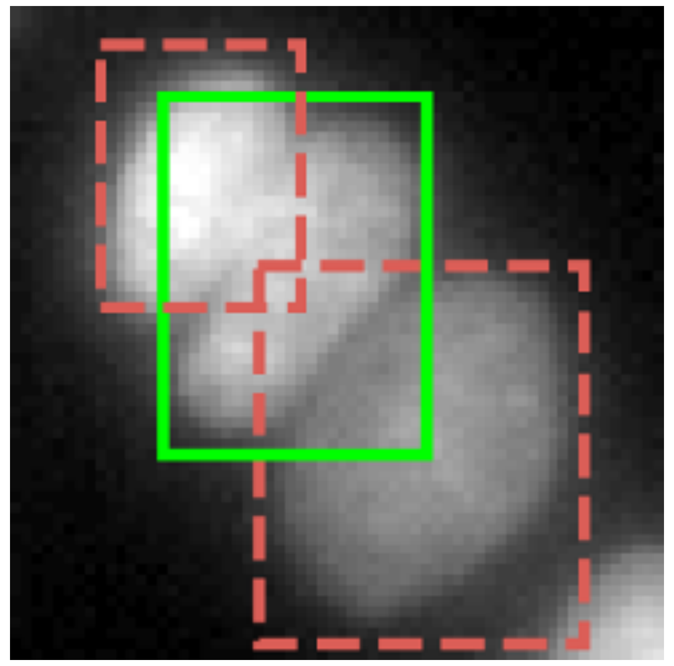
3 Class U-Net

Mislocalization

Fused segmentation maps

Top-Down

First localize, then segment

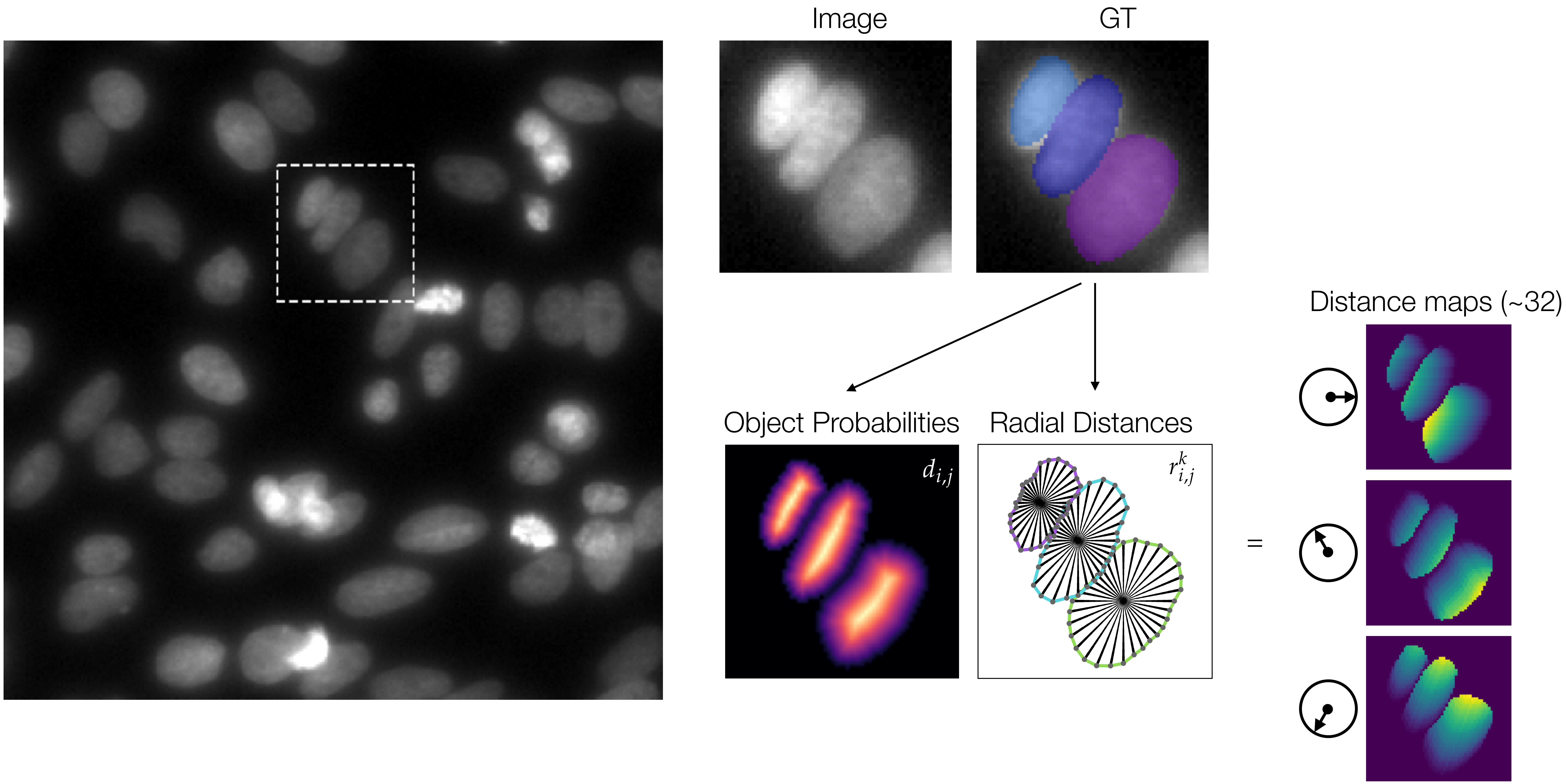


Mask-RCNN

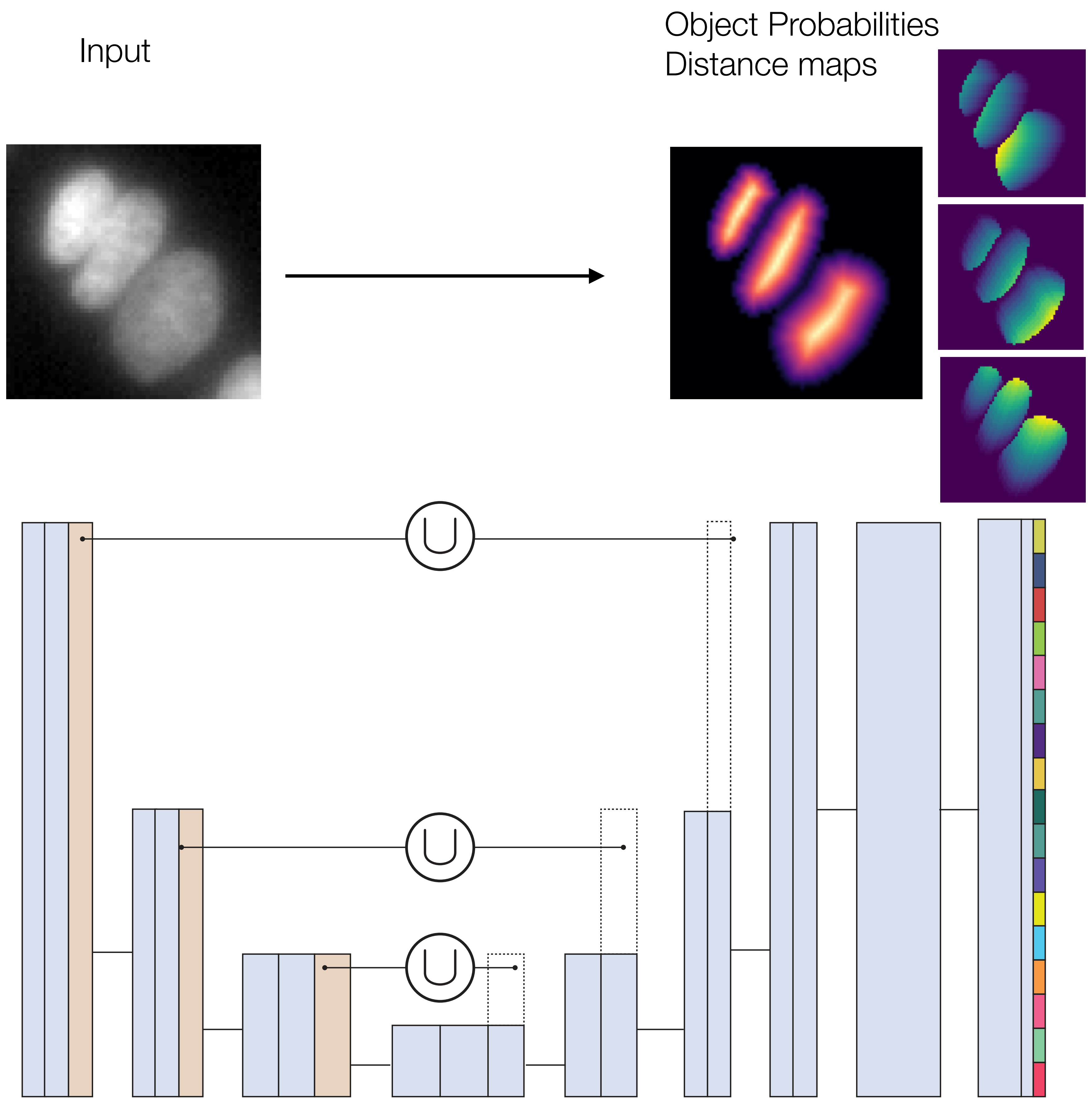
Mislocalization

Bounding box overlap > threshold

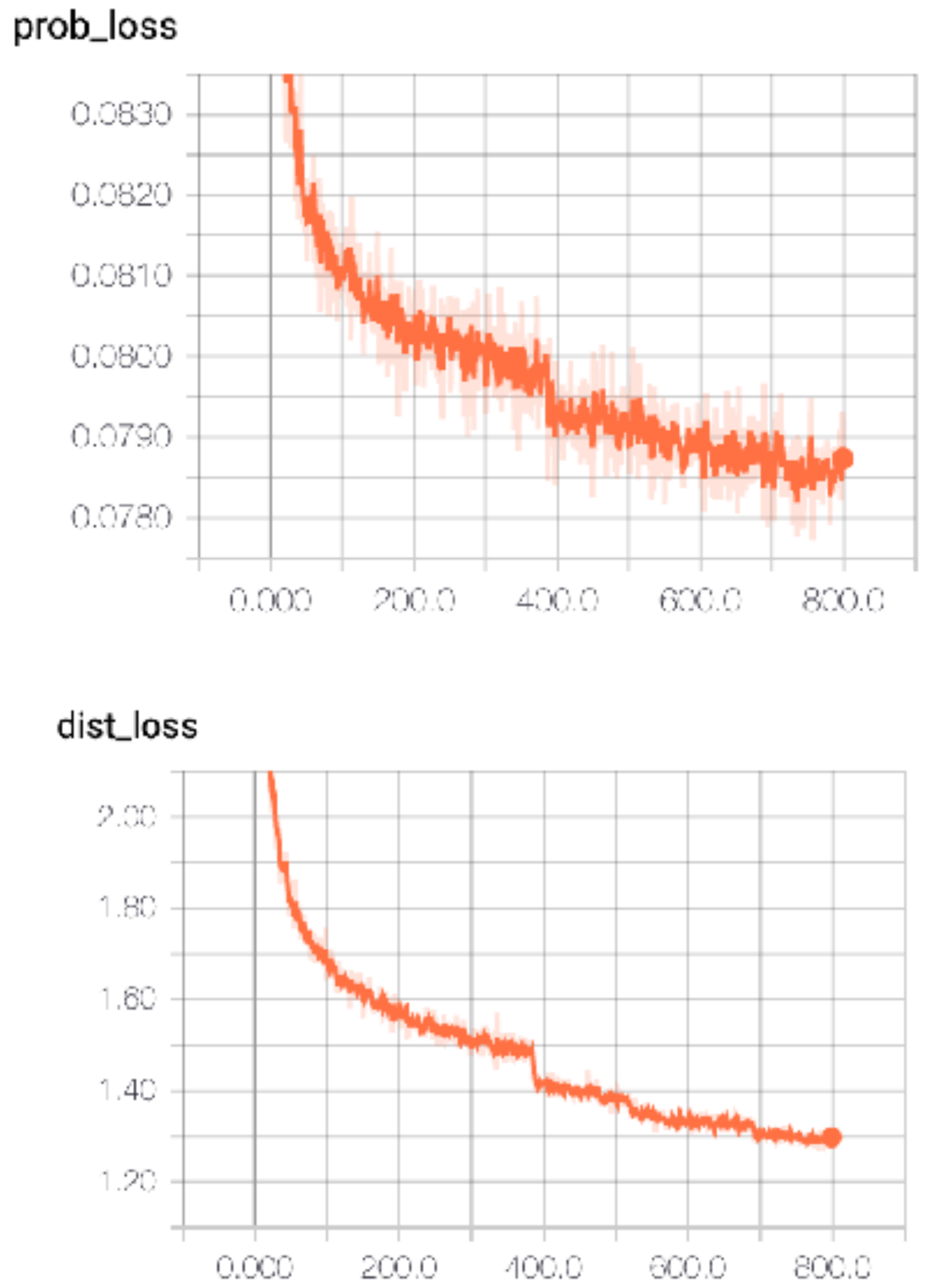
StarDist: Principle



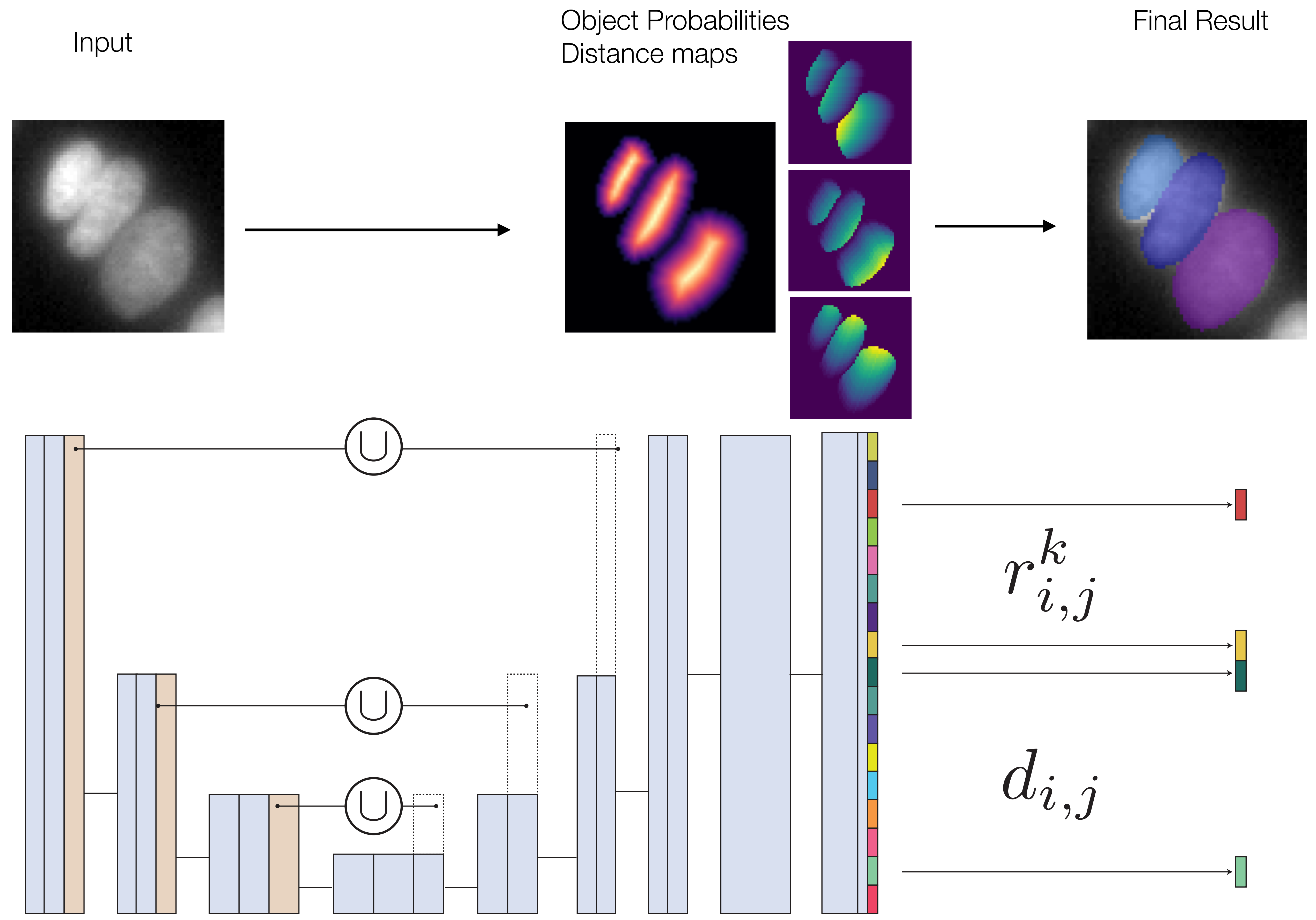
StarDist: Principle



Step 1
Dense Polygon Prediction
via a CNN (e.g. U-Net, ResNet)



