Section 2
Introduction to Machine Learning & Bioimage Analysis in Python

Most slides and materials were adapted from:

Lecture 8 of Bioimage Analysis 2020 by Robert Haase

EMBL Image Analysis with Python Course by Jonas Hartmann

Agenda

- Introduction to ML for pixel and object classification
  - DEMO: ilastik, cellpose, napari
- A **very, very brief** chat about biostatistics
- Codelab

Machine Learning for pixel and object detection and classification

Image Segmentation using thresholding

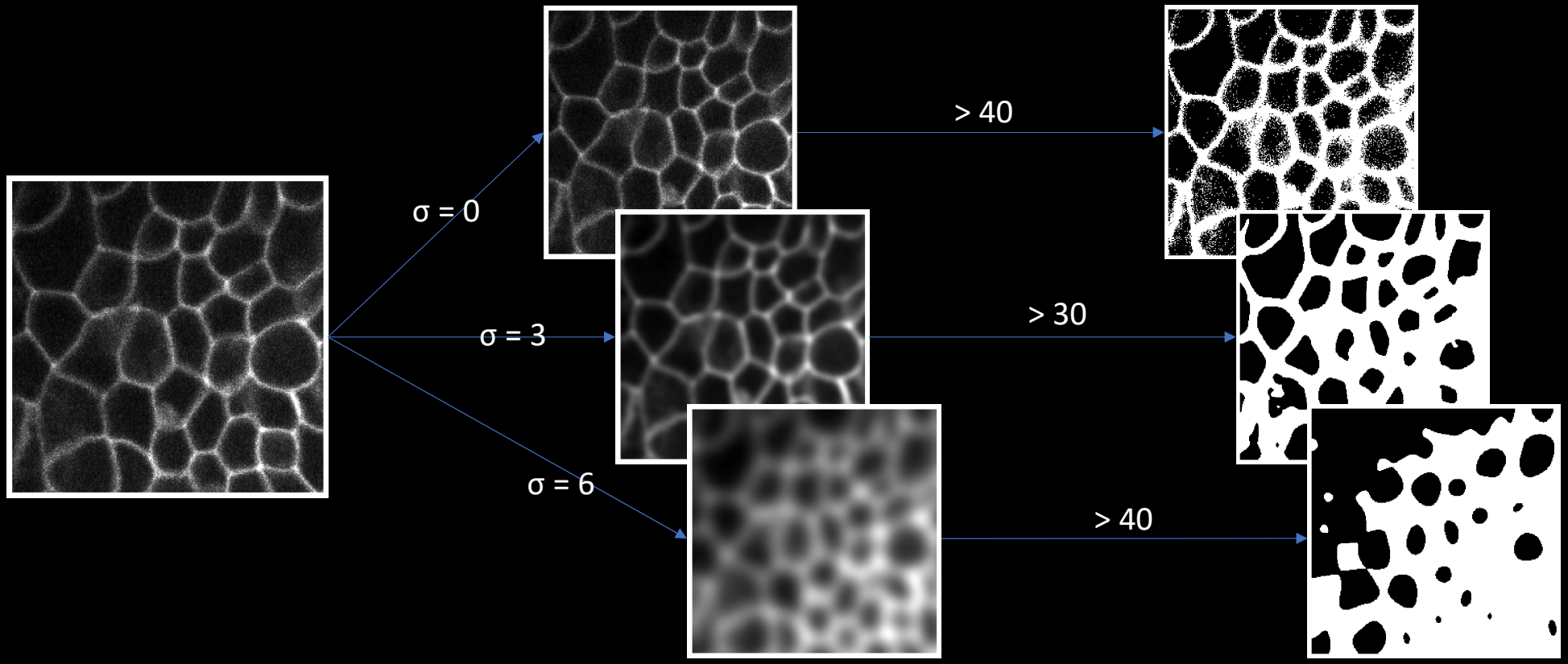


Image Segmentation using thresholding

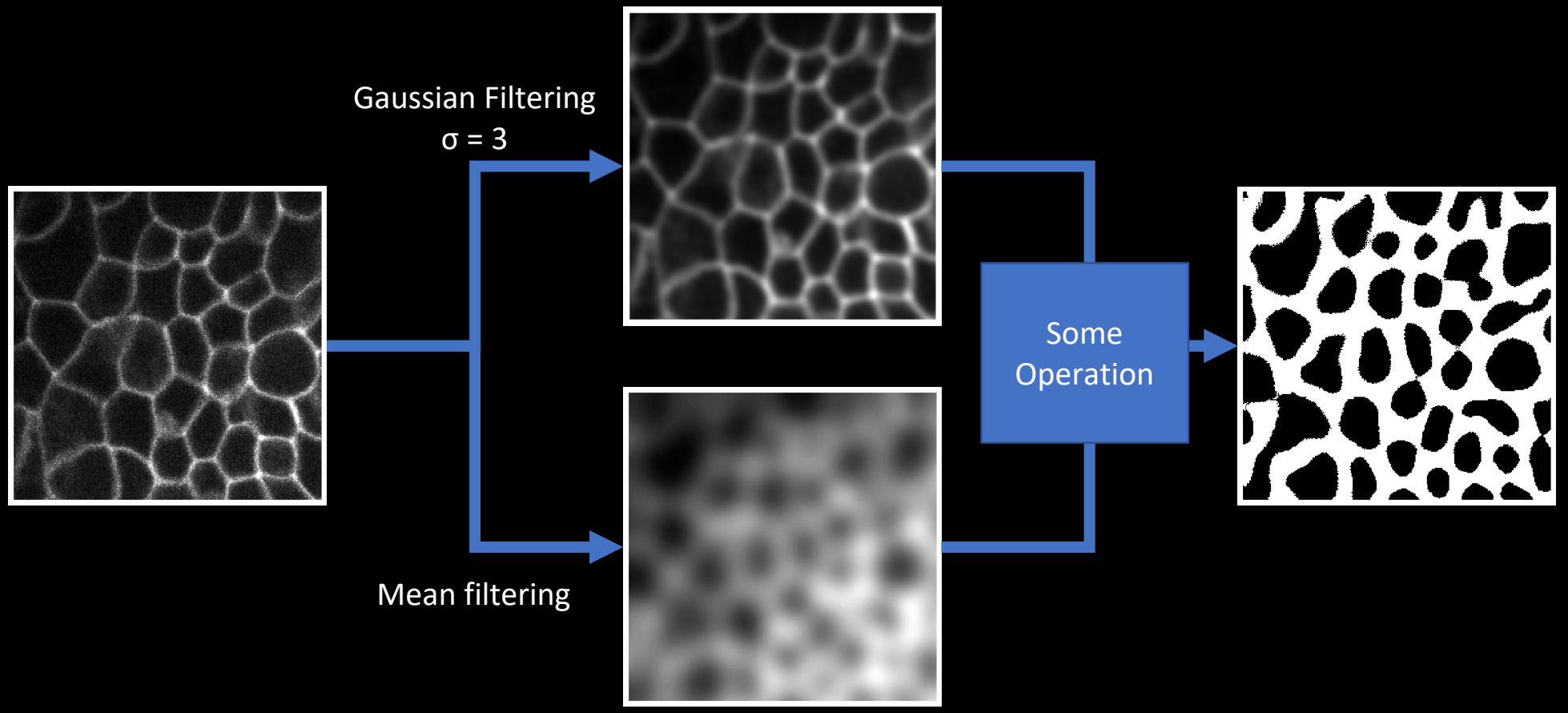
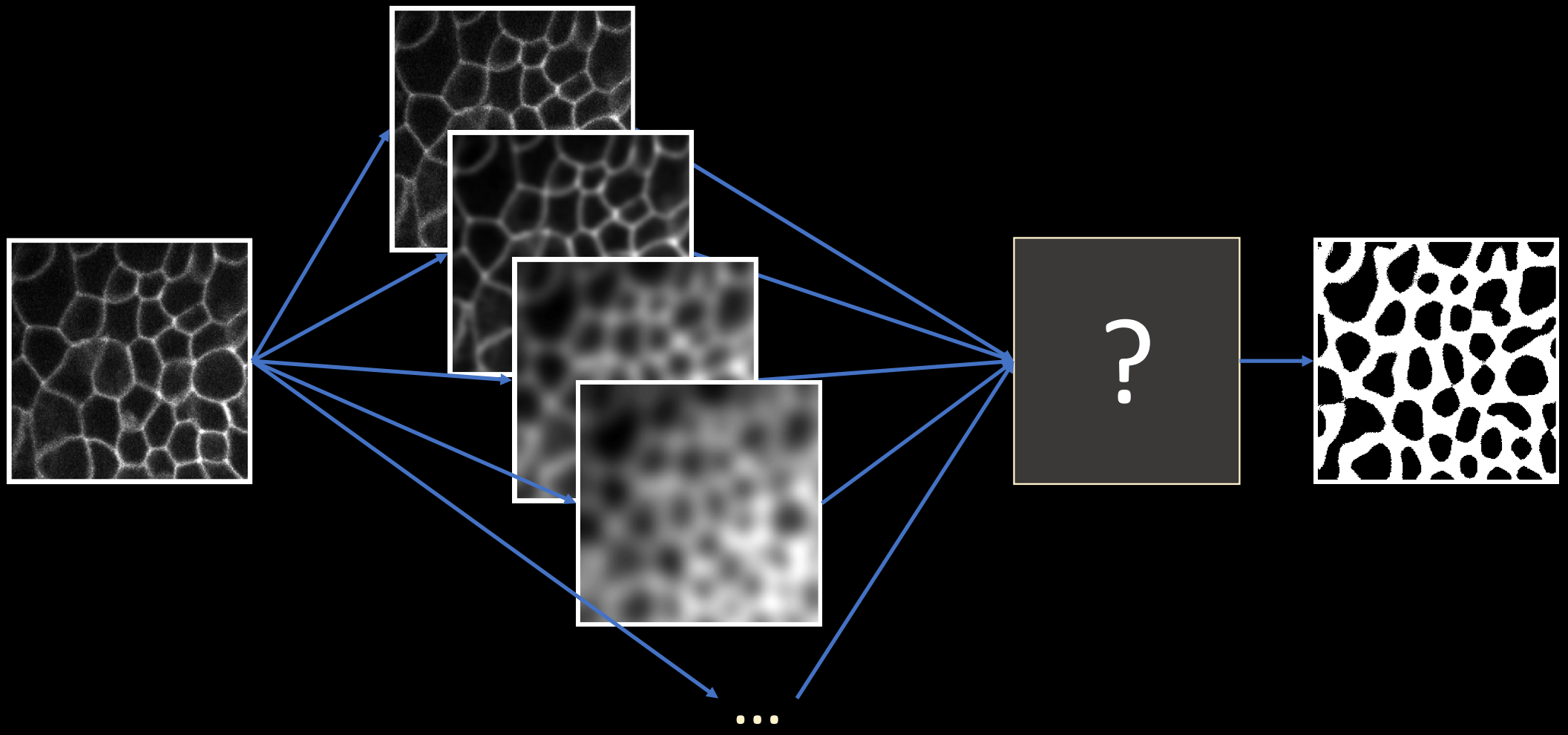
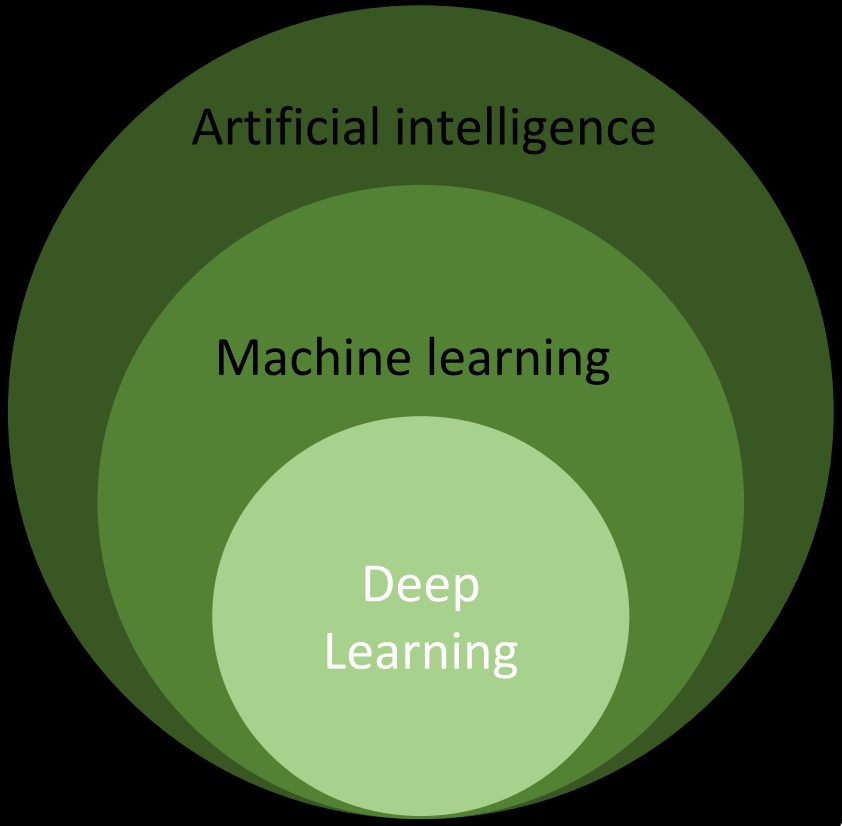
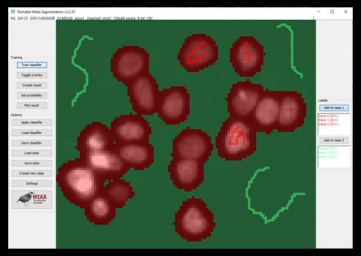