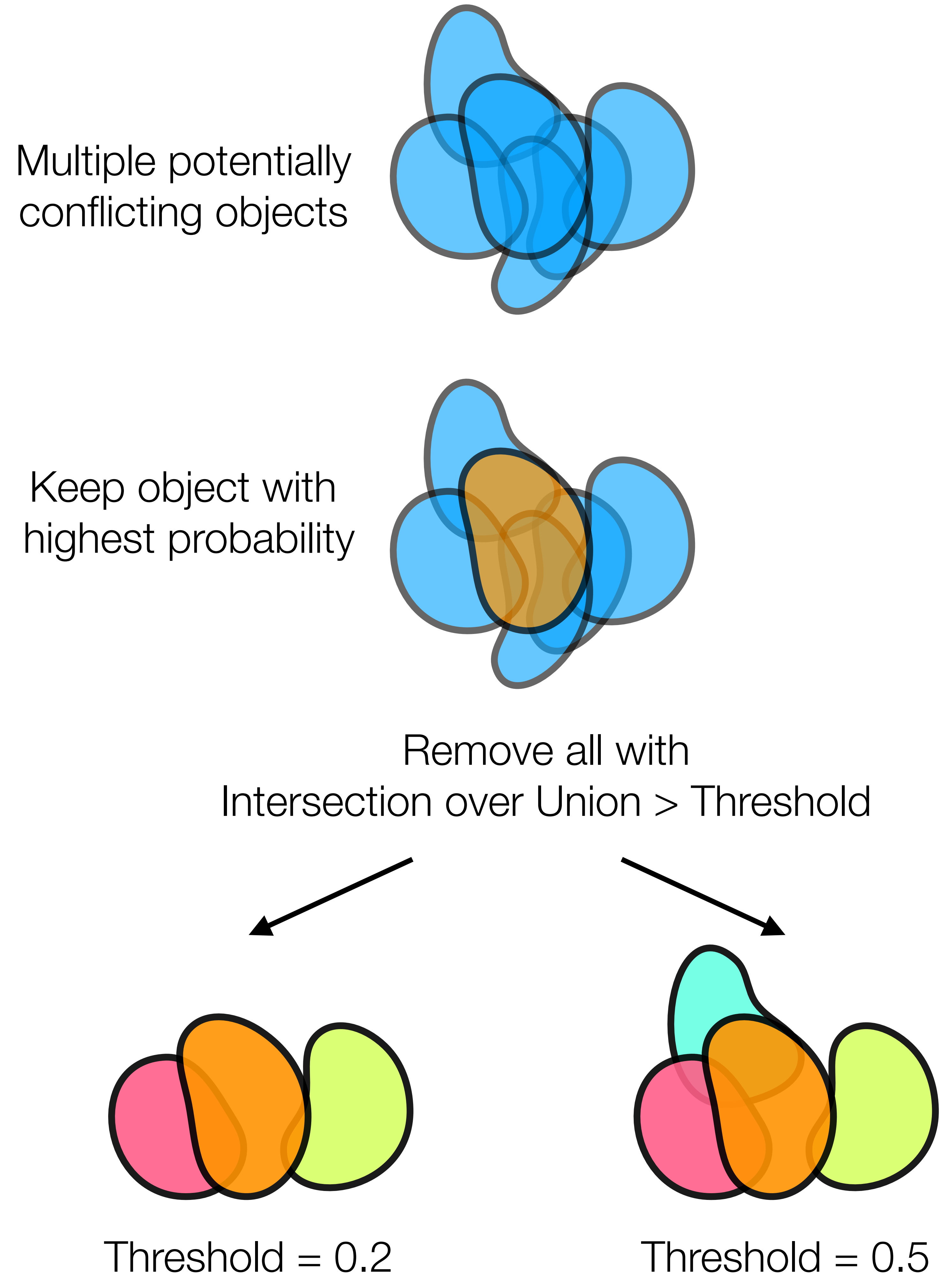
StarDist: Principle



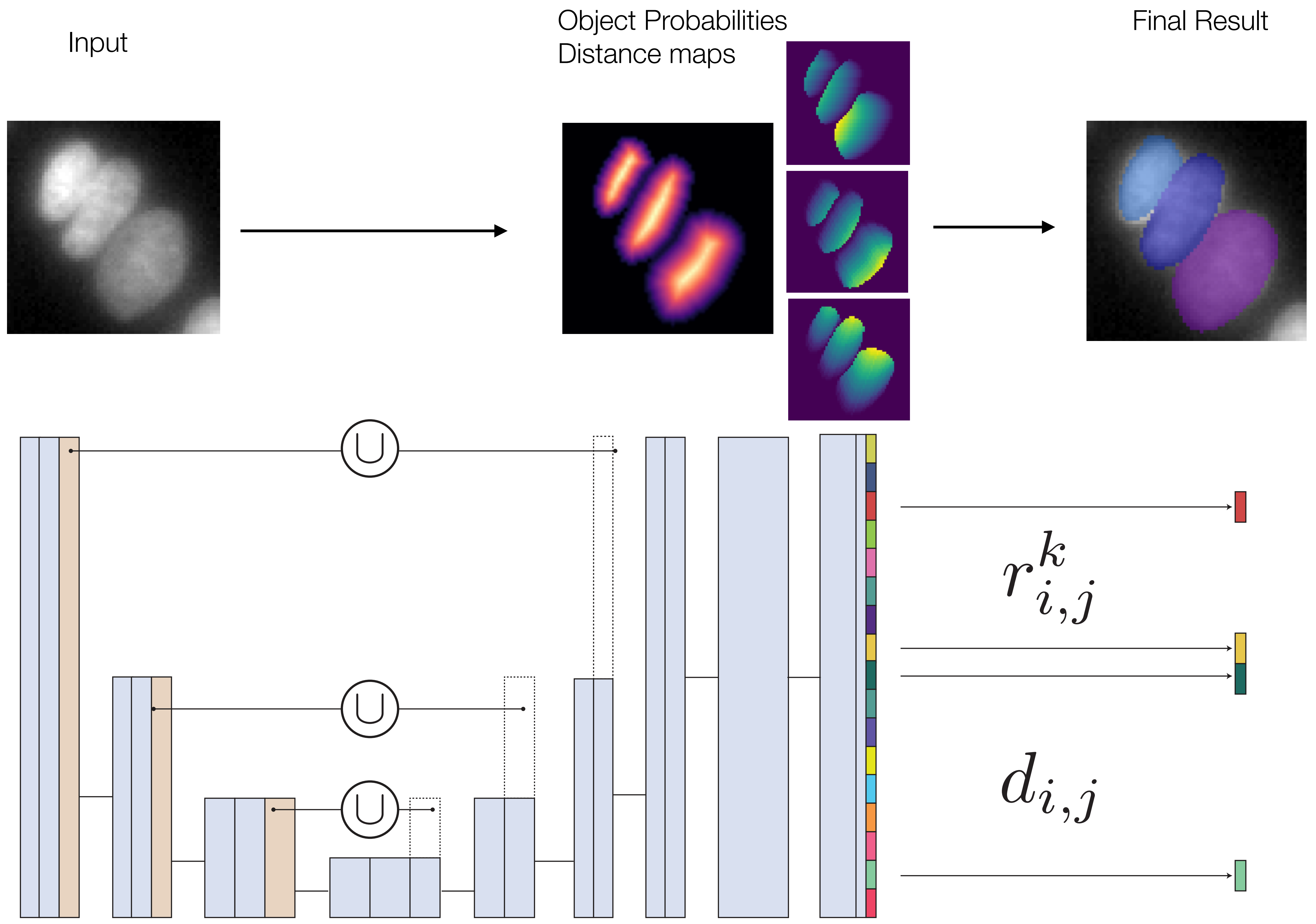
Step 1
Dense Polygon Prediction
(e.g. U-Net, ResNet)

Step 2
Polygon Selection
(Non-Maximum Suppression NMS)

Non-Maximum-Suppression (NMS)



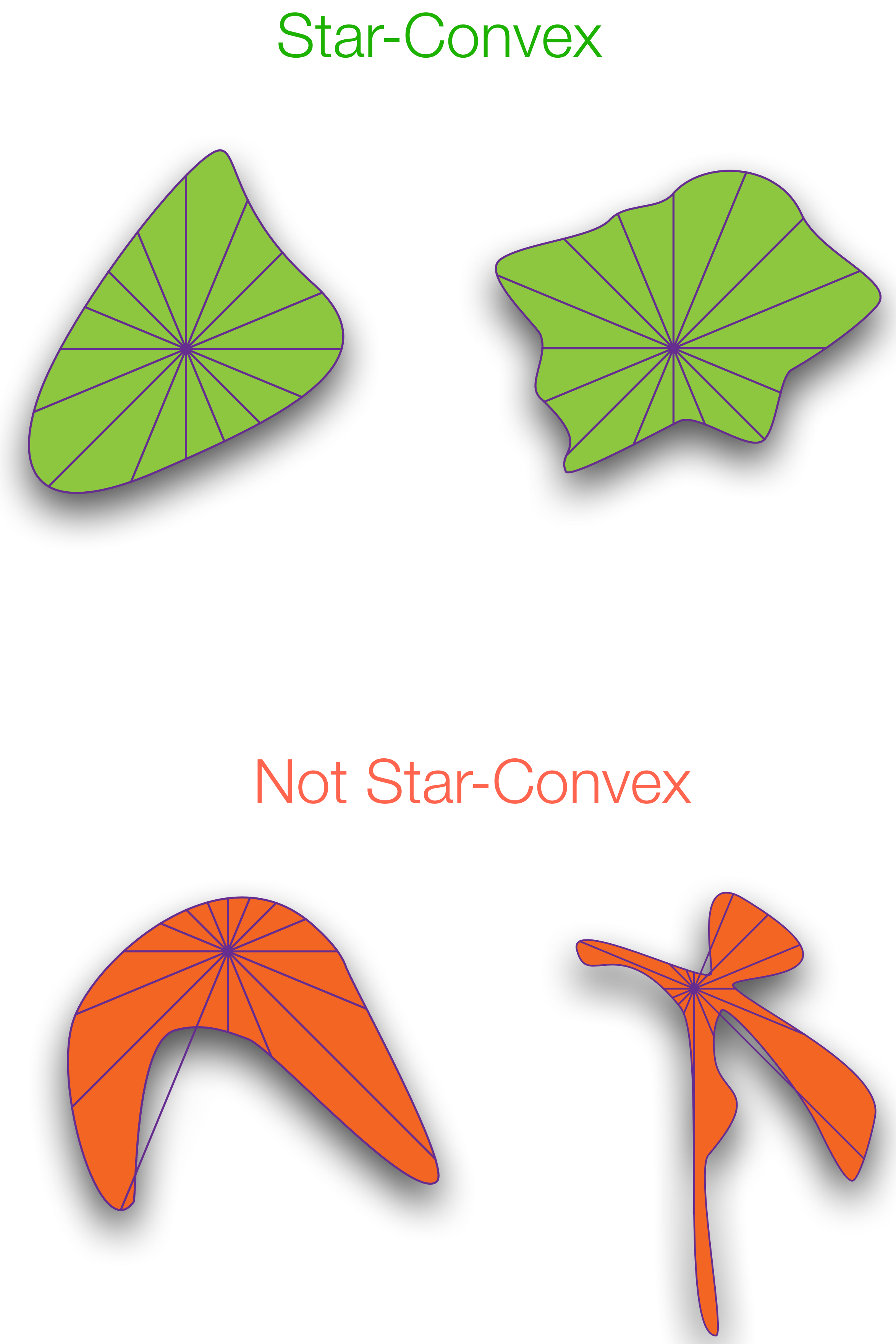
StarDist: Principle



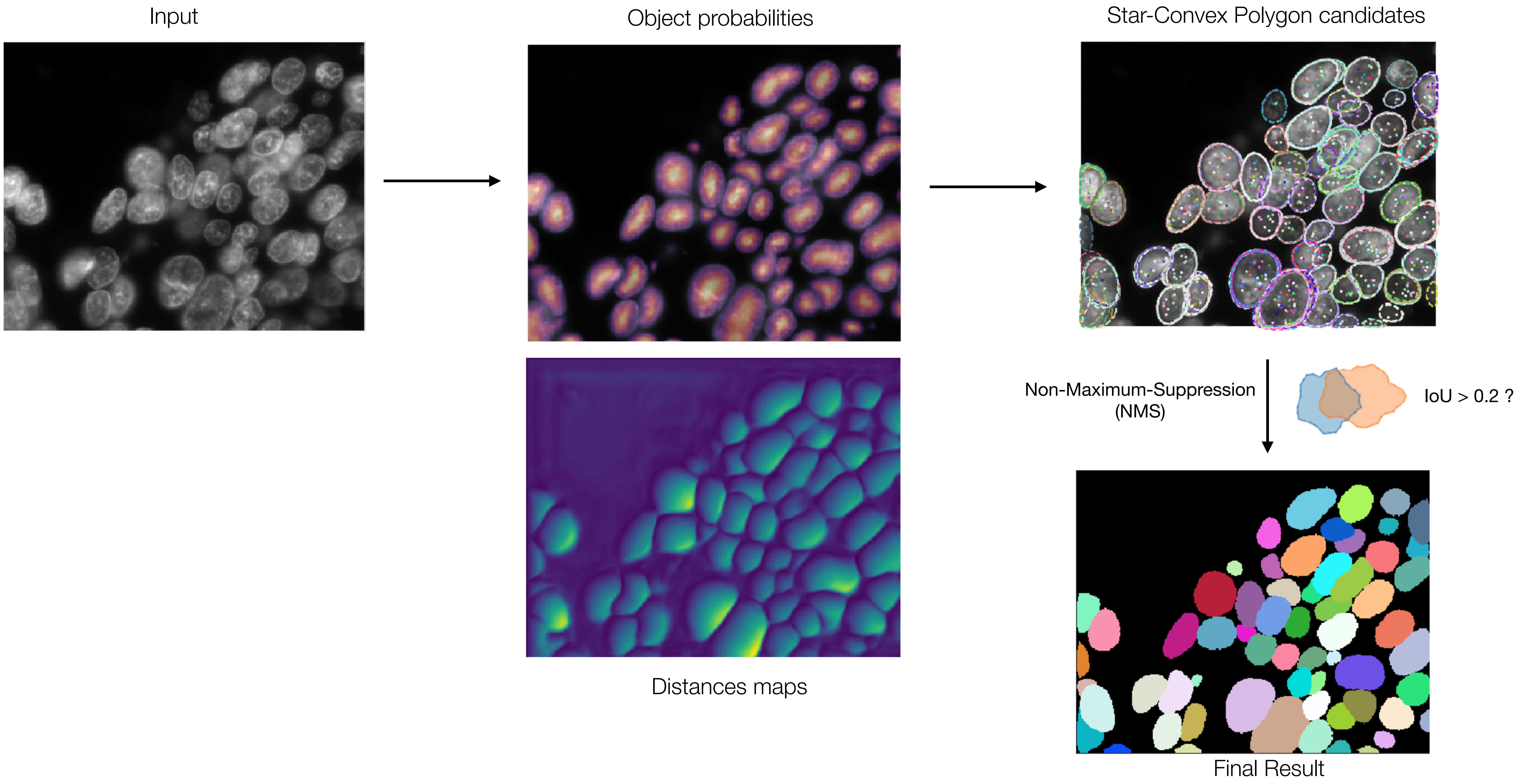
Step 1
Dense Polygon Prediction
(e.g. U-Net, ResNet)

Step 2
Polygon Selection
(Non-Maximum Suppression NMS)

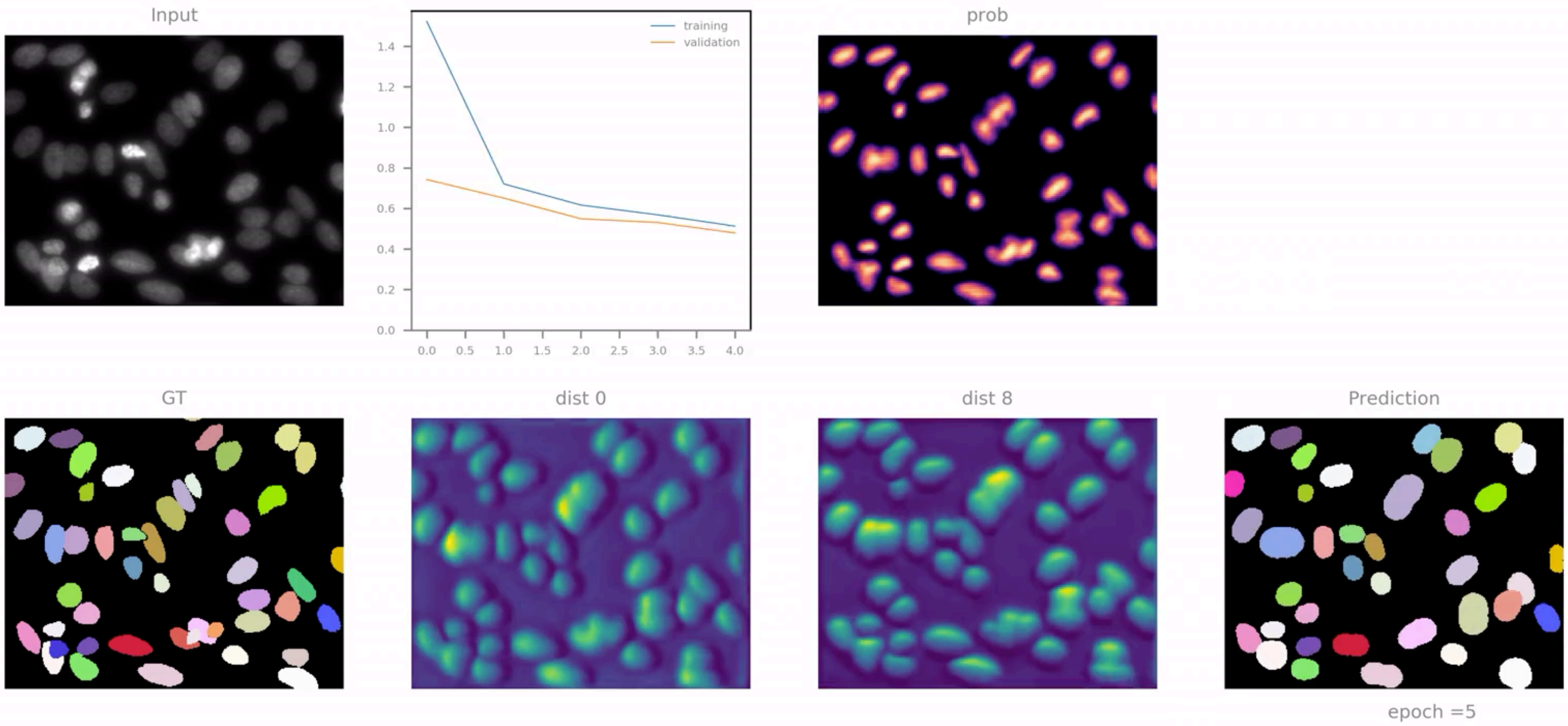
Important:
Assumes objects are *star-convex*



StarDist: Example

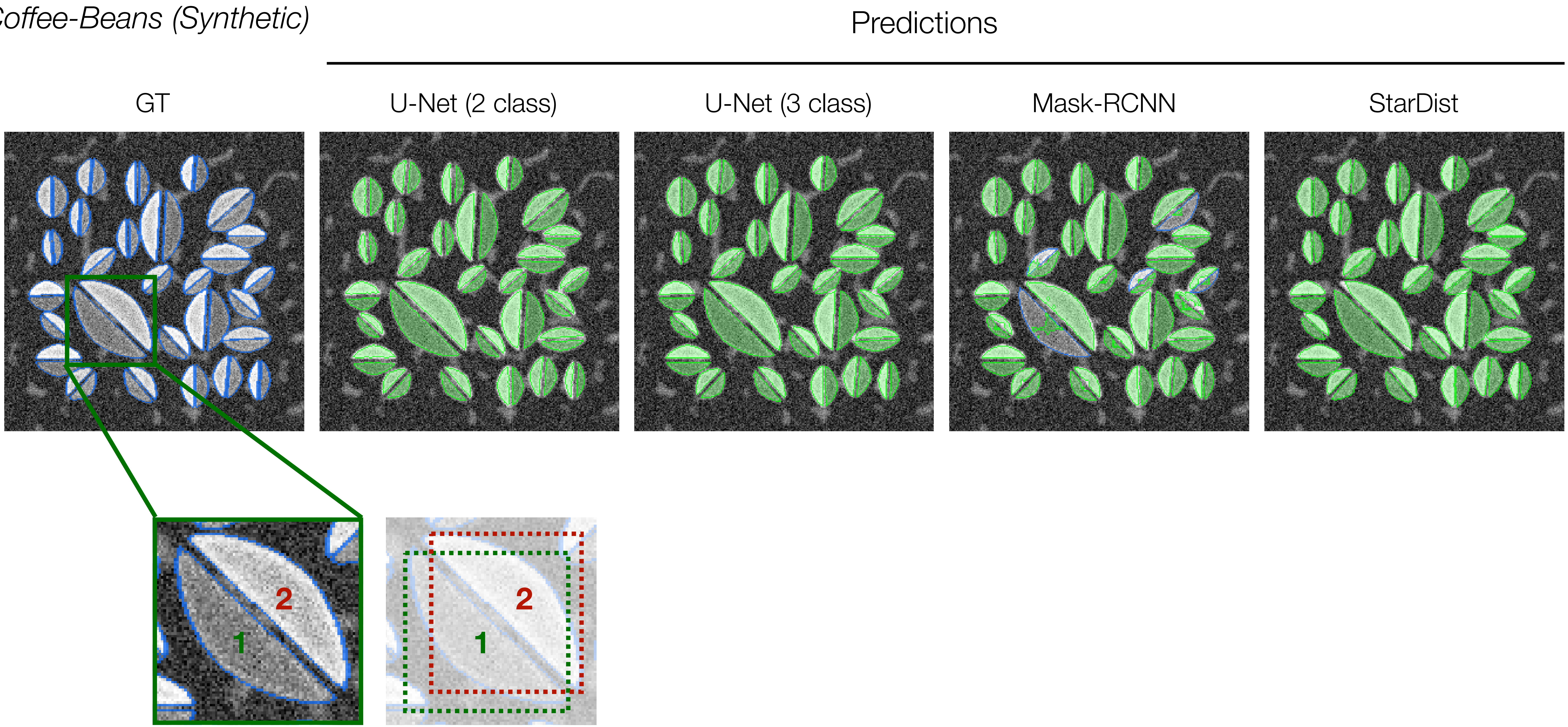


Training process



Comparison with common methods

Coffee-Beans (Synthetic)

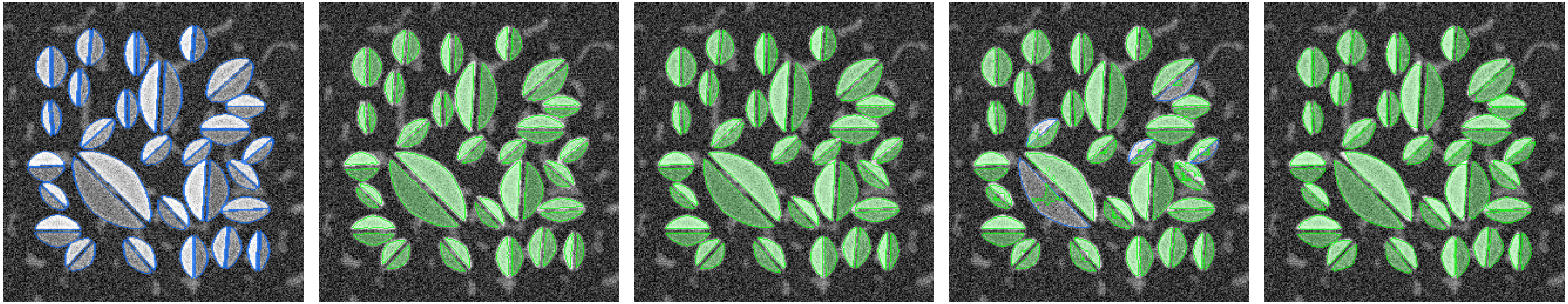


Comparison with common methods

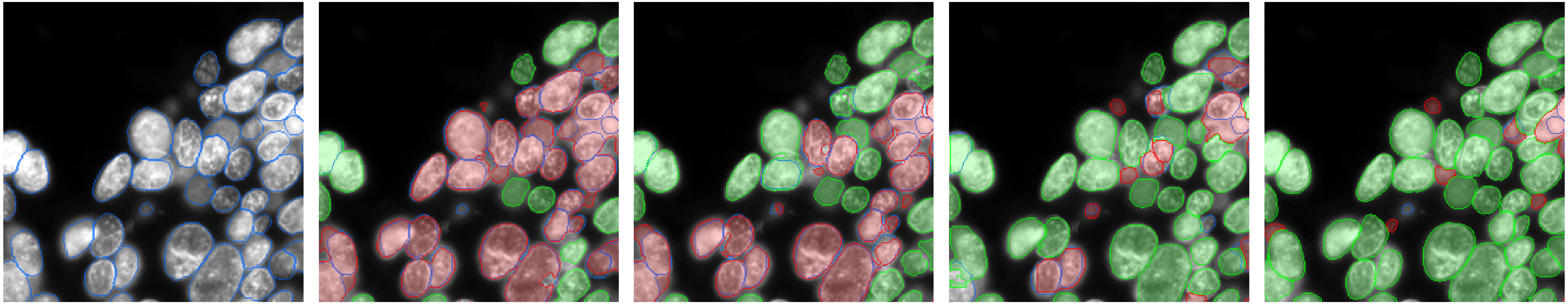
Coffee-Beans (Synthetic)

Predictions

GT | U-Net (2 class) | U-Net (3 class) | Mask-RCNN | StarDist

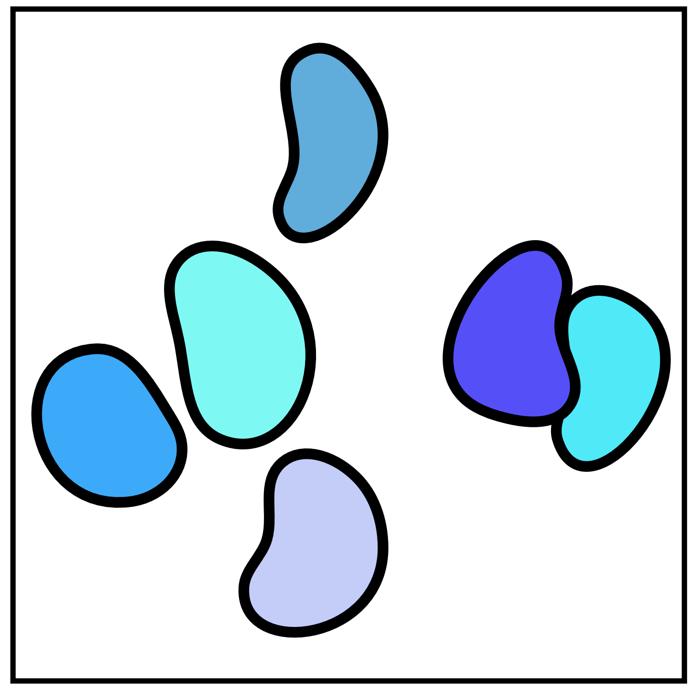


DSB 2018 (Kaggle Challenge) Caicedo et al (2019)

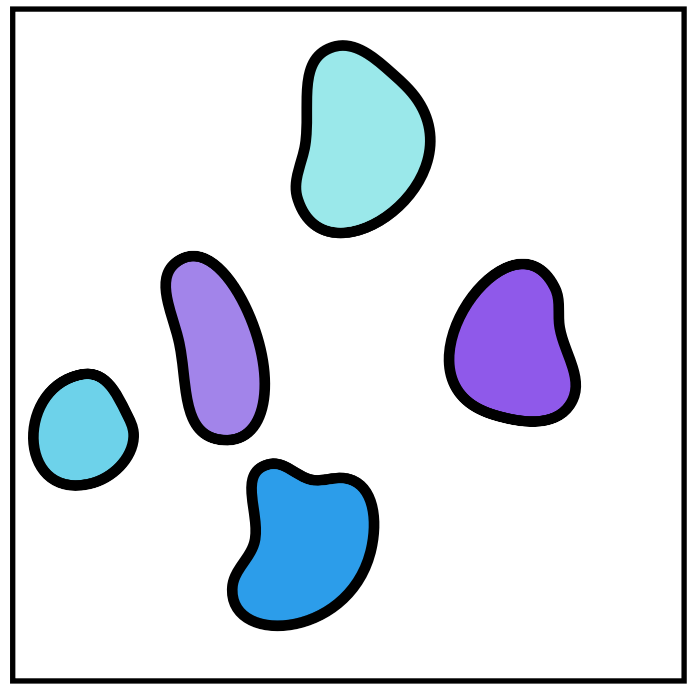


Accuracy Measures

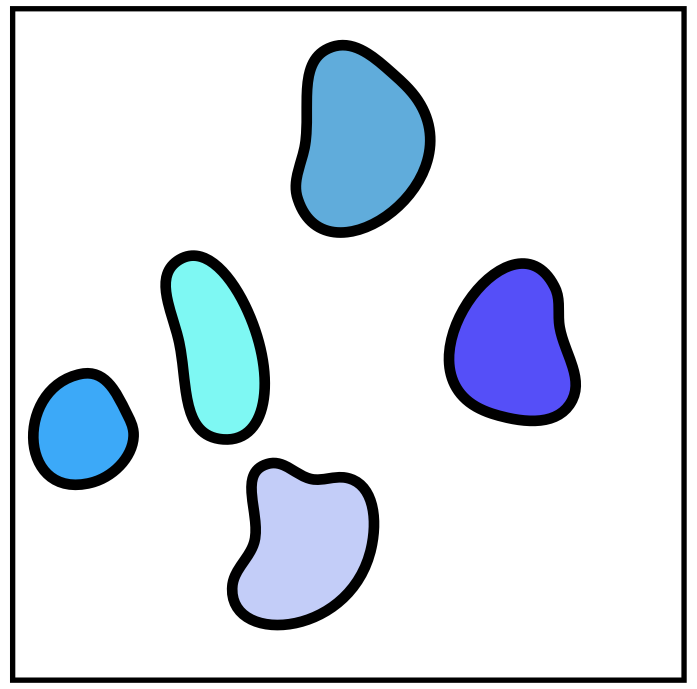
GT



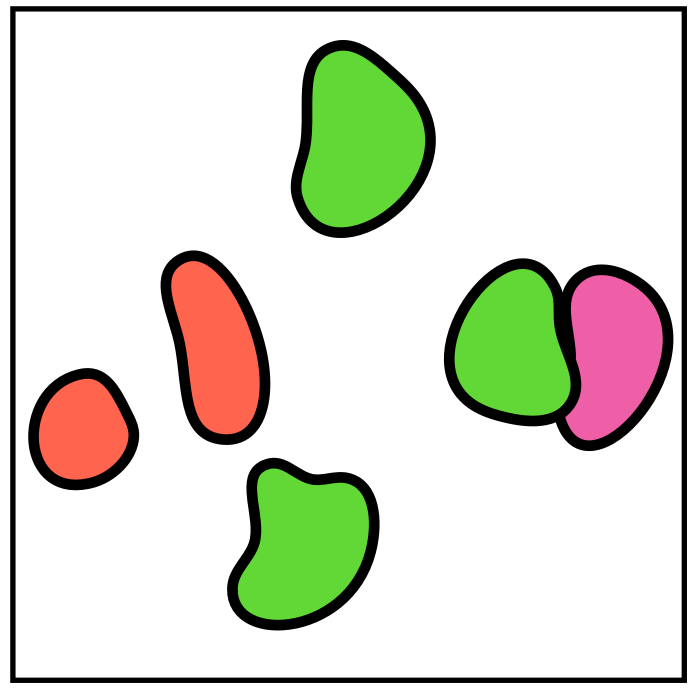
Prediction



IoU Matched Prediction

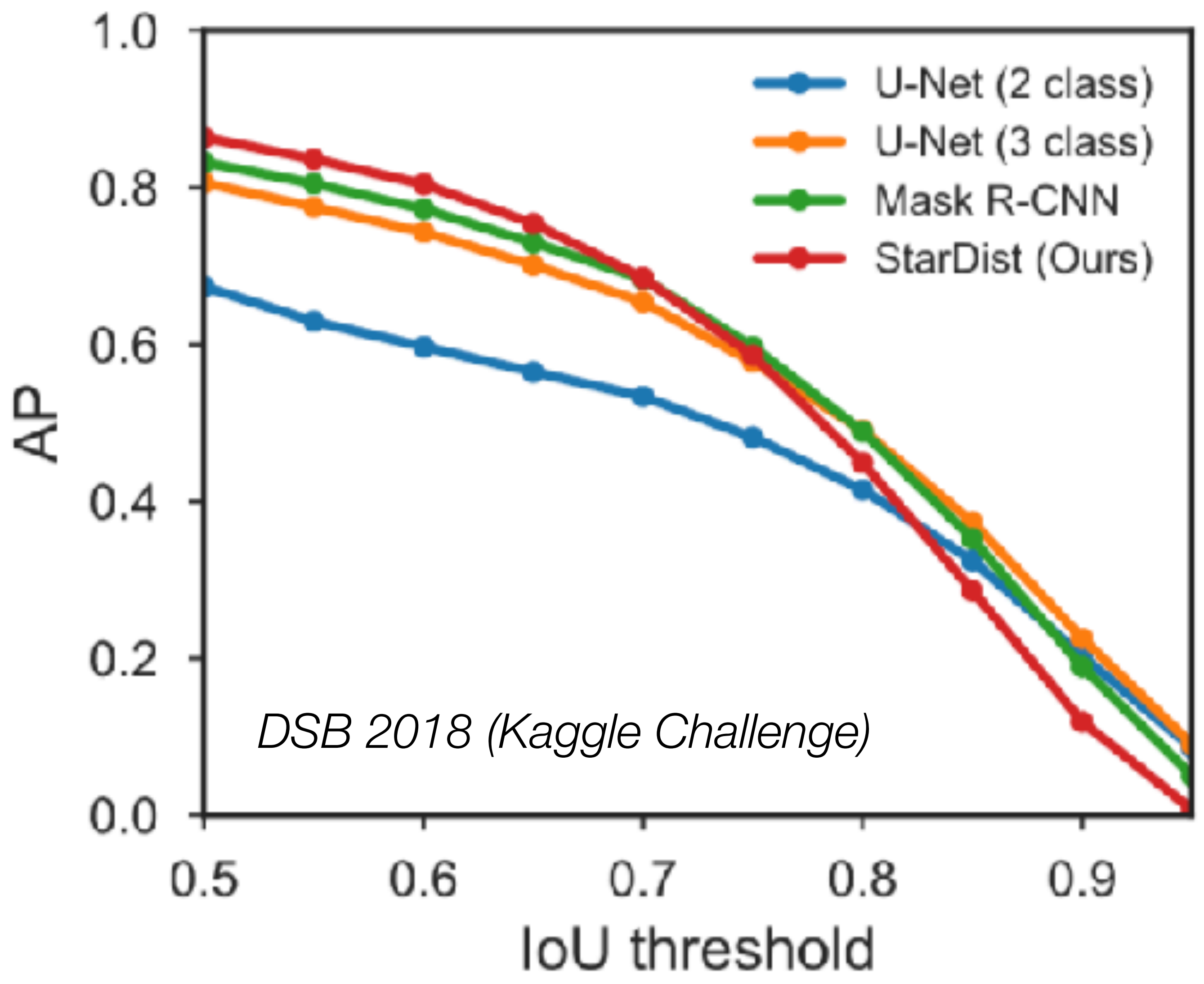


IoU > Threshold $\in (0,1)$



True Positive TP
False Positive FP
False Negative FN

low threshold: already slightly overlaps are counted as TP (high AP)
high threshold: only almost complete overlaps are counted as TP (low AP)

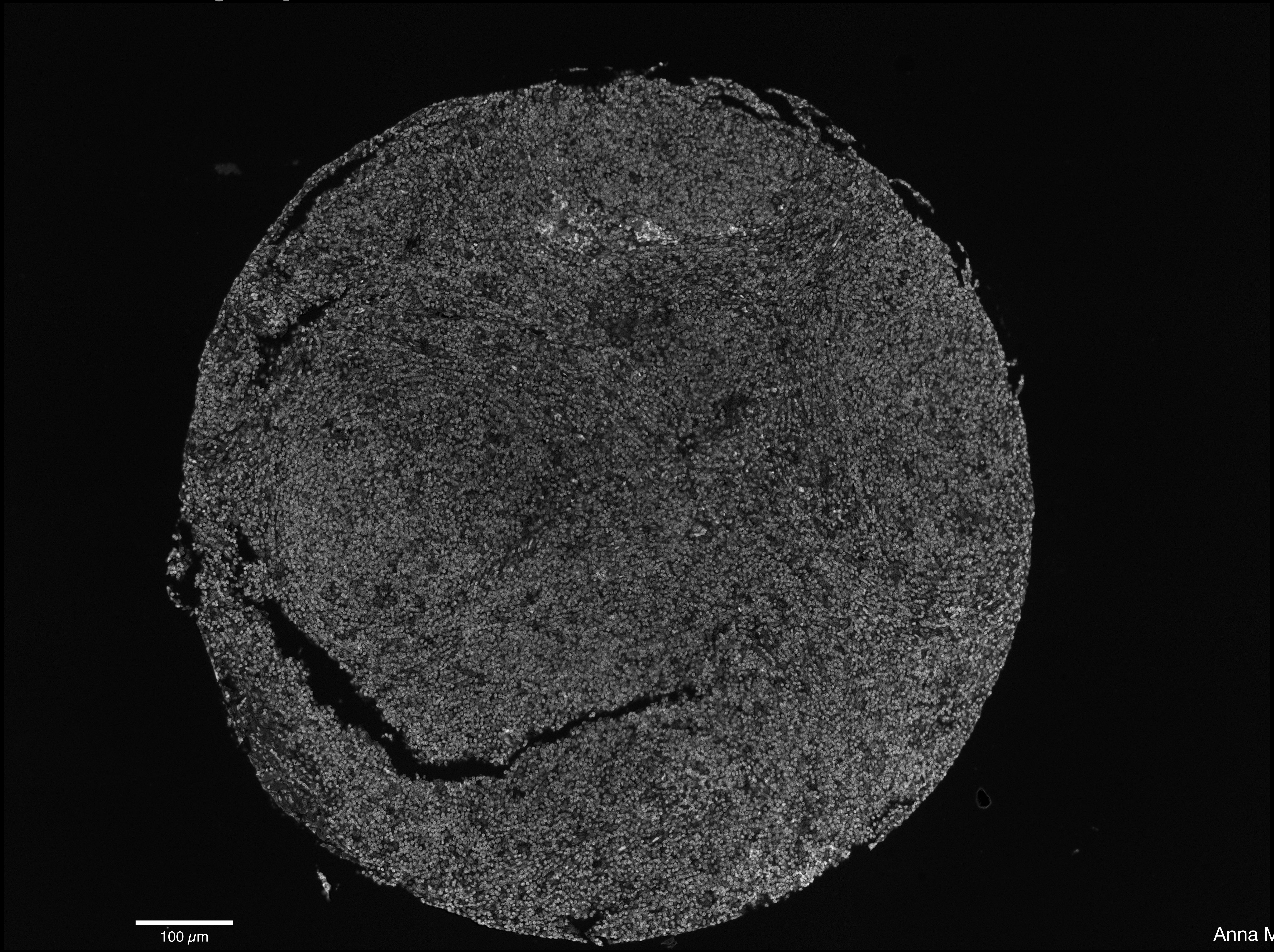


Precision $\dfrac{TP}{TP + FP}$

Recall $\dfrac{TP}{TP + FN}$

Average Precision (AP)
Accuracy $\dfrac{TP}{TP + FP + FN}$

Example: crowded lymphoma cells



Anna Maria Tsakiroglou (Manchester)