Image Segmentation using thresholding

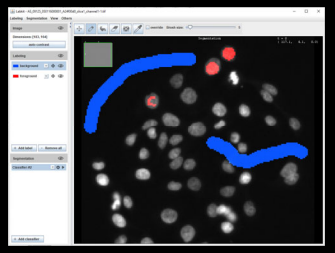


Machine Learning

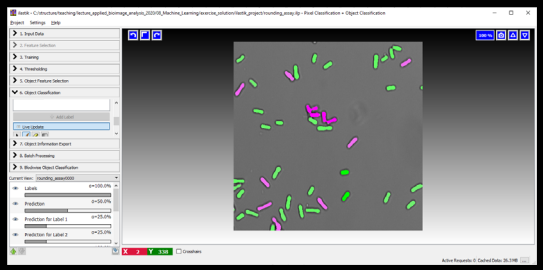




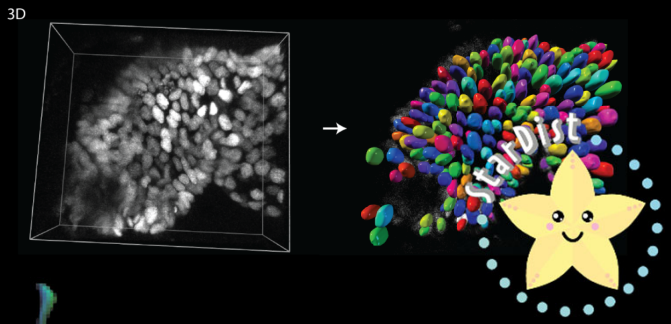
Trainable Weka Segmentation



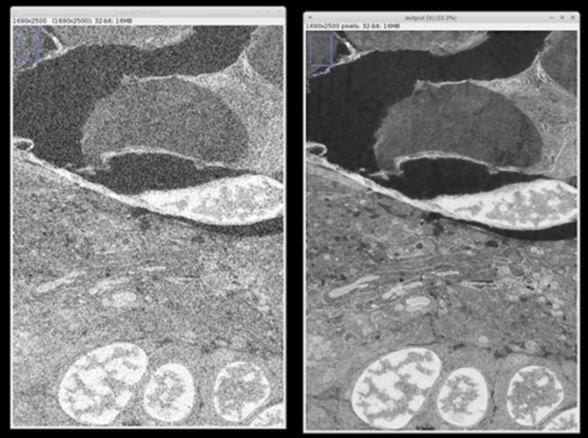
LabKit



ilastik



Cellpose

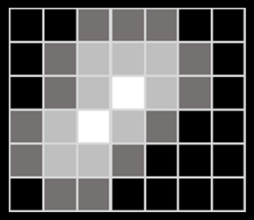


Noise2Void

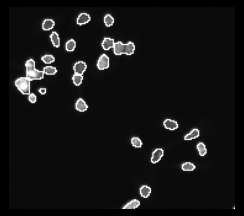
Machine Learning

- Automatic construction of predictive models from given data

Pixels,

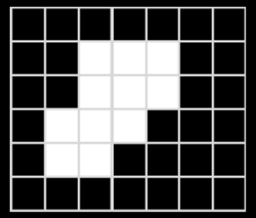


Objects,



Images,

Dense Segmentation / Binarization



Object classification

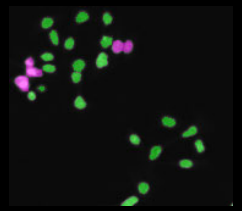
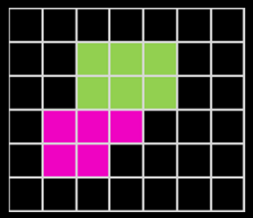
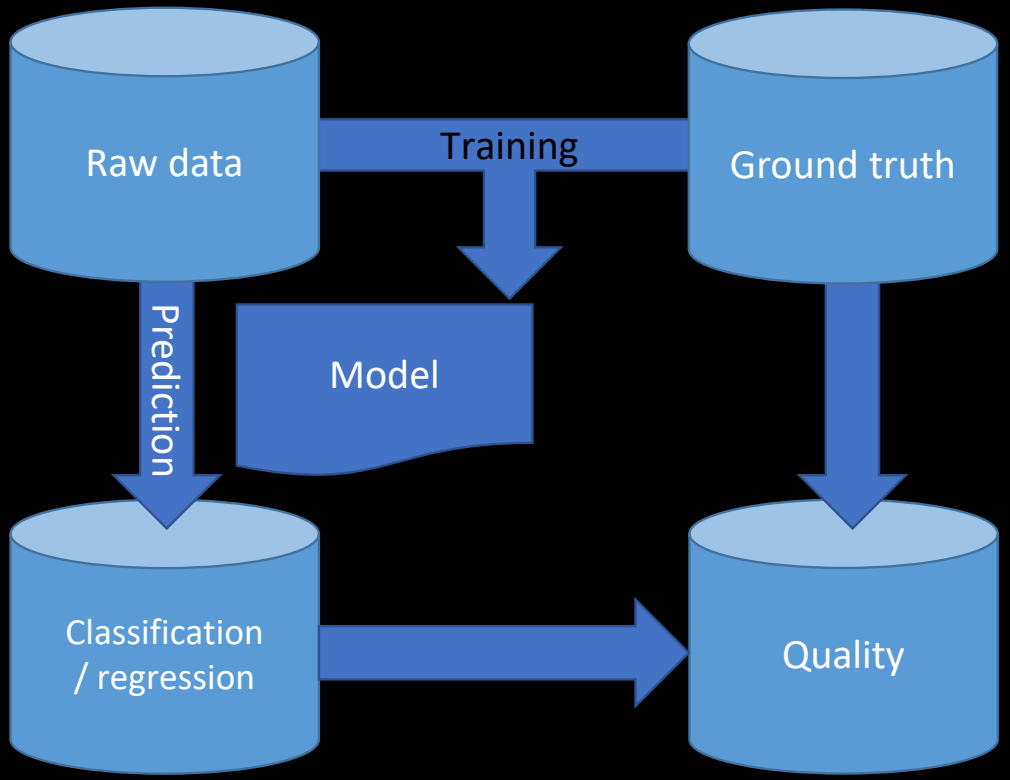


Image classification

"Cat"

Instance segmentation



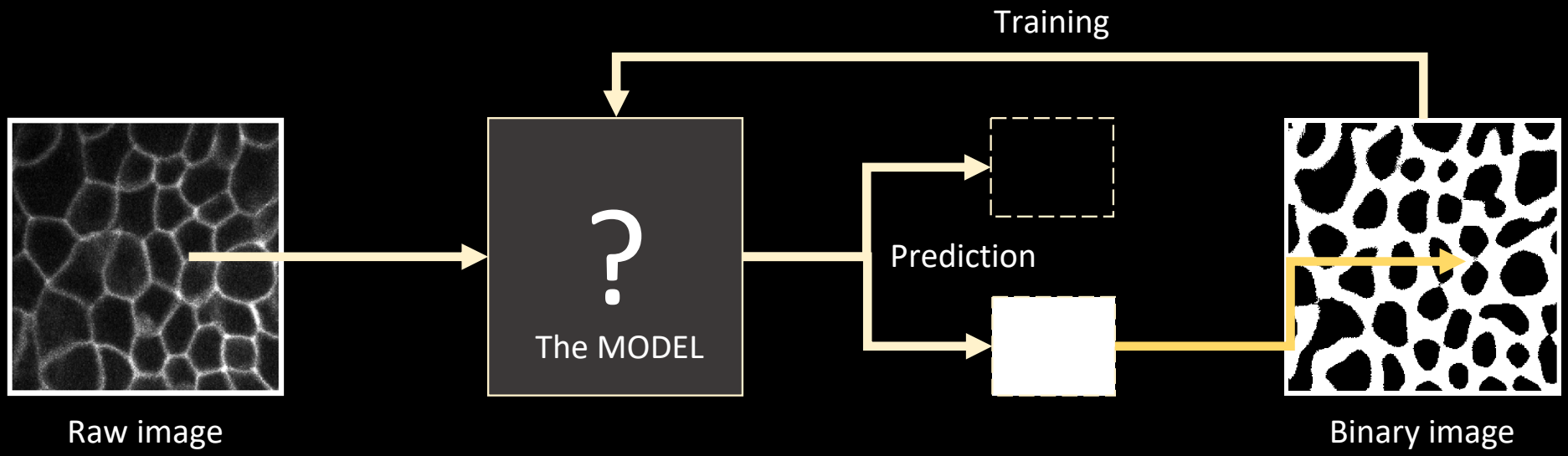


Annotated raw data, usually generated by humans

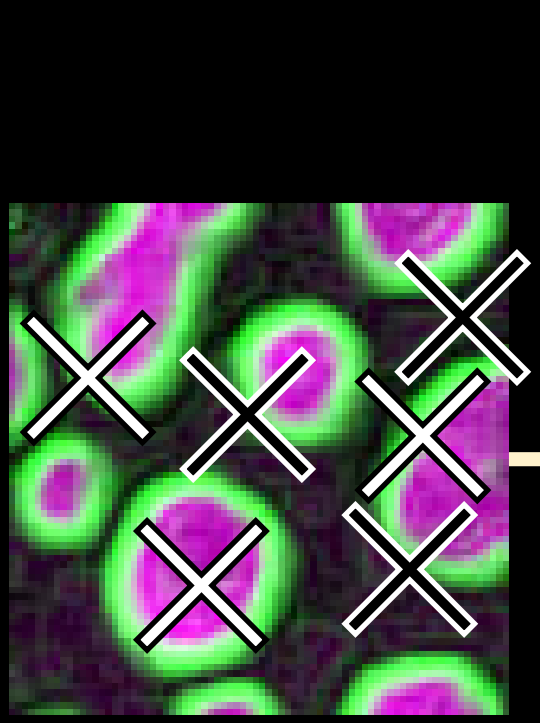
Precision, Recall

Machine learning for image segmentation

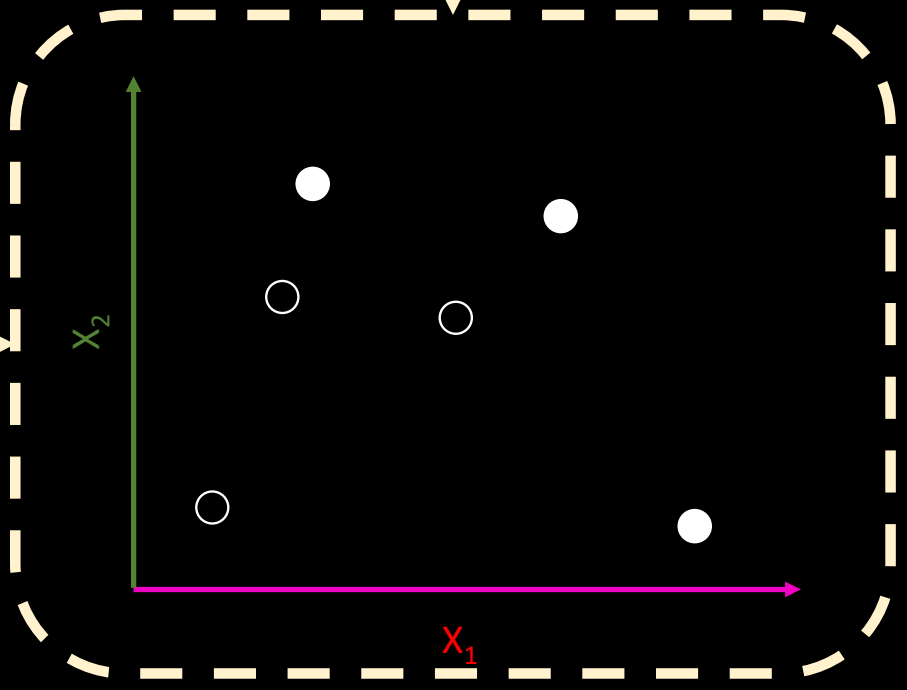
- Mostly *supervised learning*
- The computer derives a *model* or a *classifier* which can judge if a pixel should be foreground (white) or background (black)



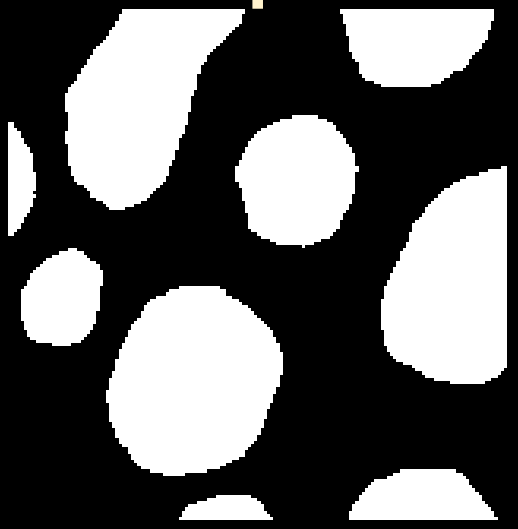
Supervised learning for pixel classification