Example: crowded lymphoma cells

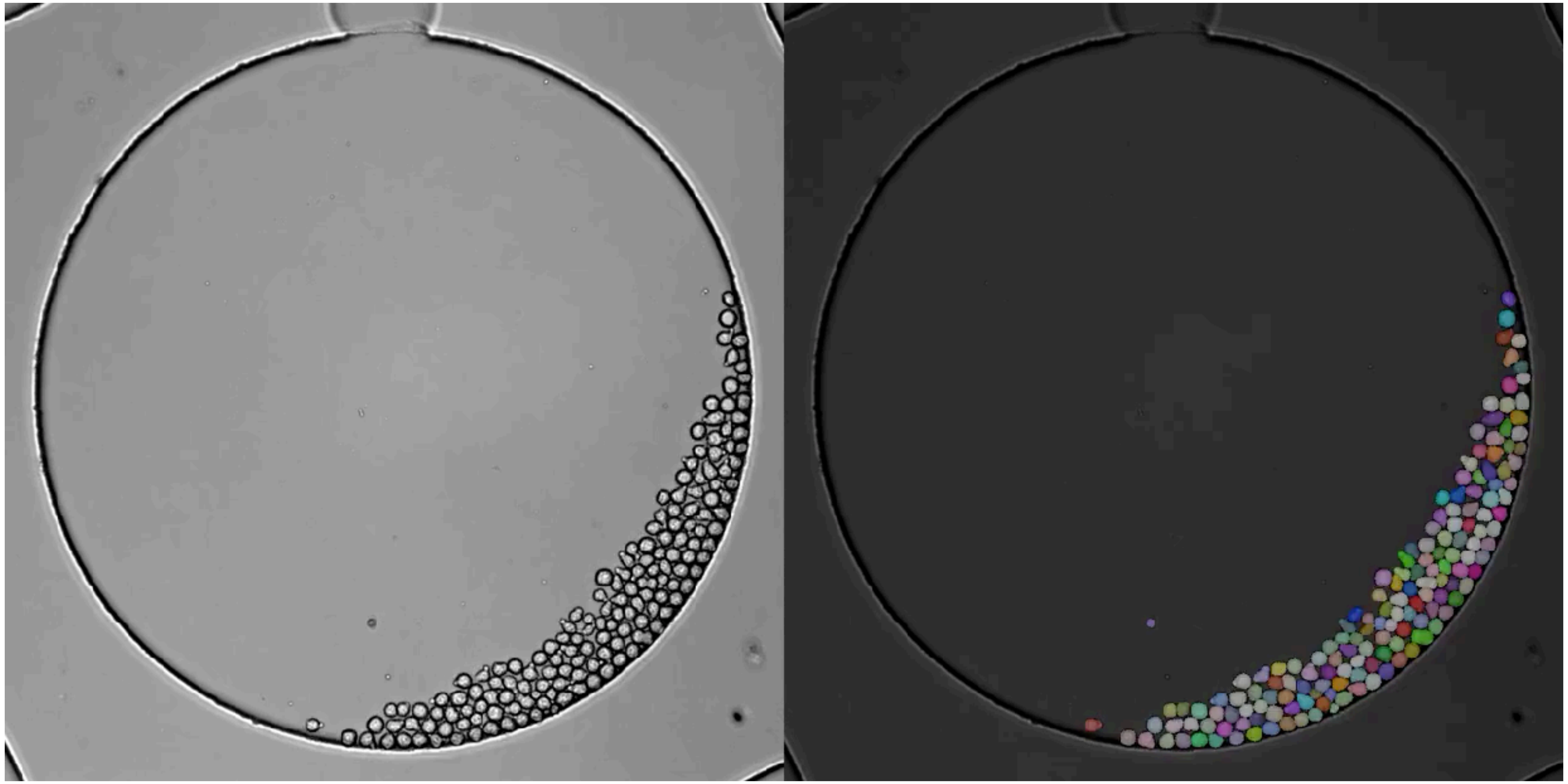
Example: Histopathology (H&E stain)

Whole Slide H&E (Sarcoma), (15mm x 6mm, 32k x 14k pixels, 1.35 GB)

Example: Histopathology (H&E stain)

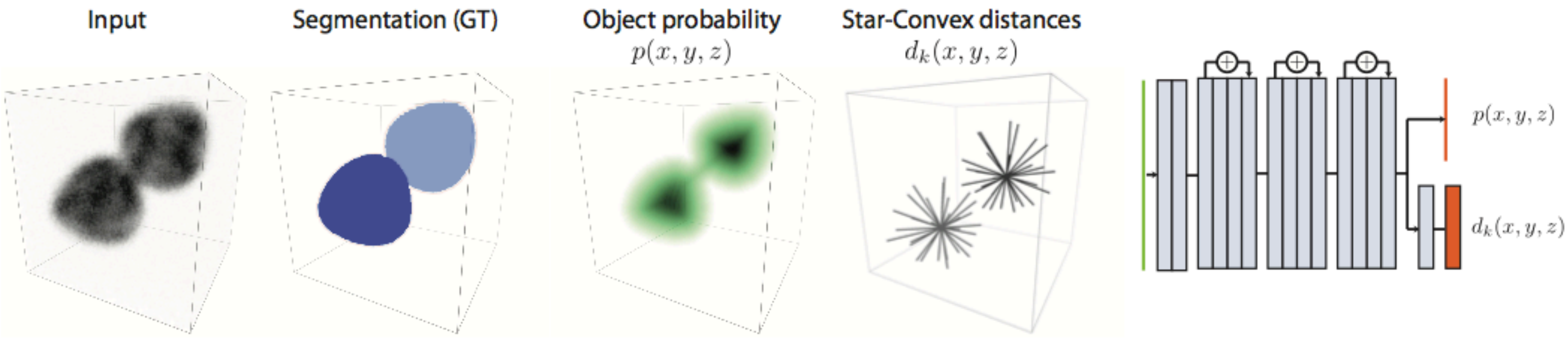

2 mm

Example: Brightfield



StarDist for 3D images

Similar approach:



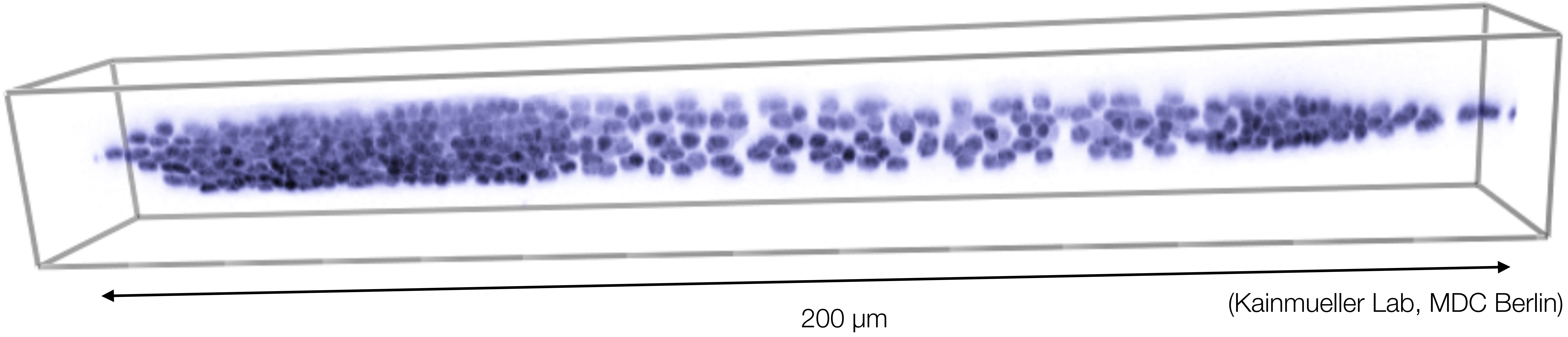
Additional Considerations:

- Ray choice: Fibonacci Lattice on sphere ~ 96 points
- Data anisotropy: Adjust rays according to GT anisotropy
- Non-Maximum Suppression of large sets of polyhedra (> 2 Mio candidates)

Examples 3D

C. elegans (L1)

- 28 Stacks
- ~15k annotated cells
- almost isotropic resolution

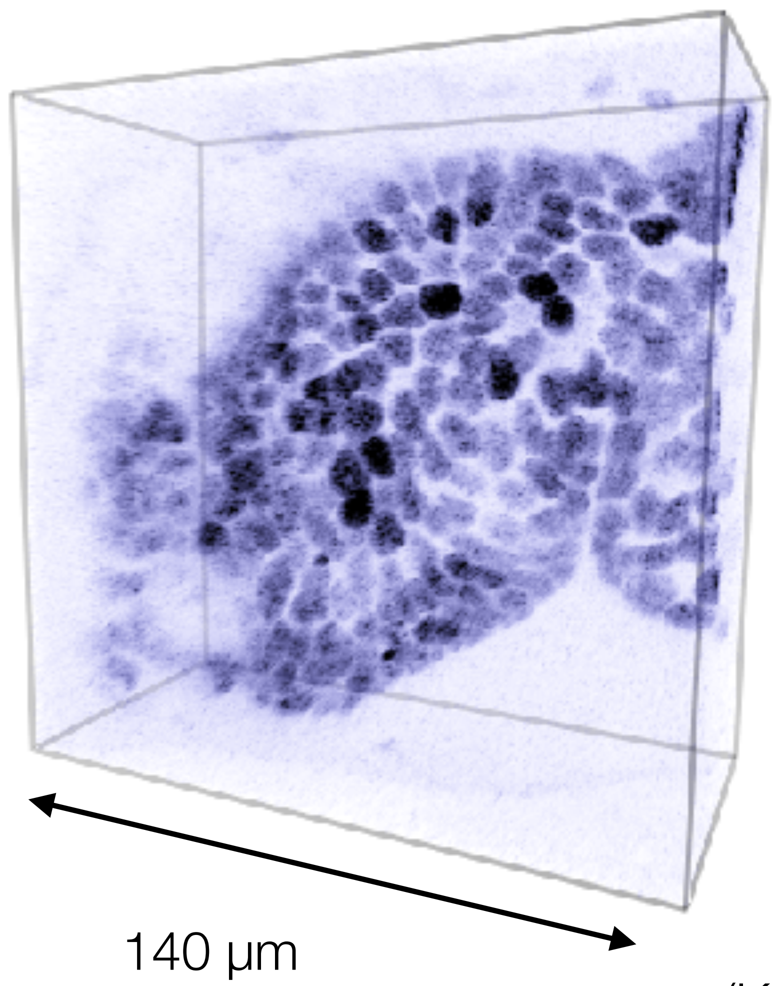


(Kainmueller Lab, MDC Berlin)

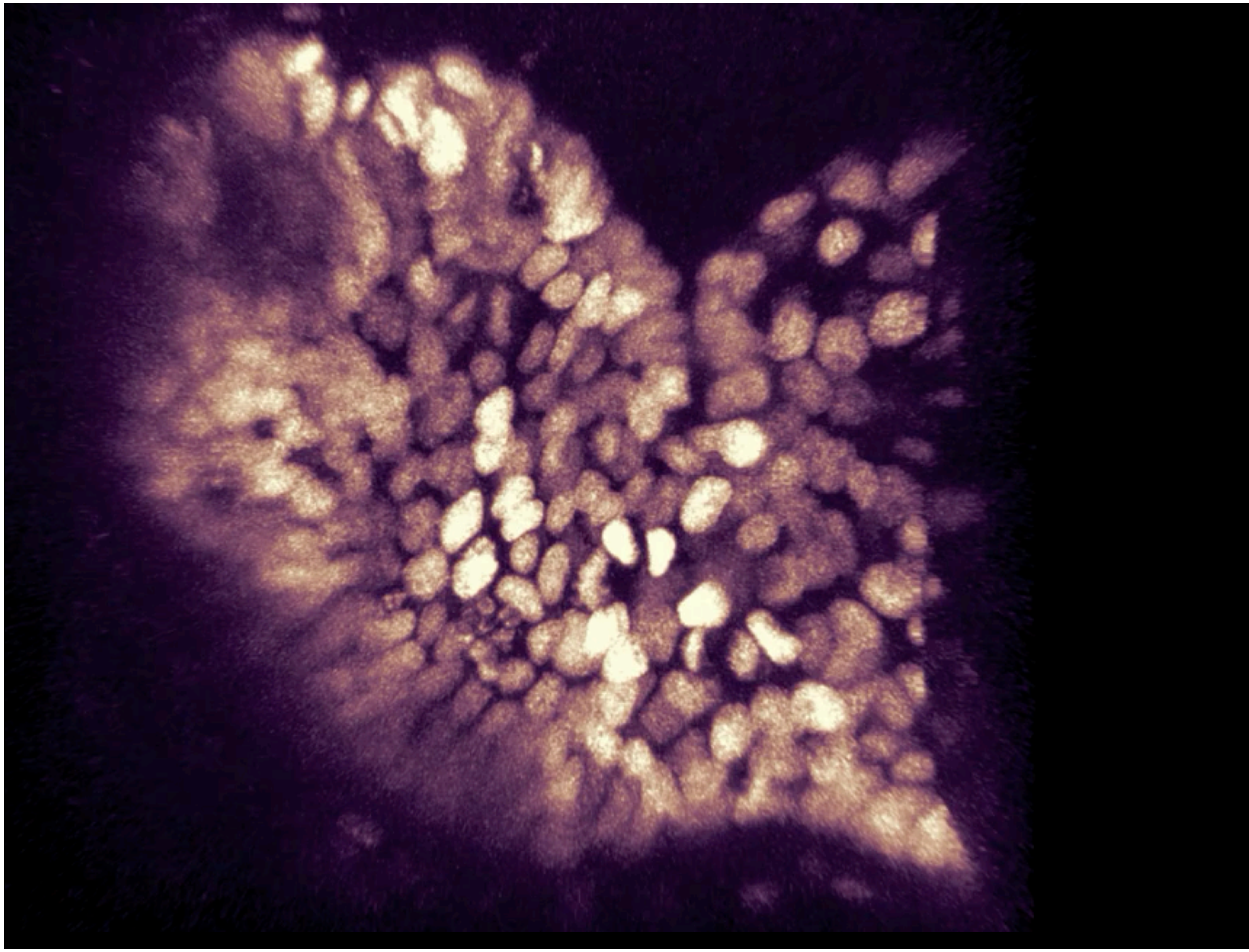
Examples 3D

Parhyale hawaiensis

- 6 stacks
- ~1500 annotated cells
- highly anisotropic resolution



(Ko Sugarawa, IGF Lyon)