Image

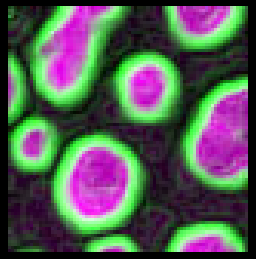


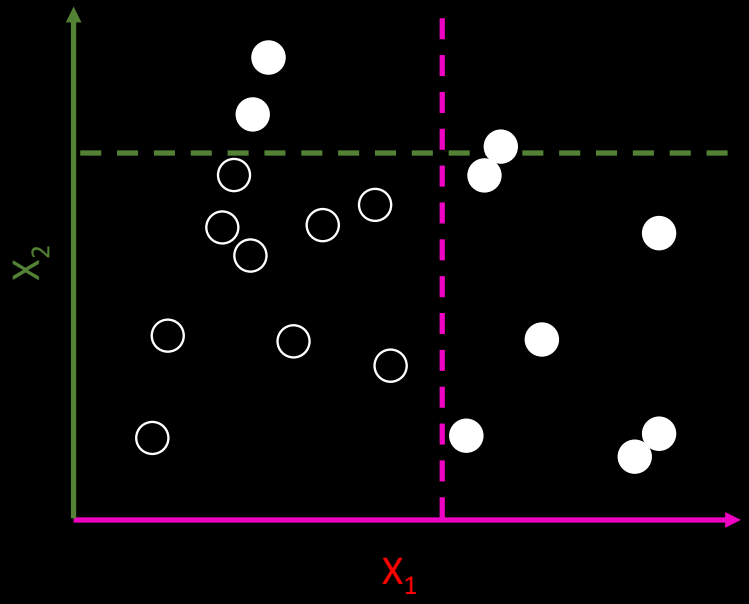
MODEL TRAINING



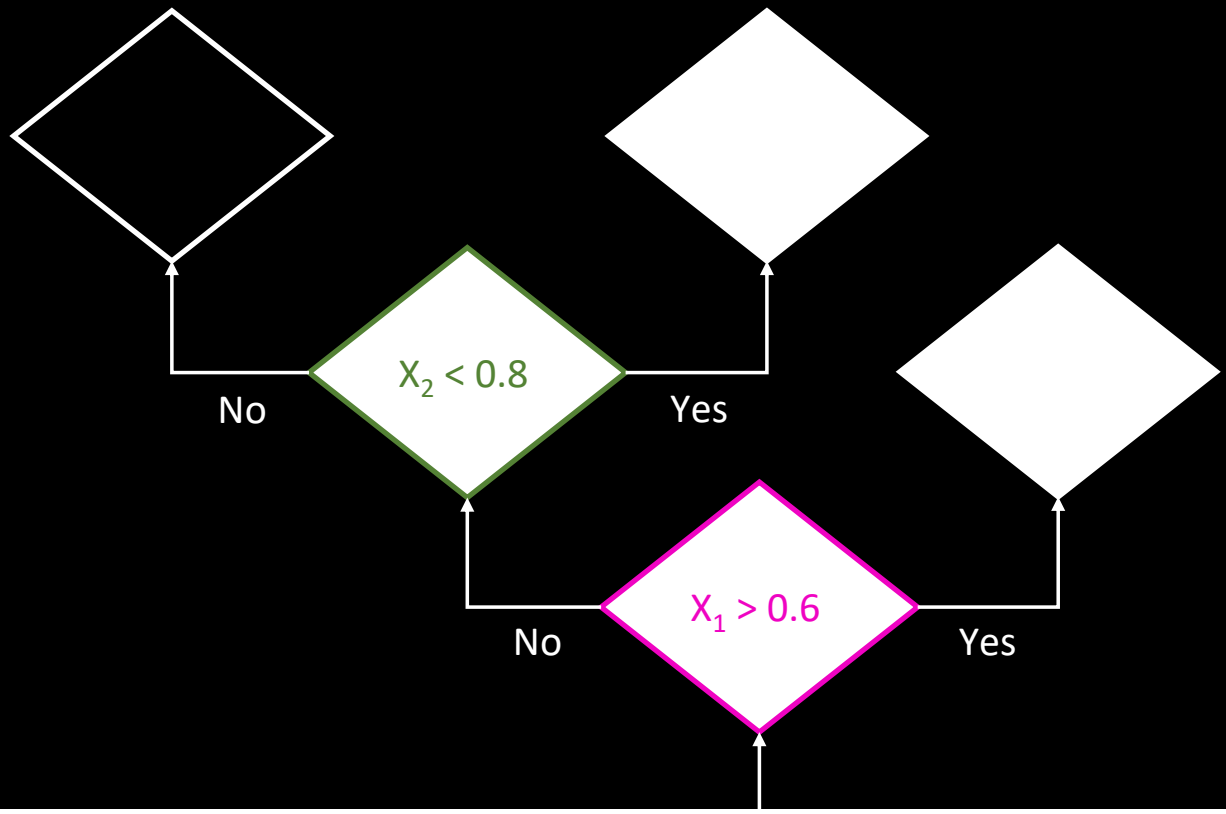
Ground Truth

Supervised learning for pixel classification

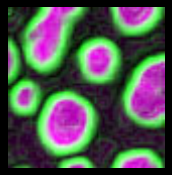




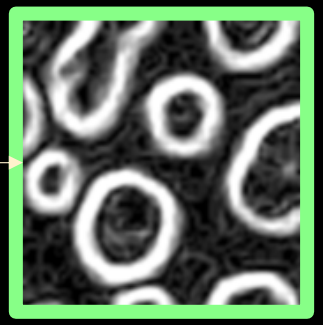
(Binary) Decision Tree

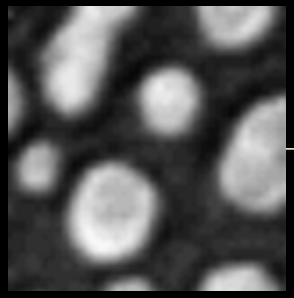


Supervised learning for pixel classification



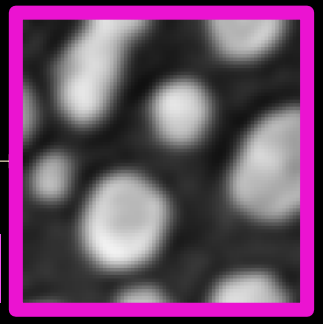
Sobel Filter

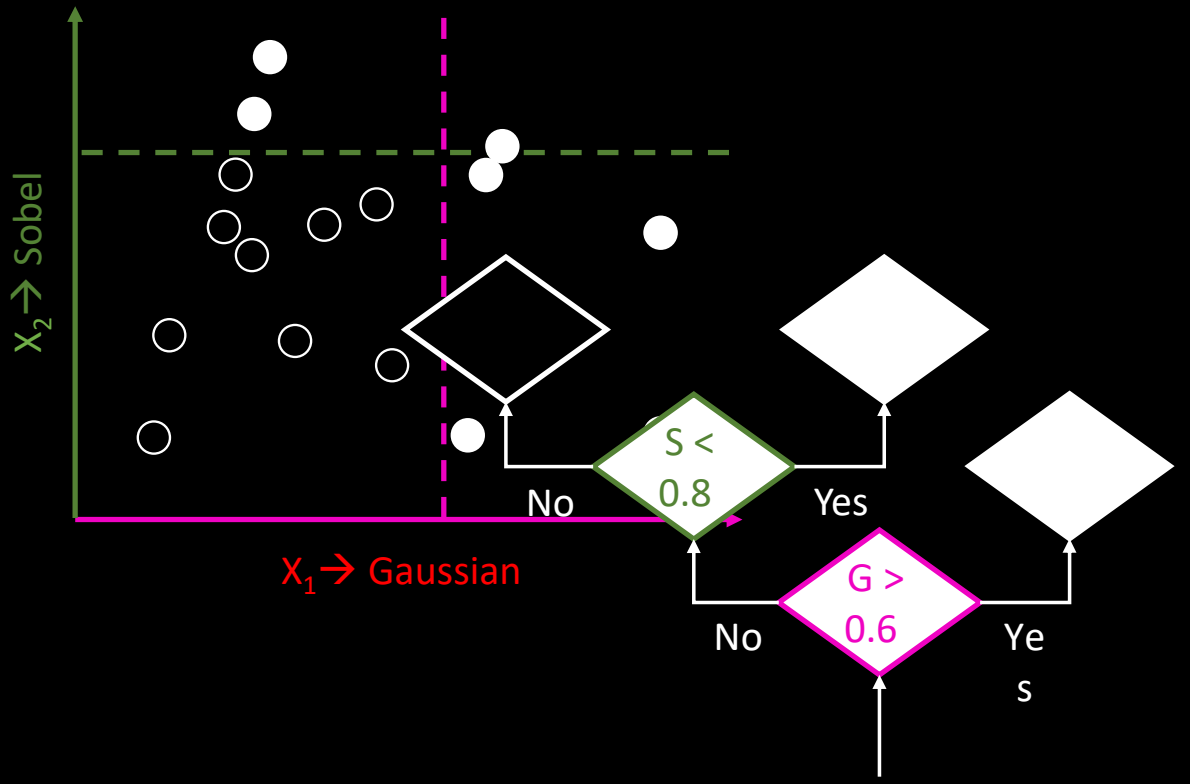




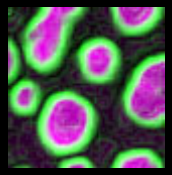
Original image

Gaussian Filter

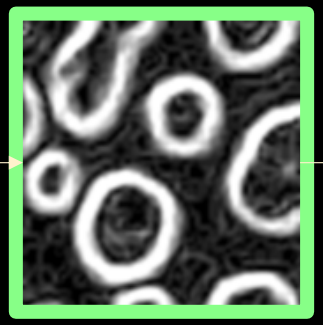


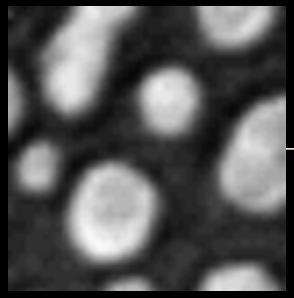


Supervised learning for pixel classification



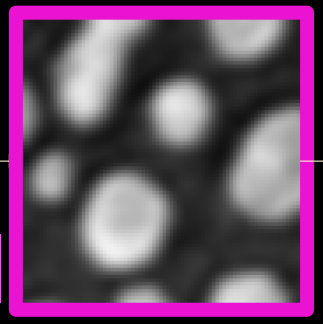
Sobel Filter





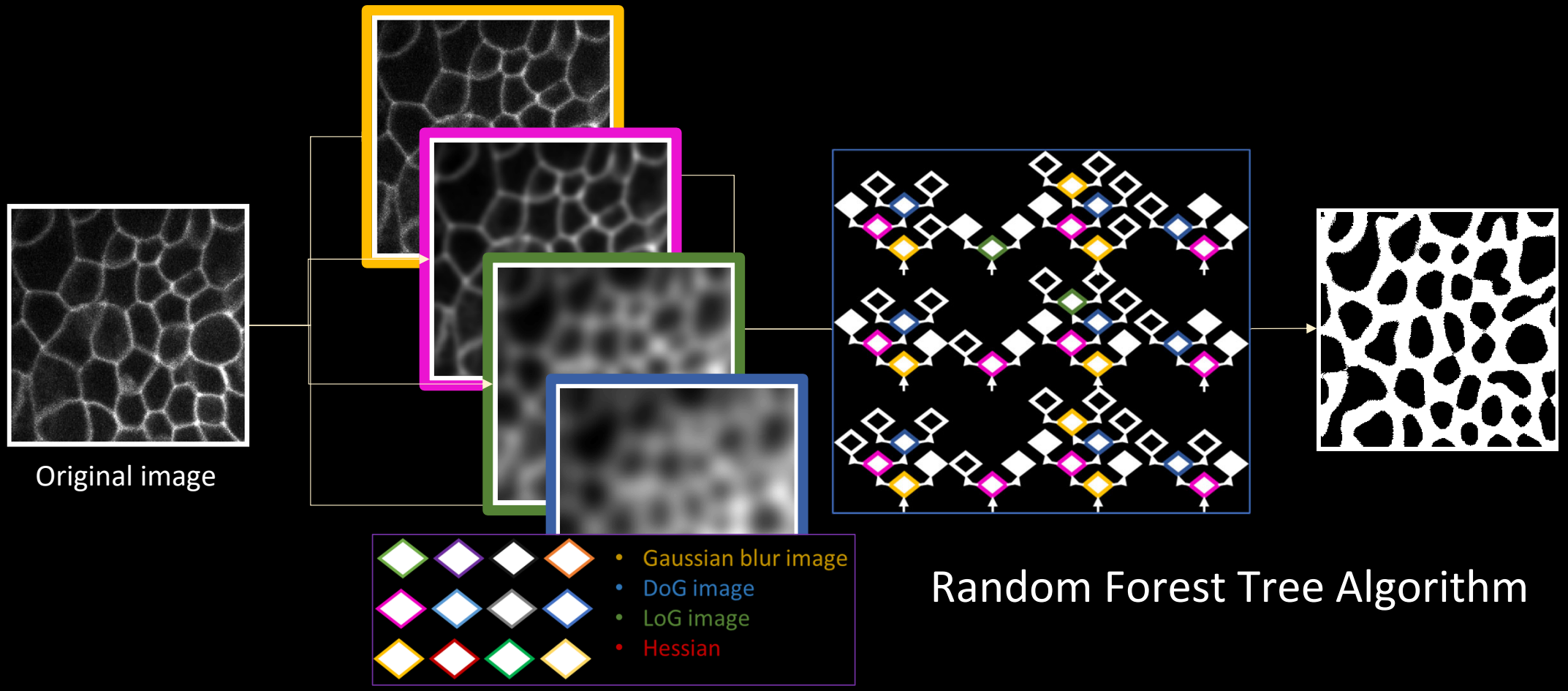
Original image

Gaussian Filter



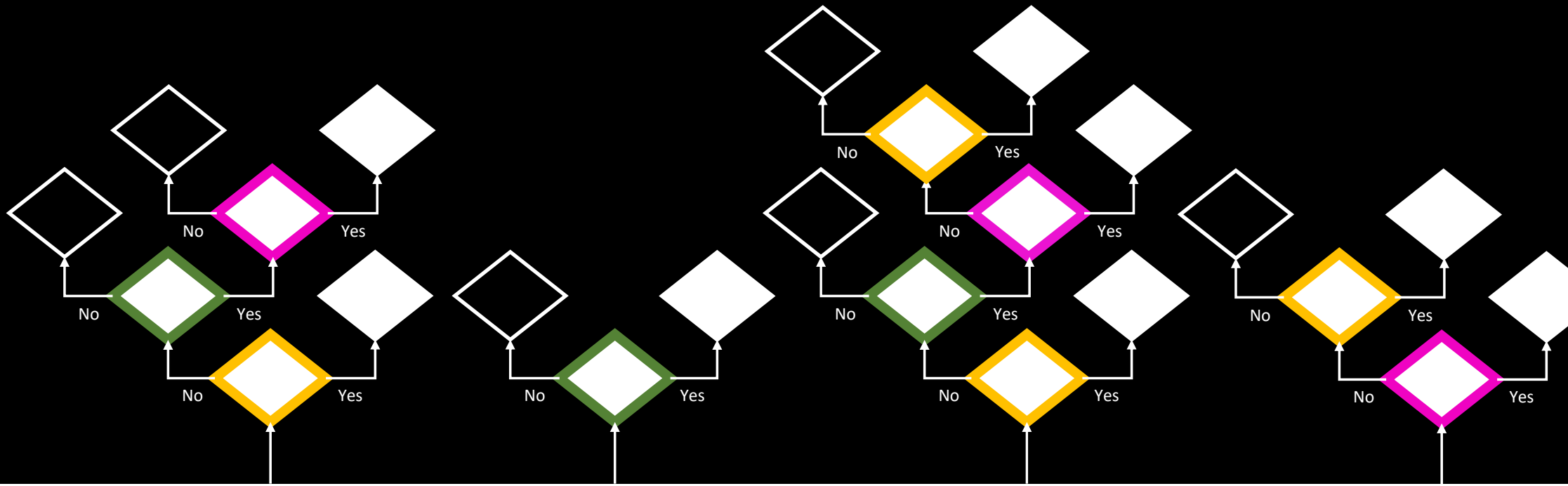
?