Questions & Answers 1

How to use StarDist

How to use StarDist

Main python library

`pip install stardist`

https://github.com/mpicbg-csbd/stardist

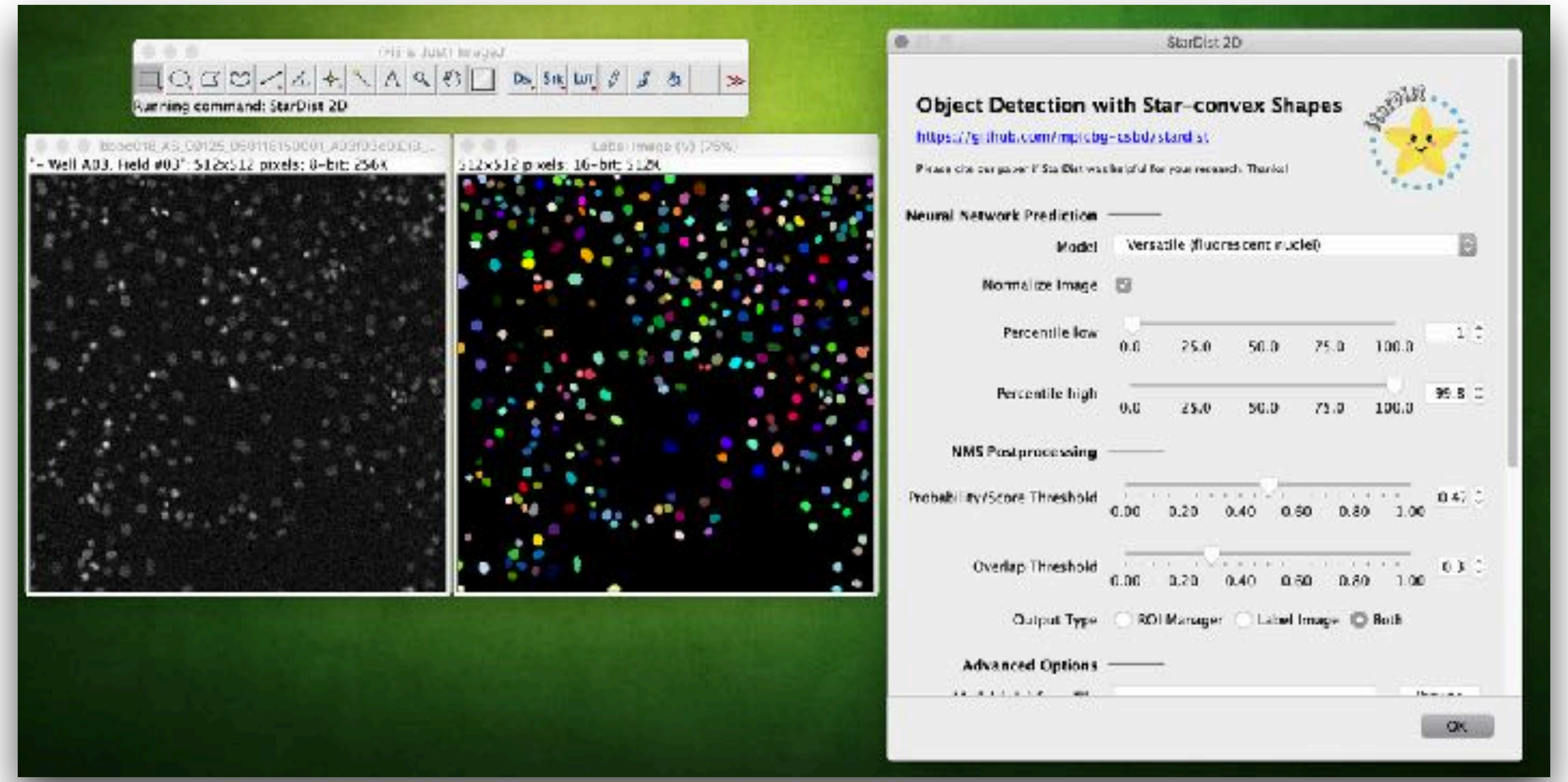
- Training and prediction for 2D/3D images
- Neural network backend keras/tensorflow via csbdeep
- Sensible training defaults
- Multi-Core NMS, tiled prediction
- Image normalization
- Segmentation/Detection measures
- Model export to Fiji

```
from stardist.models import StarDist2D, StarDist3D

model = StarDist2D(config, name = "mymodel")

Using default values: prob_thresh=0.5, nms_thresh=0.4.

model.train(X,Y,validation_data=(Xv,Yv))

Epoch 1/400
 53/100 [==============>..............] - ETA: 26s - lc
t_loss: 8.6035 - prob_kld: 0.3578 - dist_relevant_mae: 8

labels, _ = model.predict_instances(img)
```

Fiji Plugin

https://imagej.net/StarDist

- Prediction for 2D images by already trained models
- Scriptable
- CPU and GPU support (via CSBDeep-Fiji by Deborah Schmidt, MPI-CBG)



How to use StarDist on your own data

- 2D data
- similar to pretrained images (H&E, fluorescent nuclei)

→

Try pretrained models
python
CPU
(or GPU)

→

Predict on your data
python
CPU
(or GPU)

- 2D data dissimilar to pretrained
- 3D data

→

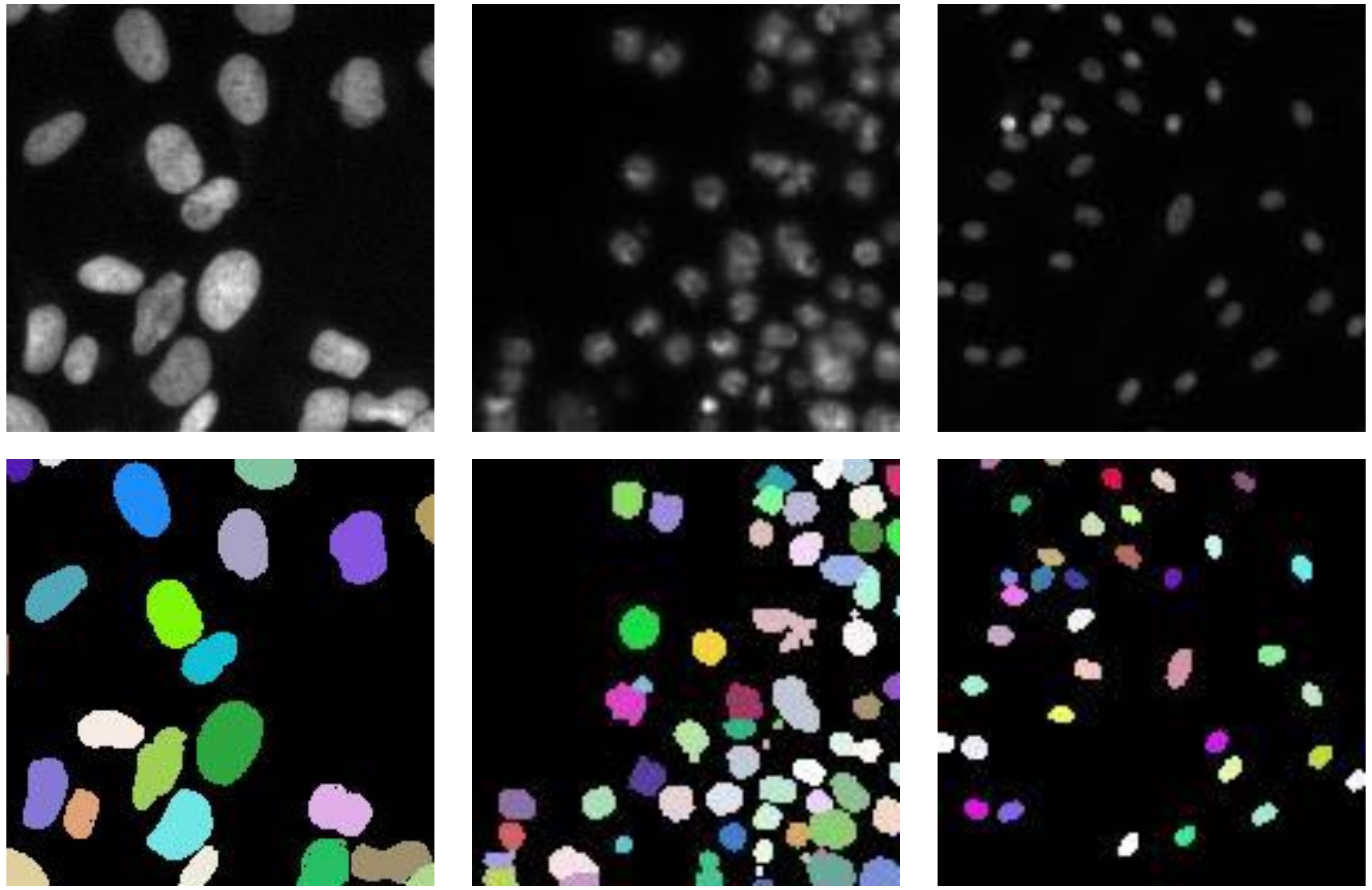
Generate Annotations & train own model
python
GPU

→

Predict on your data
python
CPU
(or GPU)

Pretrained Models (2D)

Fluorescence Microscopy
Single Channel

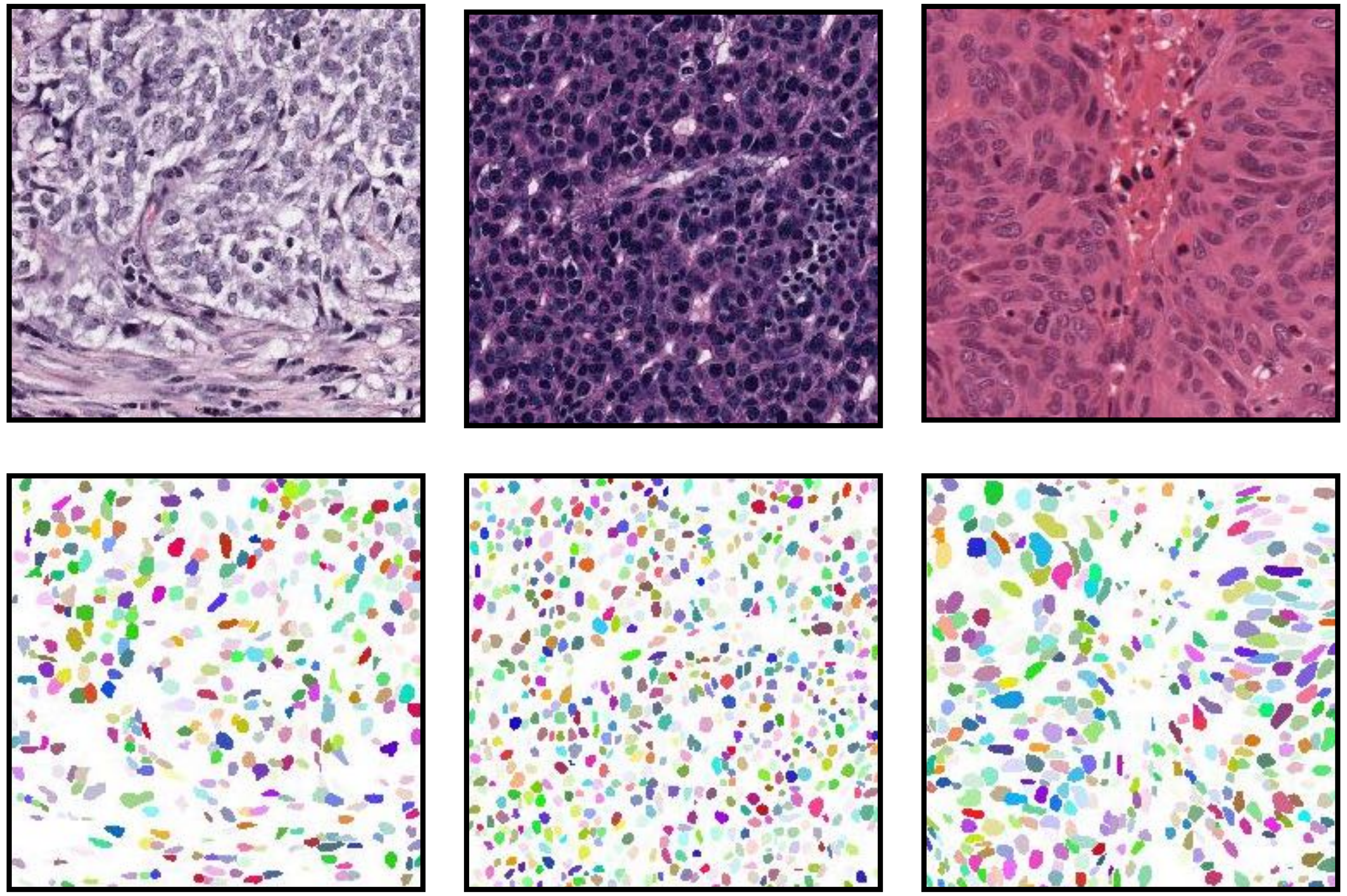


Data Science Bowl 2018

Caicedo et al. (2018)

~ 600 images (2D)
~ 20k annotations

Histopathology
RGB H&E



MoNuSeg

Kumar et al (2017)

~ 30 Images (2D)
~ 22k annotations

Pretrained Models (2D)

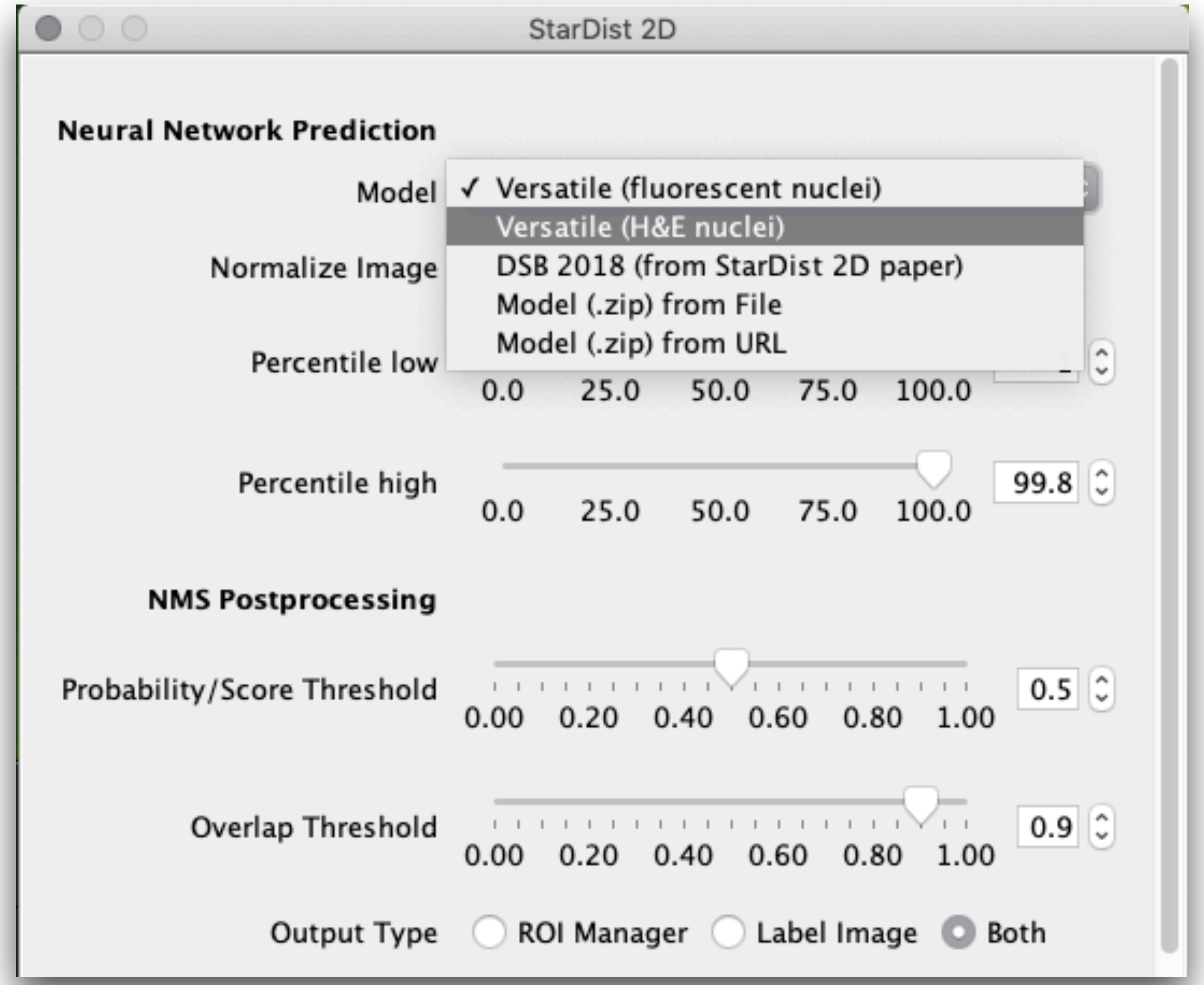
python

```
from stardist.models import StarDist2D

StarDist2D.from_pretrained()
```

```
There are 4 registered models for 'StarDist2D':

Name                  Alias(es)
────                  ─────────
'2D_versatile_fluo'   'Versatile (fluorescent nuclei)'
'2D_versatile_he'     'Versatile (H&E nuclei)'
'2D_paper_dsb2018'    'DSB 2018 (from StarDist 2D paper)'
'2D_demo'             None
```

```
model = StarDist2D.from_pretrained('2D_versatile_fluo')

labels, _ = model.predict_instances(img)
```

```
Found model '2D_versatile_fluo' for 'StarDist2D'.
Loading network weights from 'weights_best.h5'.
Loading thresholds from 'thresholds.json'.
Using default values: prob_thresh=0.479071, nms_thresh=0.3.
```



- Try different scalings
- Invert intensity, RGB -> grayscale, ….
- Play around with the prob and overlap (NMS) threshold

Demo: Pretrained models in Python and Fiji (2D)

Try pretrained models

python

CPU
(or GPU)

→

Predict on your data

python

CPU
(or GPU)

Training of custom models

If the pretrained models do not work on your 2D images (or your data is 3D) you need to training your own model