Supervised learning for pixel classification

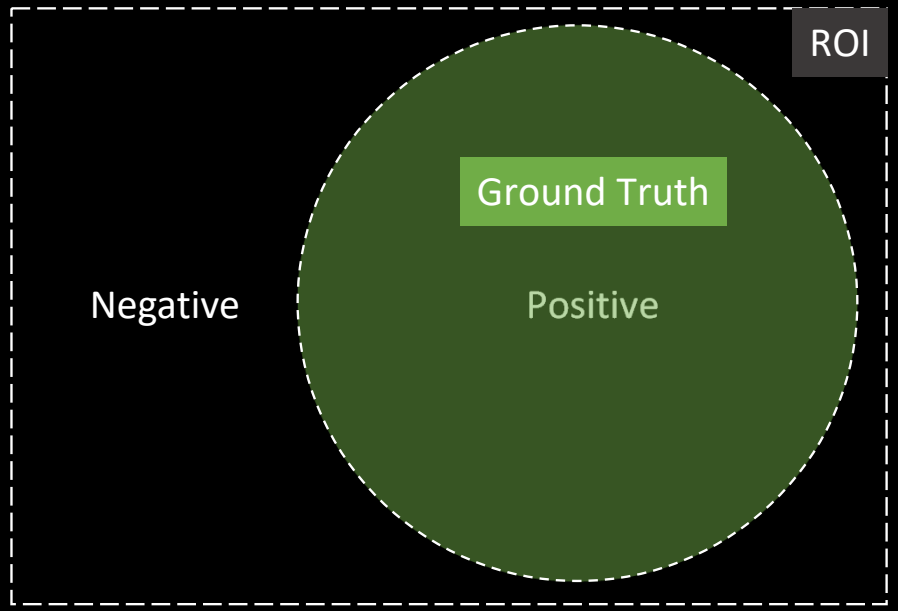


Brief Introduction to the Random forest based image segmentation

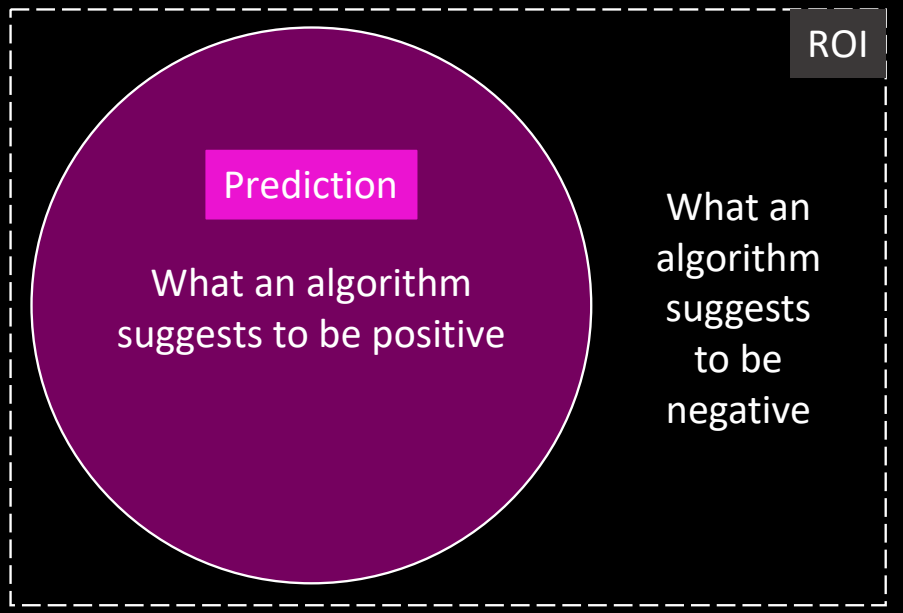
- Decision trees are classifiers, they decide if a pixel should be white or black
- Random decision trees are randomly initialized, afterwards evaluated and selected
- Random forests consist of many random decision trees



Algorithm evaluation

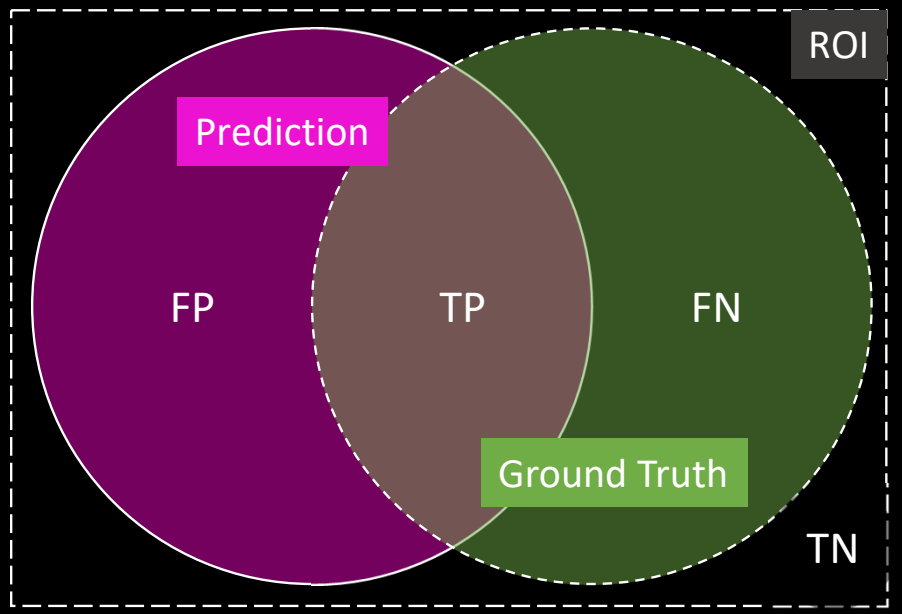


Algorithm evaluation



Algorithm evaluation

- In general
  - Define what's positive and what's negative.
  - Compare with a reference to figure out what was true and false
  - the Theory of Sets



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

What fraction of points that were predicted as positives were really positive?

Recall (a.k.a. sensitivity) $\frac{TP}{TP + FN}$

What fraction of positives points were predicted as positives?

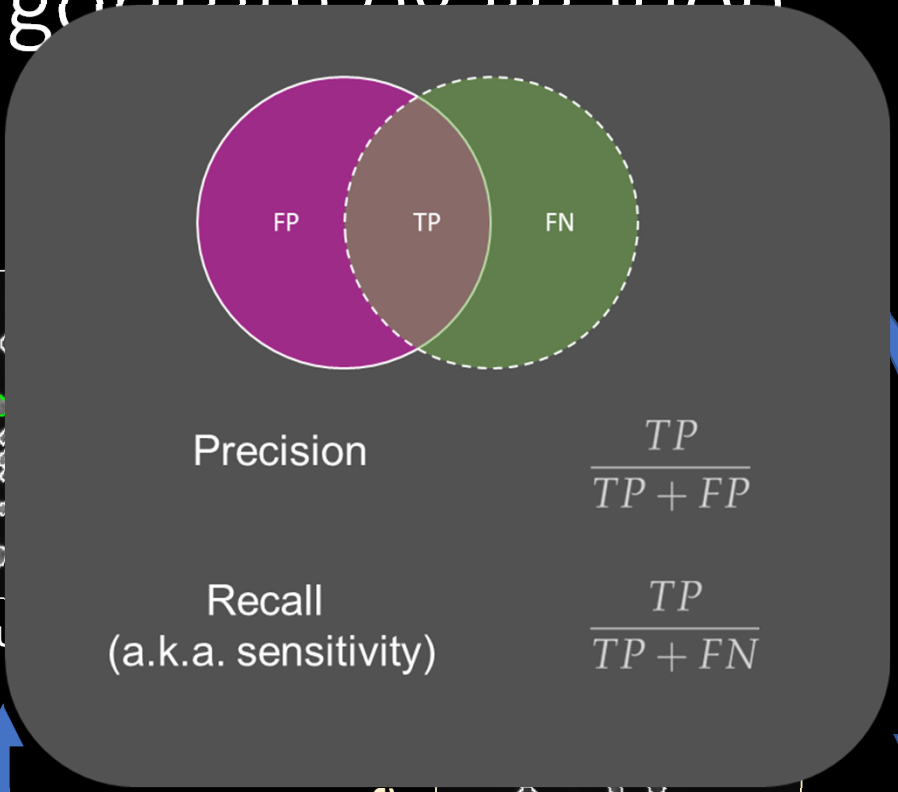
TP True-positive

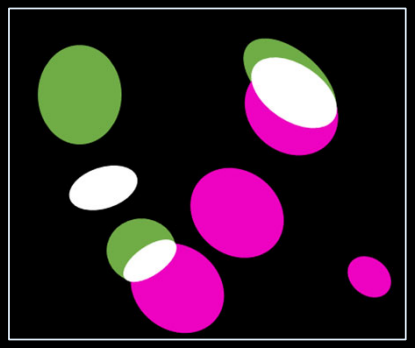
FN False-negative

FP False-positive

TN True-negative

Algorithm evaluation





Objects with at least 50% *pixel-wise overlap* between Prediction and Ground Truth

True positive: 2
False positives: 3
False negatives: 2

Precision: 2/5 = 40%
Recall: 2/4 = 50%

Training set

Grou

Data

Test se

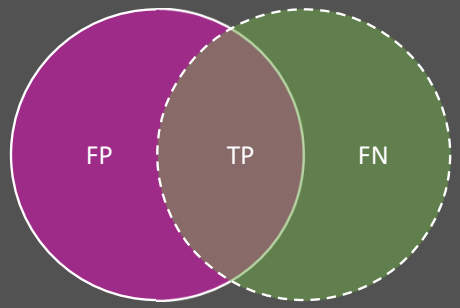
Raw data

Prediction

Prediction

Algorithm Evaluation

Ground truth



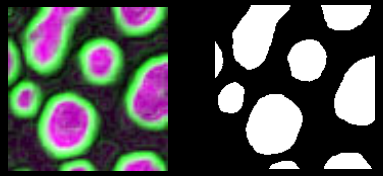
Precision $\frac{TP}{TP + FP}$

Recall
(a.k.a. sensitivity) $\frac{TP}{TP + FN}$

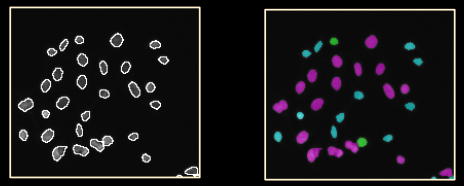
Object classification

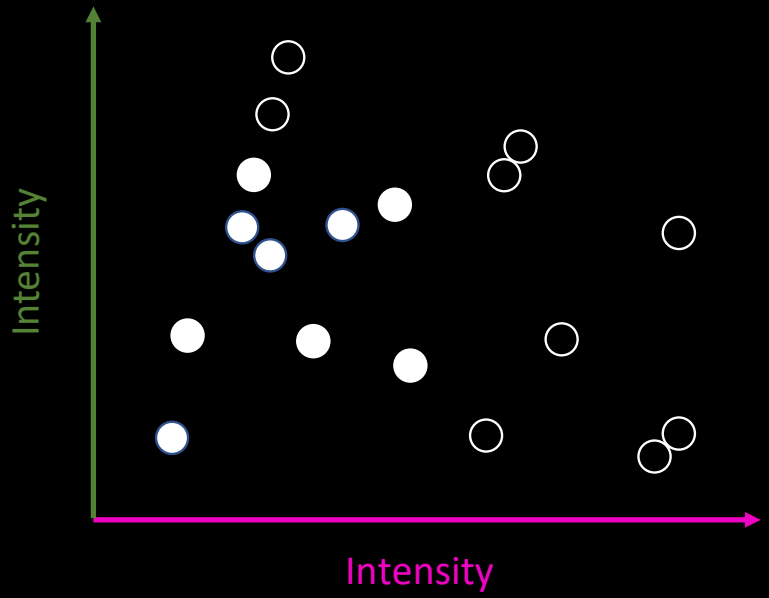
- What if we exchange pixel features with object features?

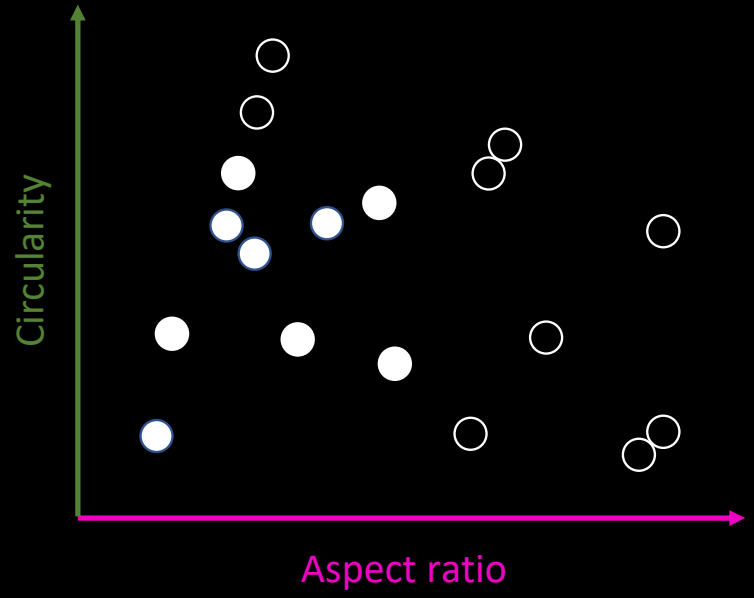
Pixel classification



Object classification



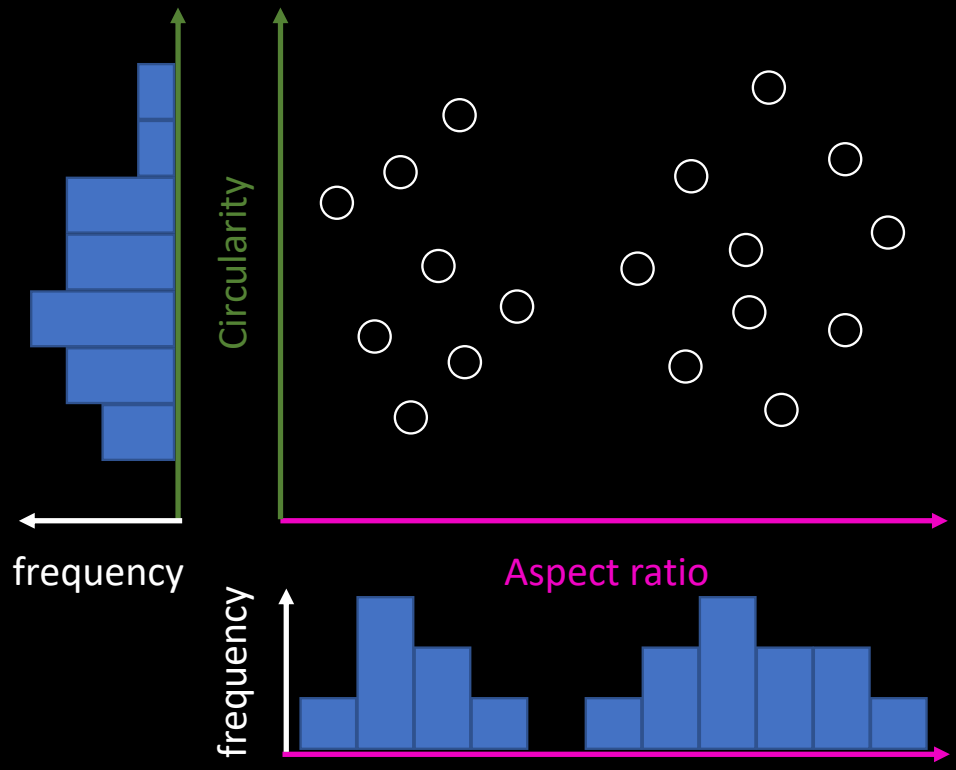




- The algorithms work the same but with different
  - Features
  - Number of features
  - Tree / forest parameters
  - Selection criteria

Unsupervised machine learning

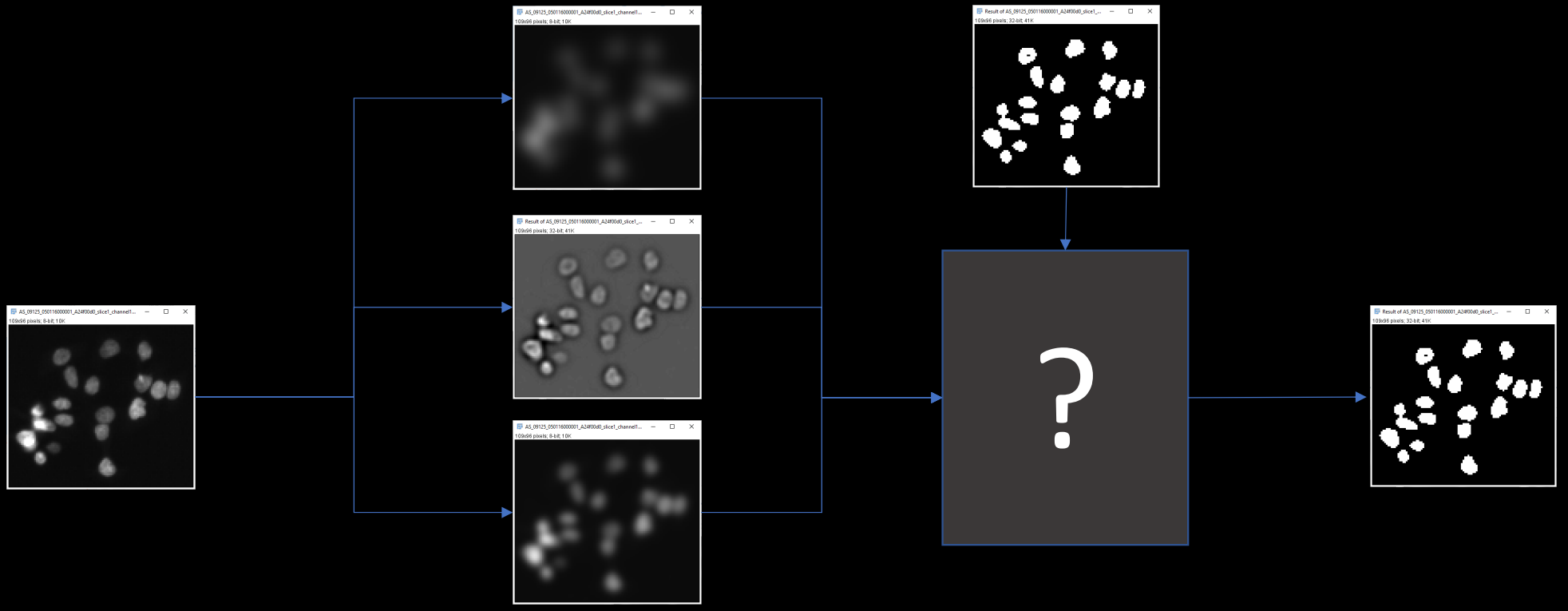
- If you don't provide ground truth, the algorithm is *unsupervised*.