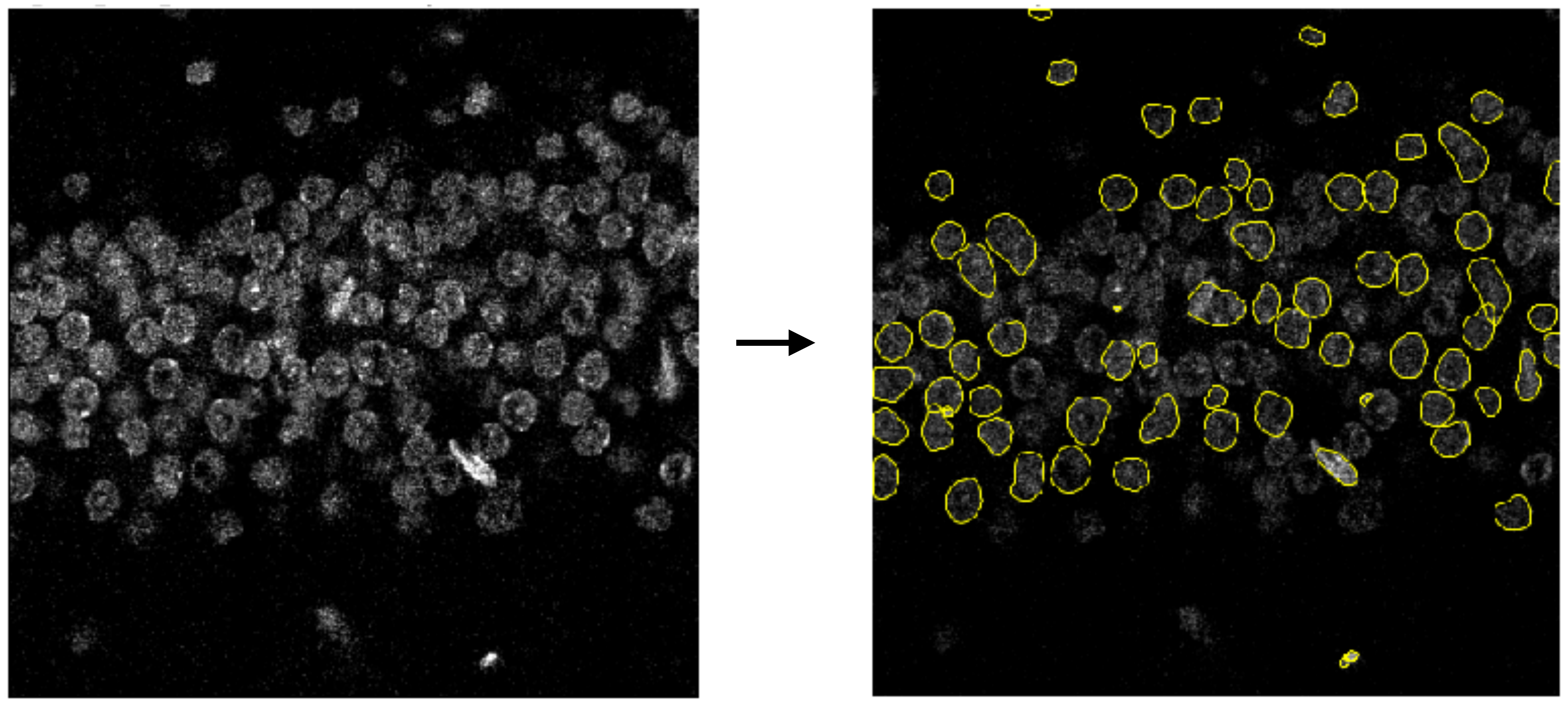
Generate Annotations & train own model

python **GPU**

→

Predict on your data

python **CPU (or GPU)**



Data from Silvia Monari, EPFL

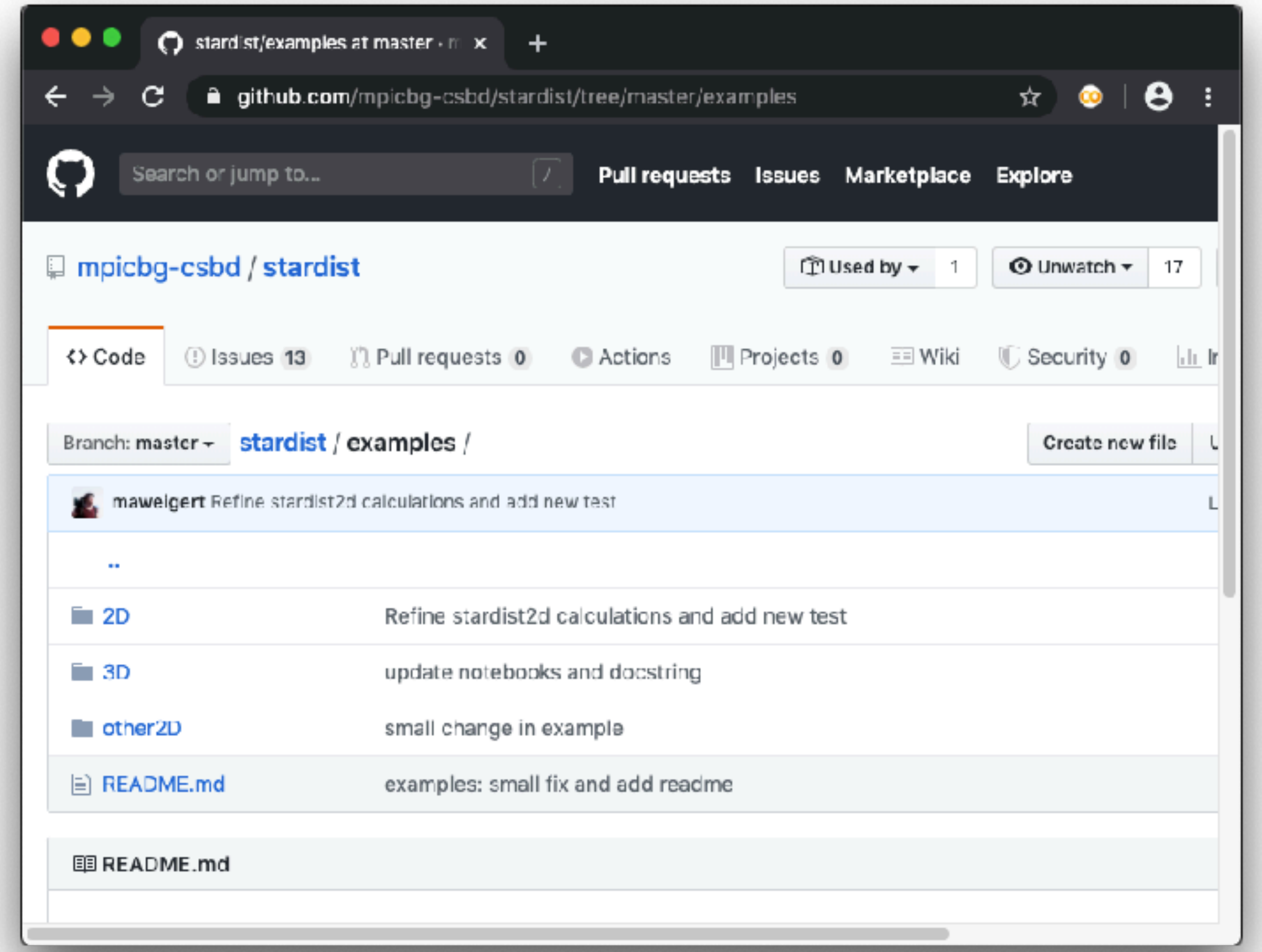
- Needs user annotated Image/label pairs
- GPU workstation
- Python
- Training time from scratch:
  - 2D: 30min-2h
  - 3D: 4h-12h

Training of custom models

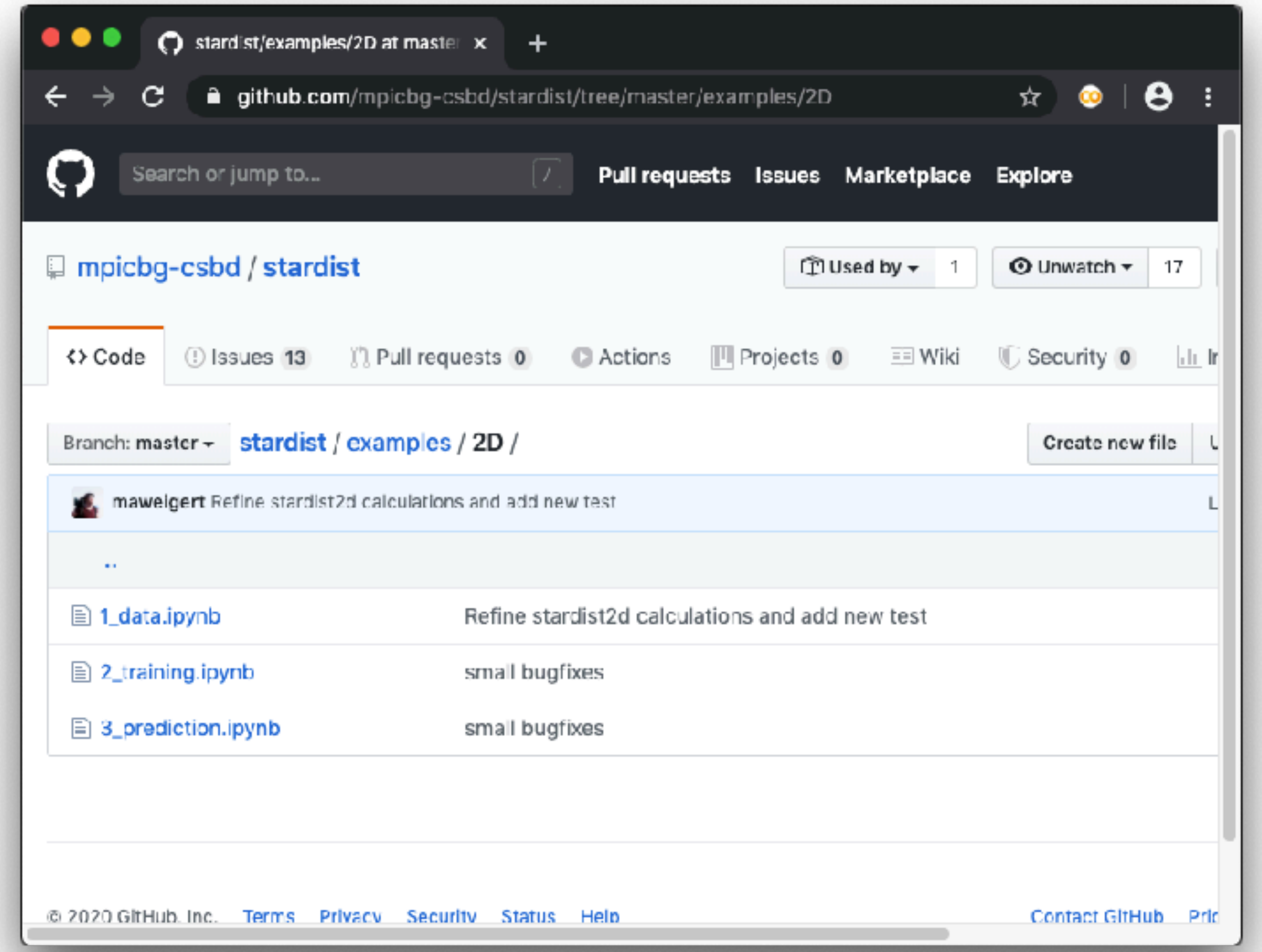
Demo Jupyter Notebooks for 2D and 3D data that you can adapt and run on your (or your facilities) GPU-workstation

https://github.com/mpicbg-csbd/stardist/tree/master/examples

Jupyter Notebooks examples in 2D and 3D



1. Data preparation/inspection
2. Training of the StarDist model
3. Prediction on new images



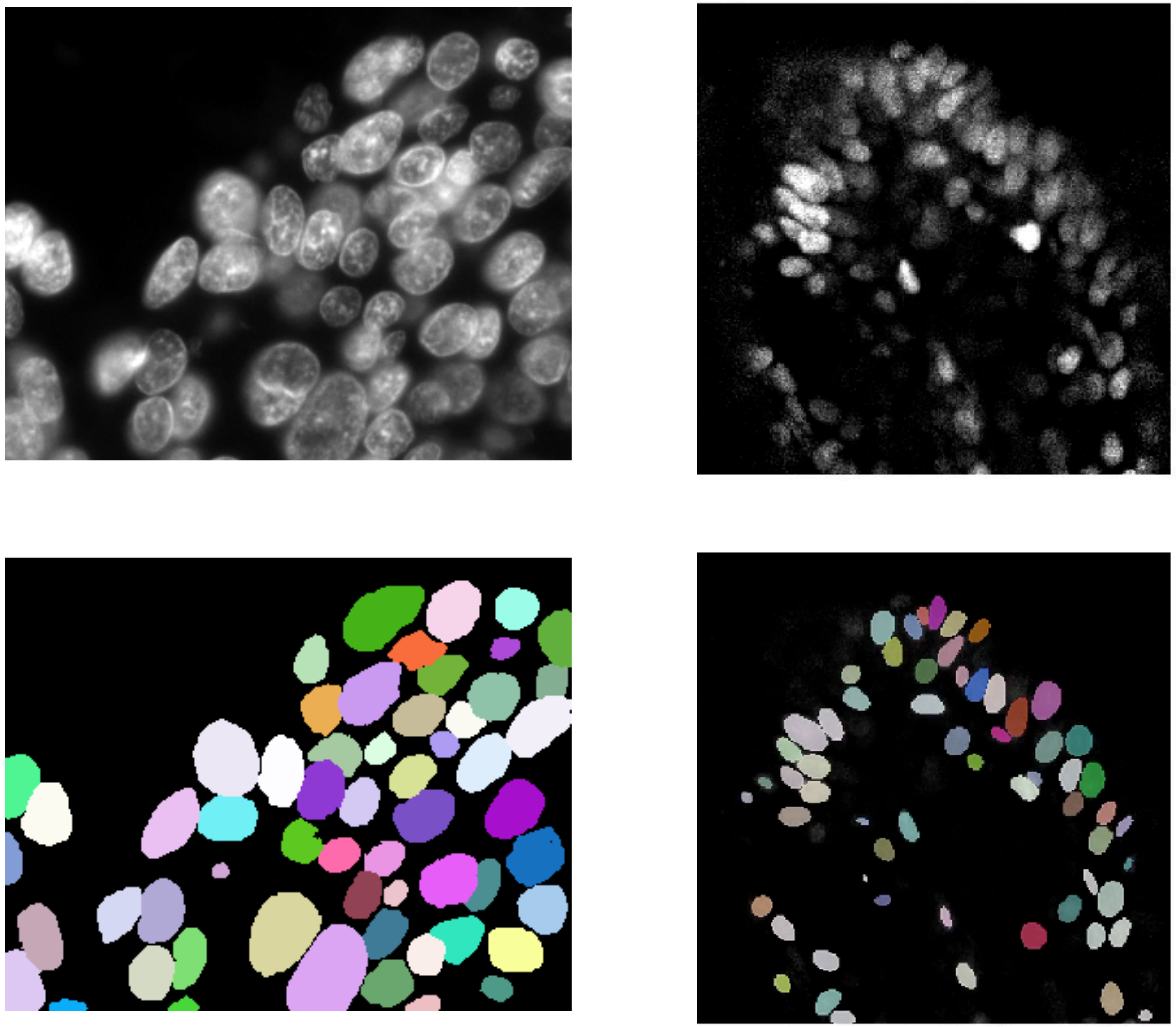
Training data generation

Generate corresponding Image/Mask pairs:

2D **3D**

Images

Masks



- Image/Mask: tif format
- Mask: Every cell pixel has to have a different label (dense labeling)
- Use crops from different regions/timepoints
- Size: at least 128^2 (2D) or 96^3 (3D)
- Number: N > 10 (2D) and N > 4 (3D)

Training data generation

```
data/
├── train
|   ├── images
|   |   ├── img_1.tif
|   |   ├── img_2.tif
|   |   ├── img_3.tif
|   |   ├── img_4.tif
|   |   ├── img_5.tif
|   |   └── img_6.tif
|   └── masks
|       ├── mask_1.tif
|       ├── mask_2.tif
|       ├── mask_3.tif
|       ├── mask_4.tif
|       ├── mask_5.tif
|       └── mask_6.tif
└── test
    ├── images
    |   ├── img_6.tif
    |   ├── img_7.tif
    |   └── img_8.tif
    └── masks
        ├── mask_6.tif
        ├── mask_7.tif
        └── mask_8.tif
```

Generate corresponding Image/Mask pairs:

- Image/Mask: tif format
- Mask: Every cell pixel has to have a different label (dense labeling)
- Use crops from different regions/timepoints, e.g.
  2D: > 10 of size 128x128
  3D: > 4 of size 32x96x96
- Split into (non-overlapping) train and test for validation

Annotation Software (2D)

Fiji/ImageJ

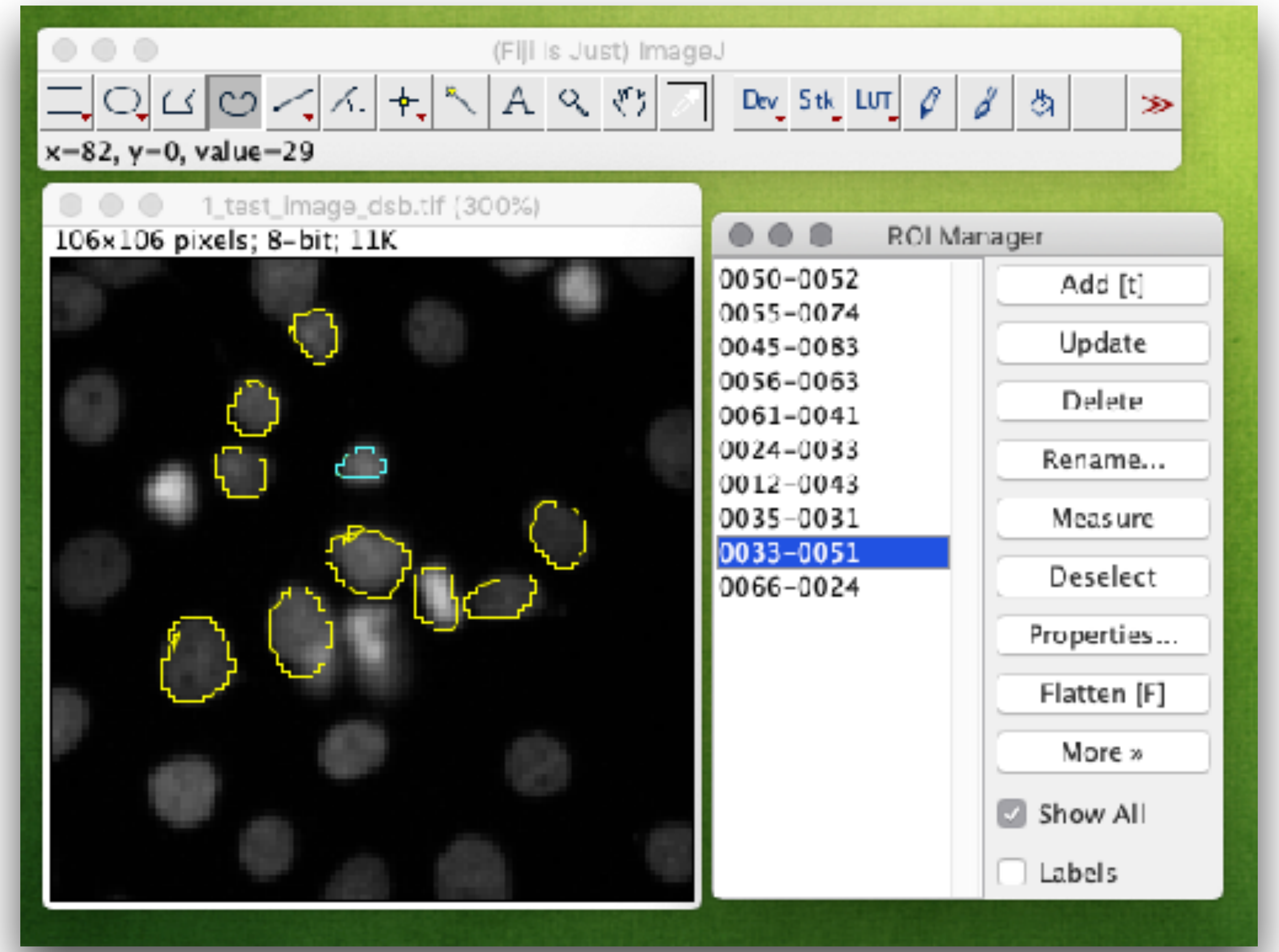
- Draw Roi per object (Roi-Manager)
- Convert to label mask (e.g. see here)

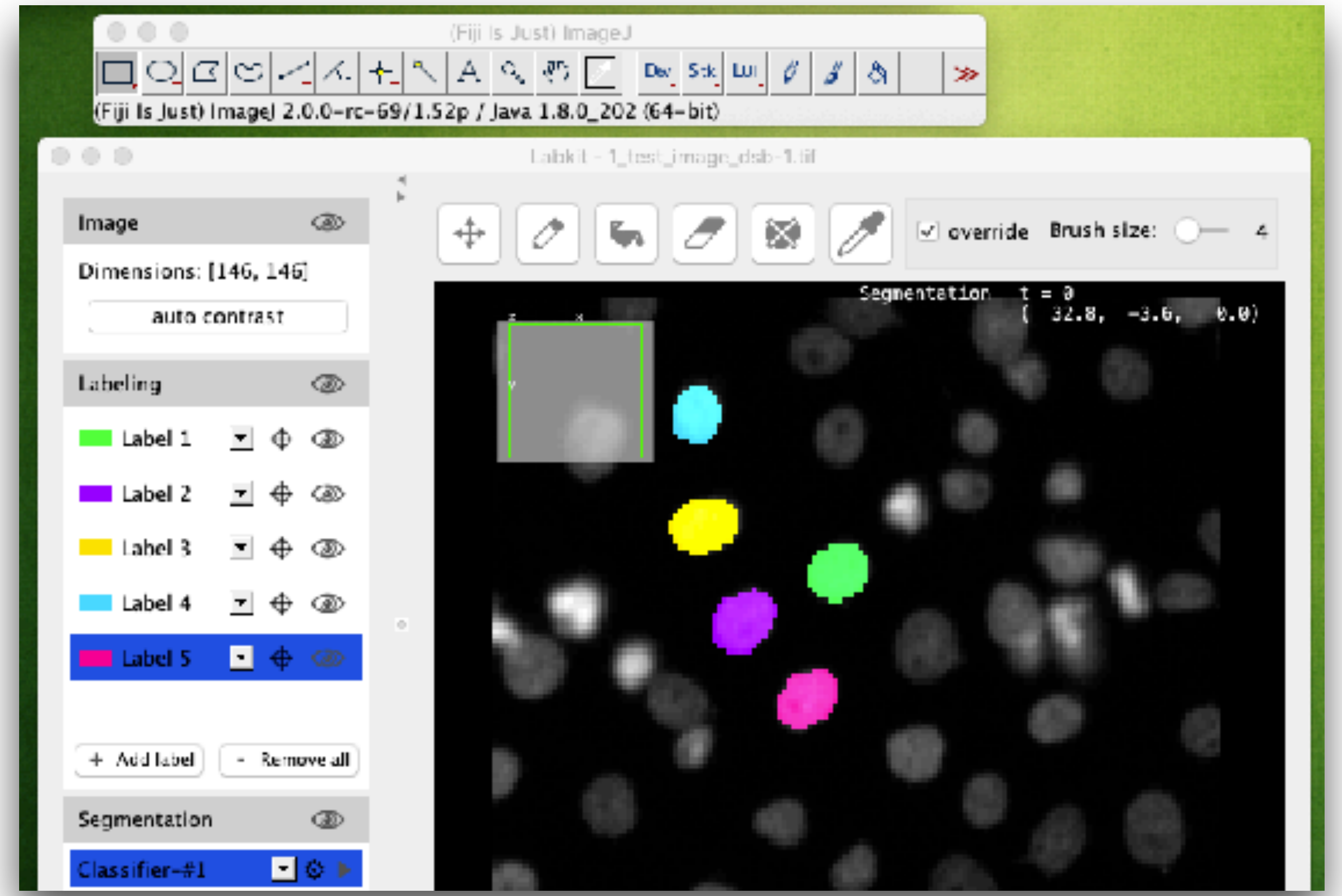
Fiji + LabKit

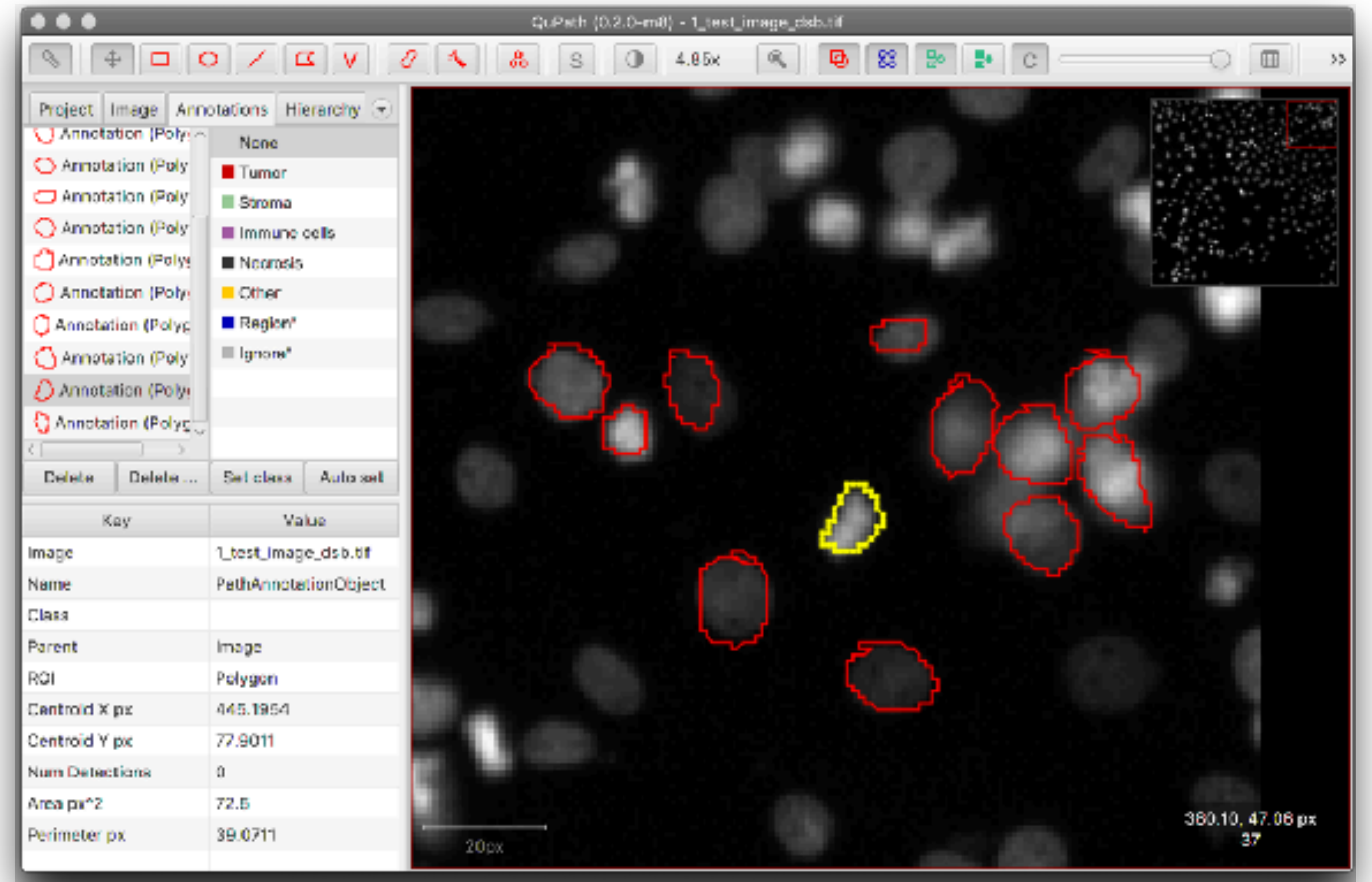
- Directly draw label mask

QuPath

- Draw ROI per object
- Convert to label mask







http://fiji.sc/ Schindelin et al. (2012)
Schneider, Rasband et al (2012)

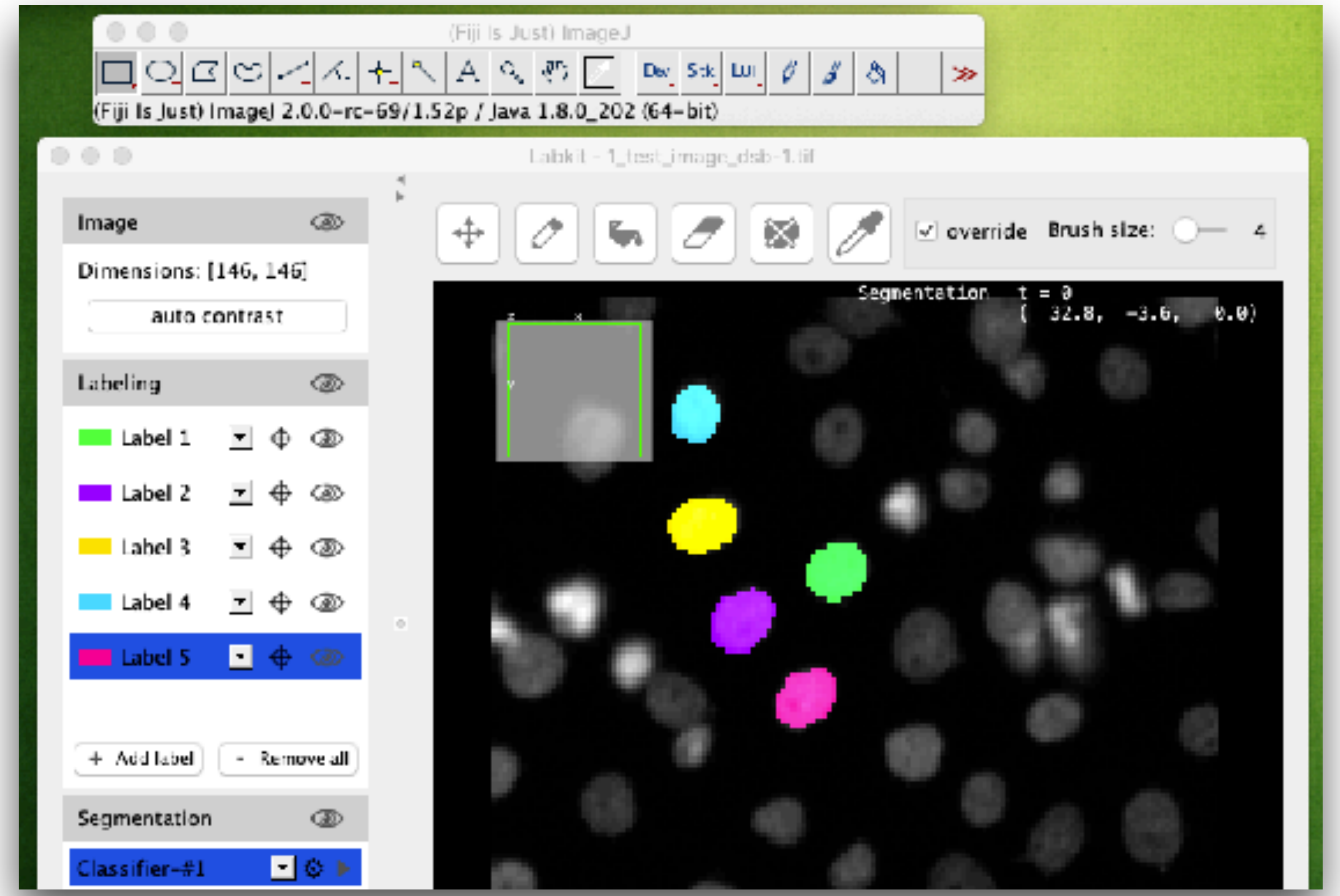
https://imagej.net/Labkit M. Arzt, MPI-CBG

https://qupath.github.io Bankhead et al. (2017)

Annotation Software (3D)

Fiji + LabKit

- Directly draw label mask
- Reorder z -> t (to not miss a plane)

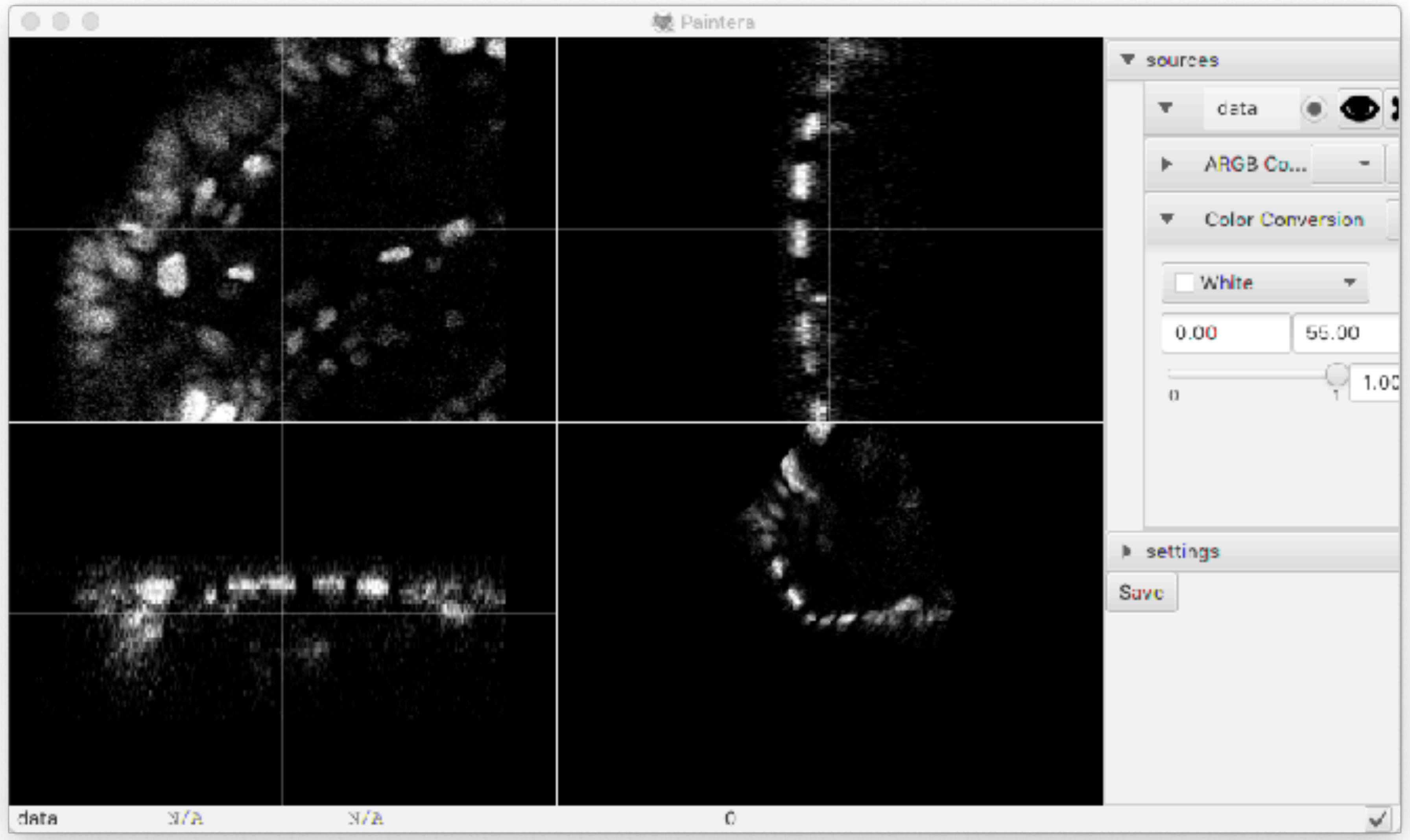


https://imagej.net/Labkit

M. Arzt, MPI-CBG

Paintera

- For very large volumes (> GBs)
- Powerful but slightly steeper learning curve



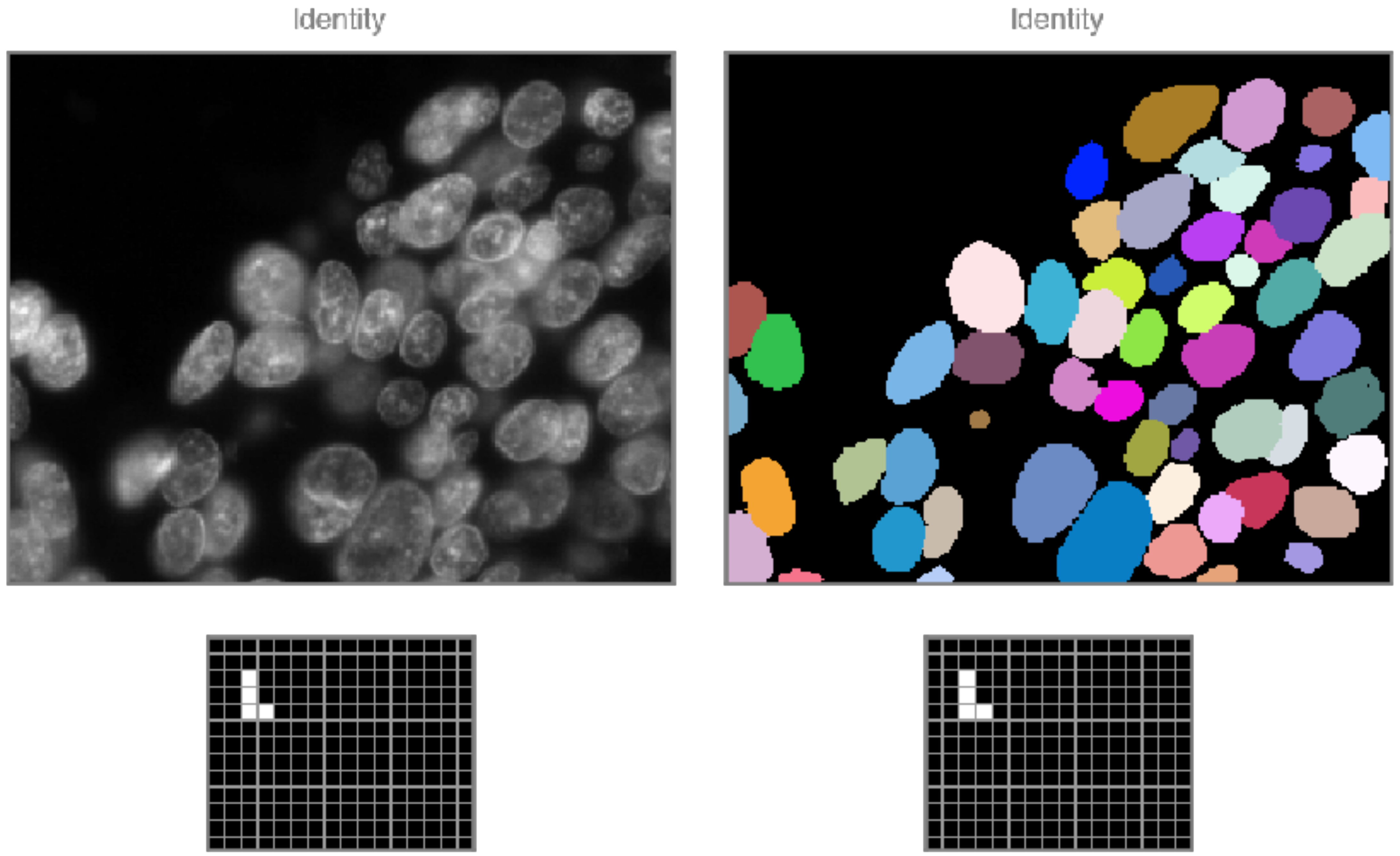
https://github.com/saalfeldlab/paintera

P. Hanslovsky, S. Saalfeld et al Janelia

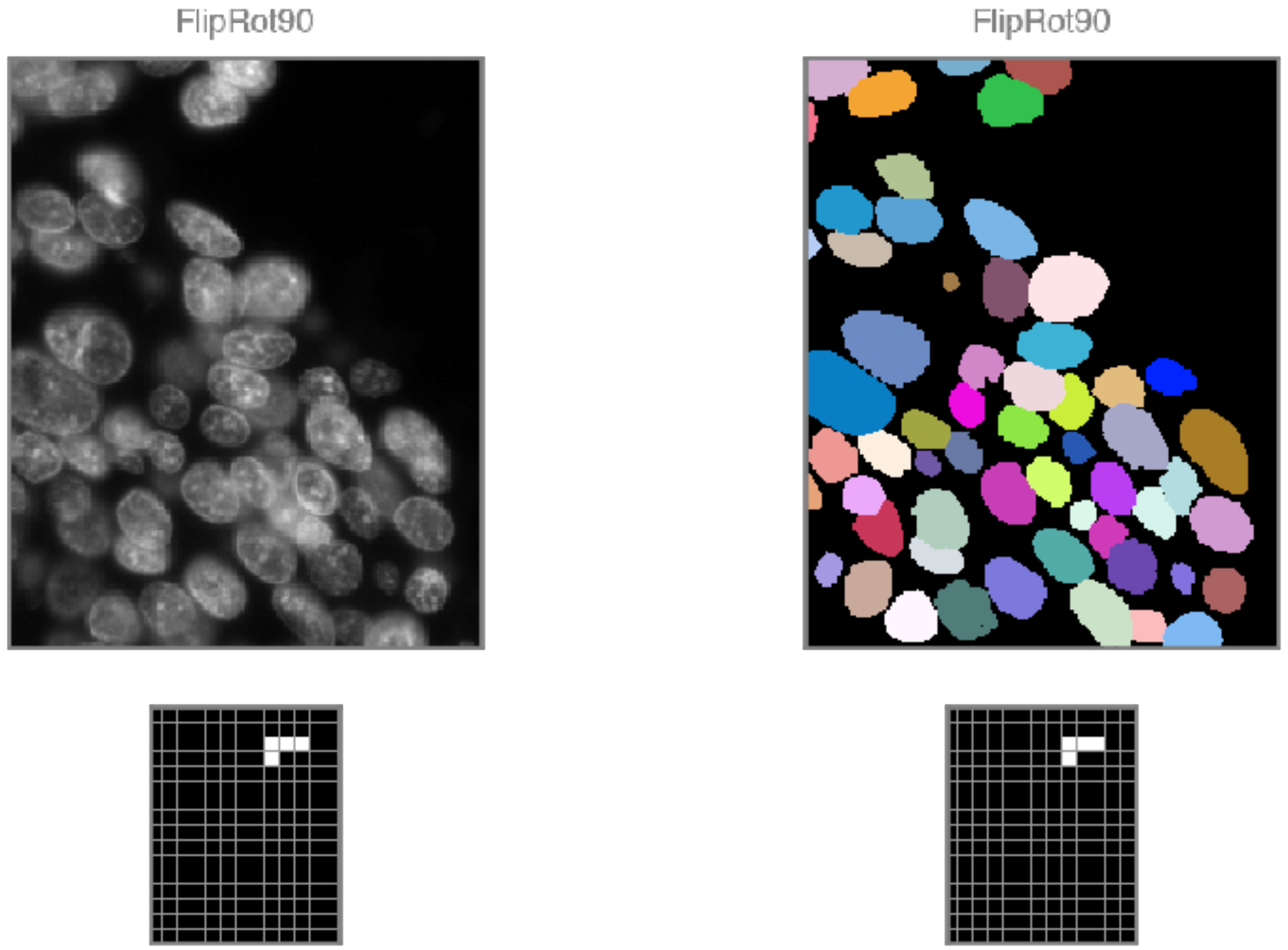
Data Augmentation

Artificially create more training data by transforming existing images/masks into different, yet plausible versions

Original

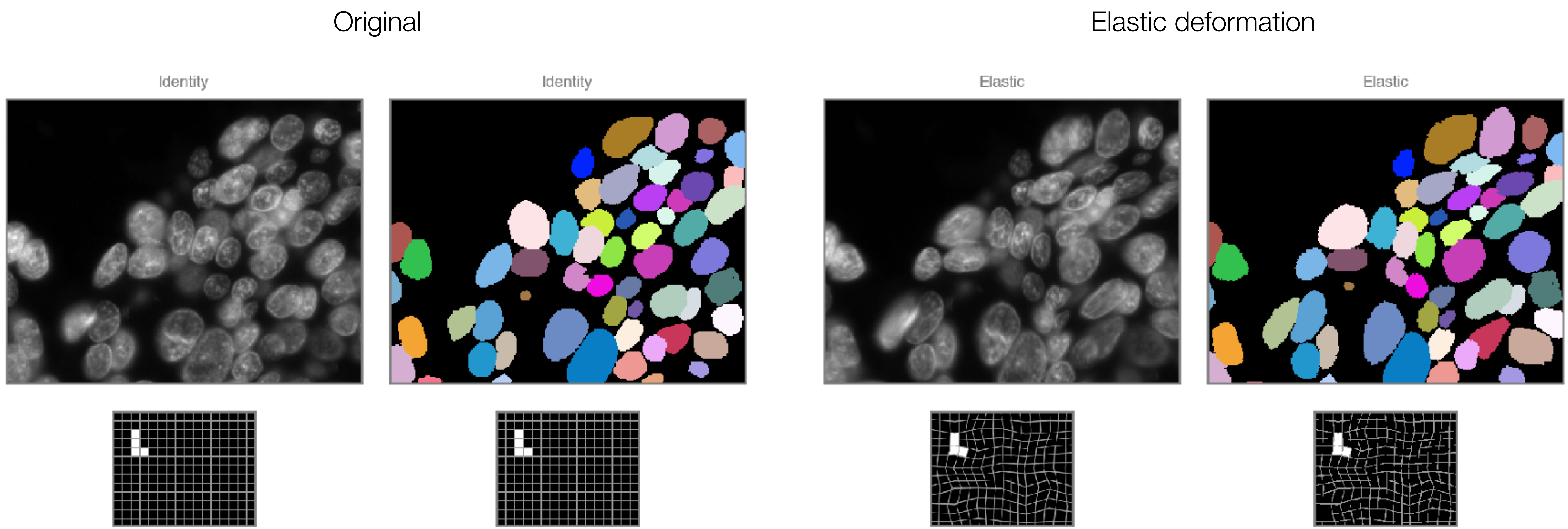


Flip/90 degree Rotation



Data Augmentation

Artificially create more training data by transforming existing images/masks into different, yet plausible versions

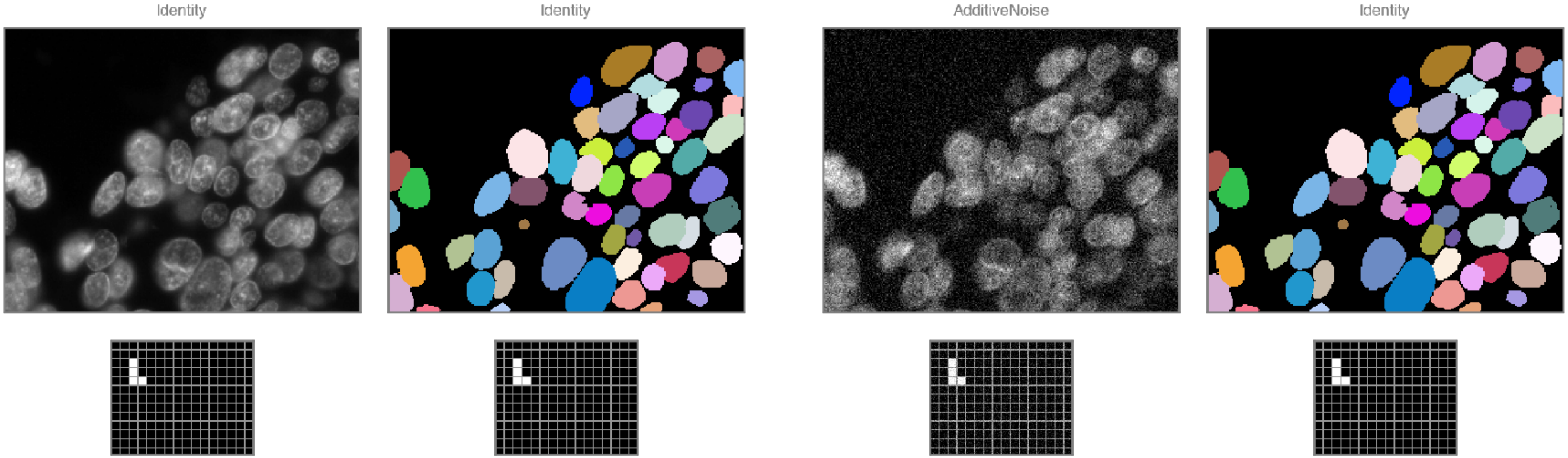


Data Augmentation

Artificially create more training data by transforming existing images/masks into different, yet plausible versions

Original

Noise/Intensity shift



Demo: Training of custom models (python)

Generate Annotations & train own model

python

GPU

→

Predict on your data

python

CPU
(or GPU)

Webinar Demo on Google Colab
(Click to try it out!)

Where to ask questions / get help?

Github project issues page

- Technical questions
- Bugs, unexpected behavior
- Missing functionality

https://github.com/mpicbg-csbd/stardist/issues

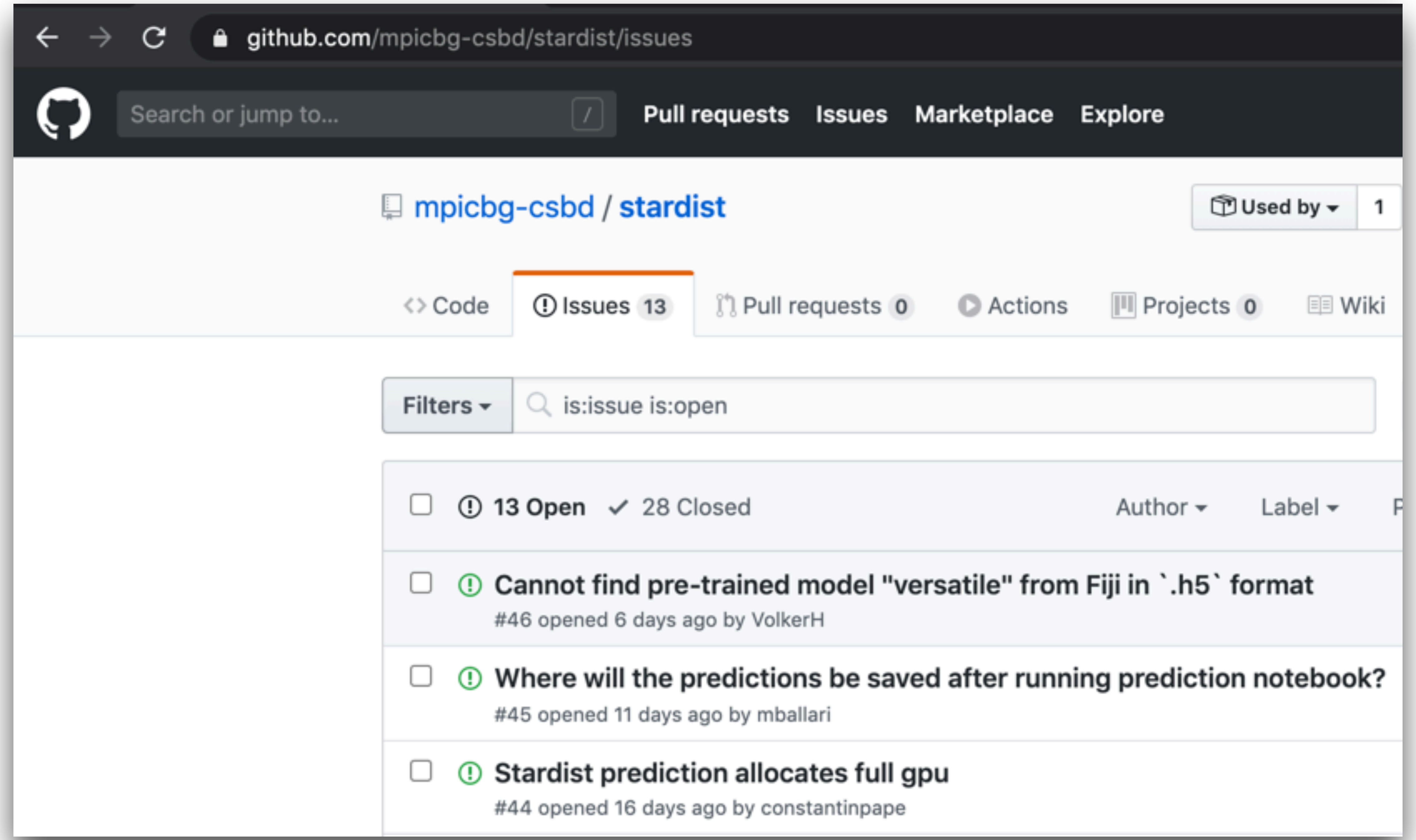
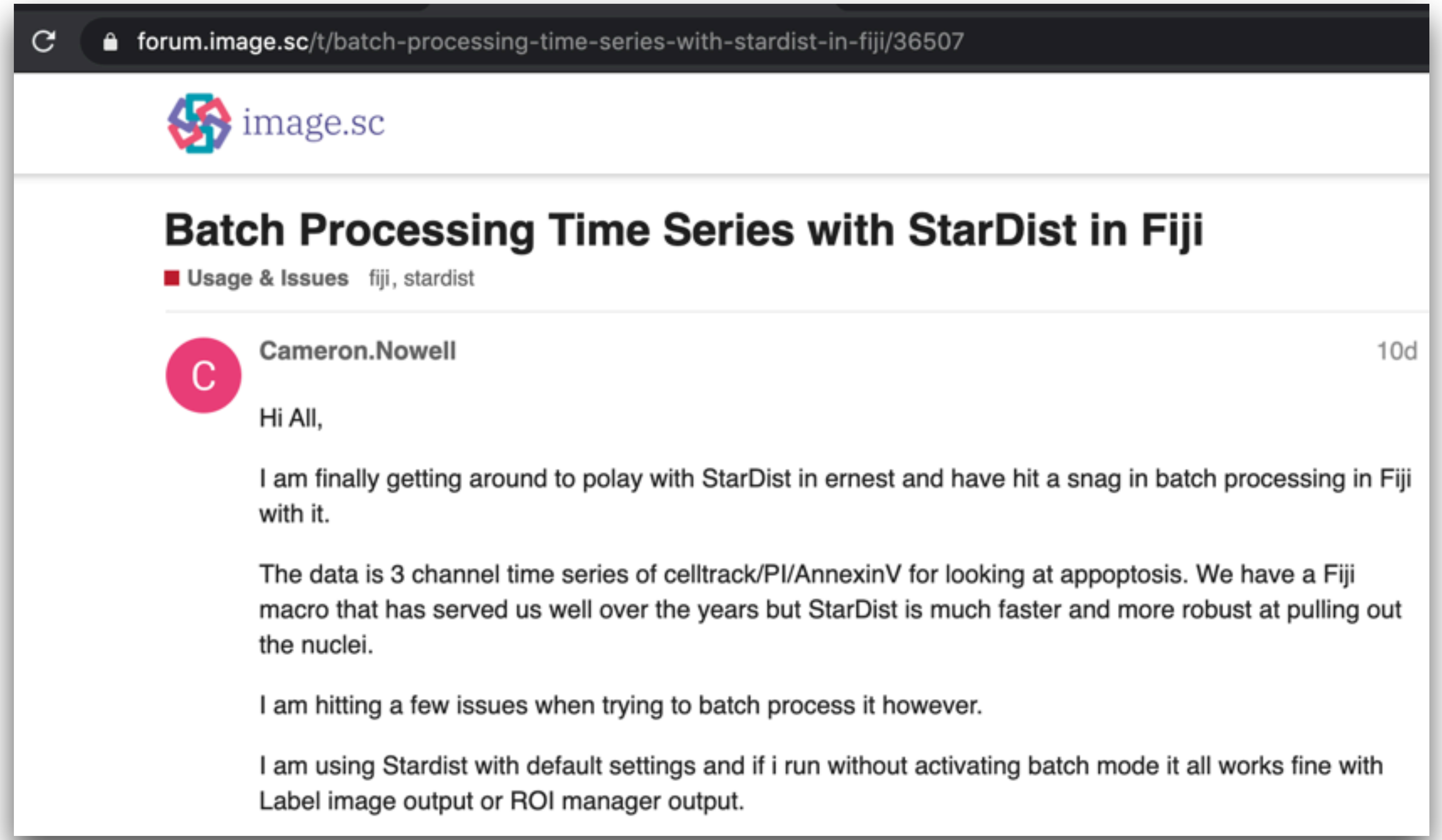


Image.sc forum

- Usage questions
- Best practices
- Problems with training/data

https://forum.image.sc **(use tag “stardist”!)**



StarDist in a core facility

Questions & Answers 2