Further reading:
- Principal component analysis
- Cluster analysis

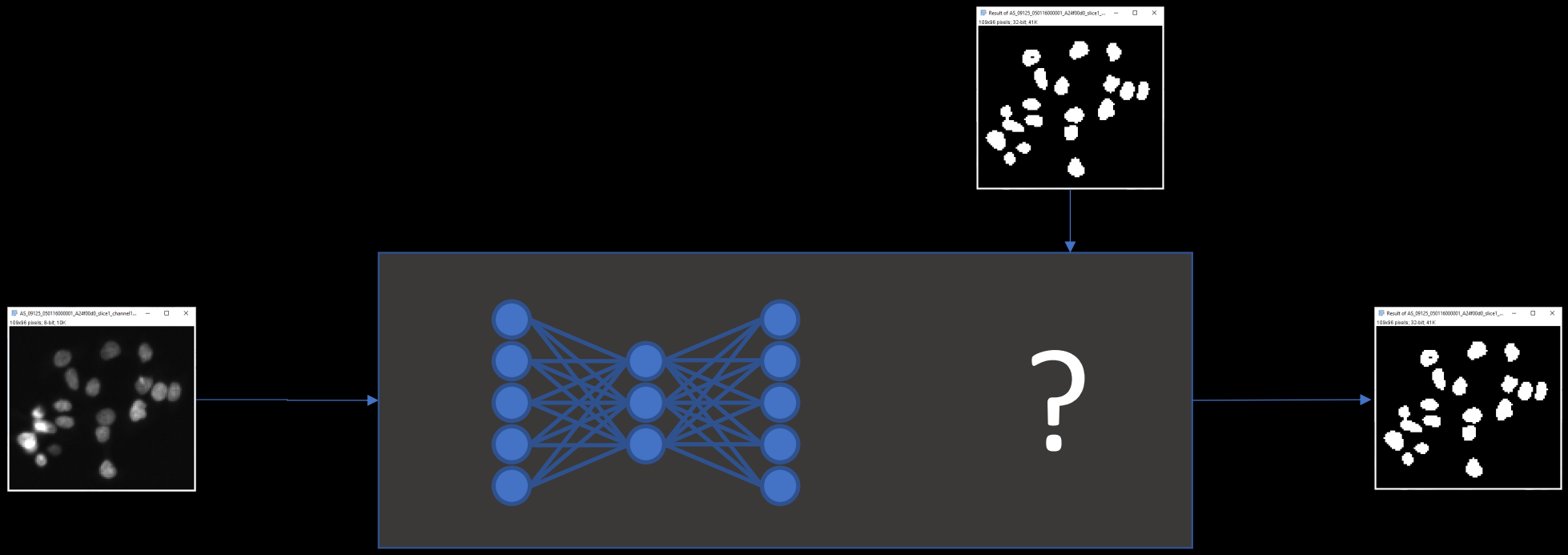
Machine learning for image analysis

- In machine learning, we typically select features for training our classifier



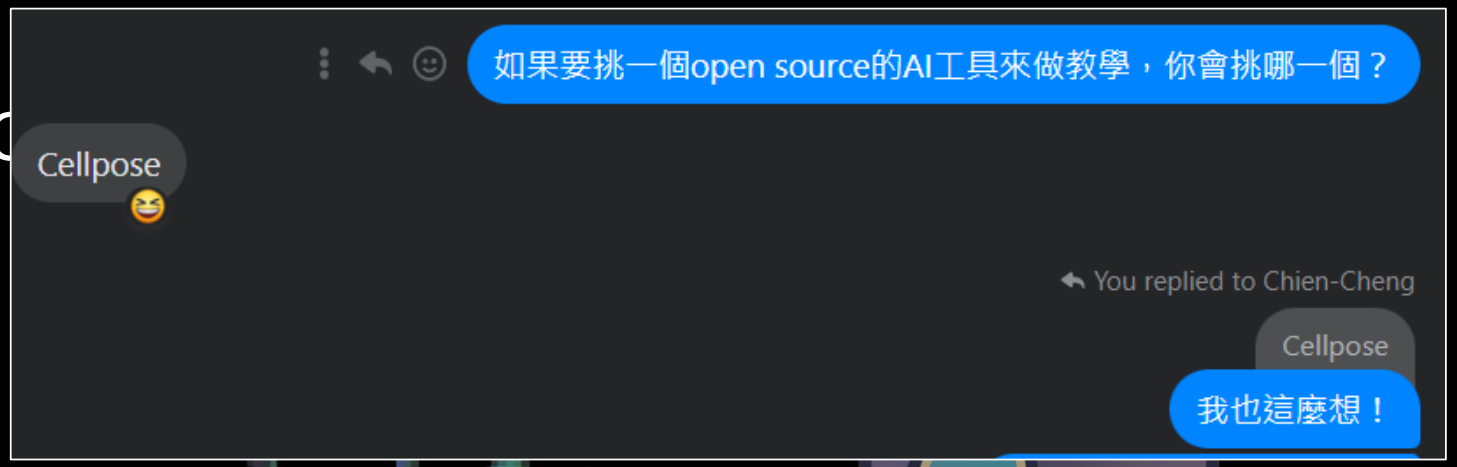
Deep learning for image analysis

- In deep learning, this selection becomes part of the black box



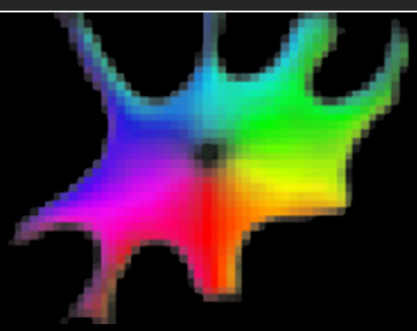
Convolutional neural networks

Introduction to



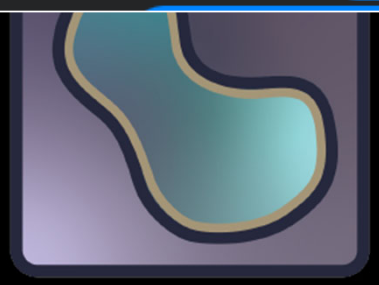
ilastik

Image segmentation
Object classification
Etc. etc.



cellpose

Image segmentation



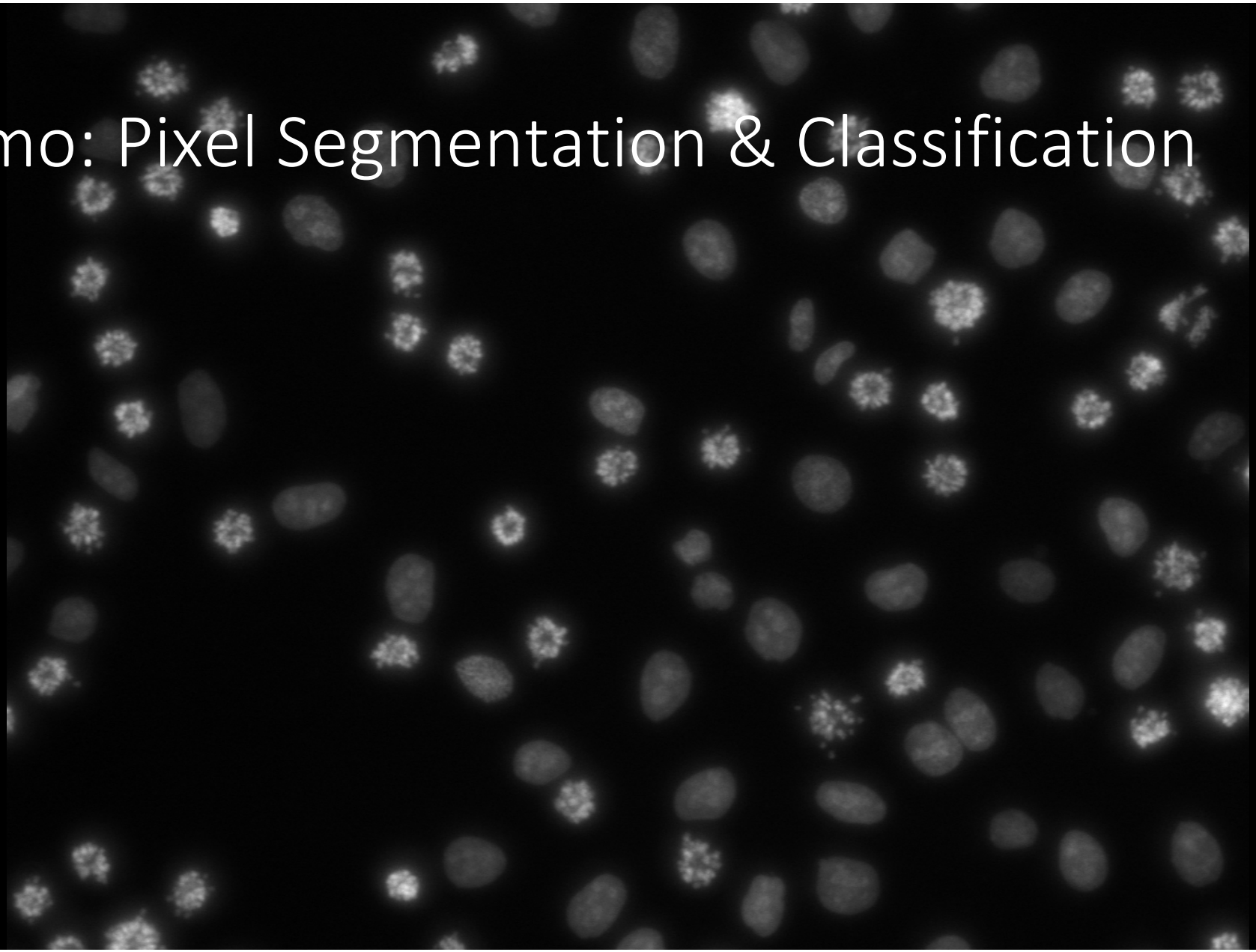
napari

Volumetric visualization
“Becoming” the python imageJ

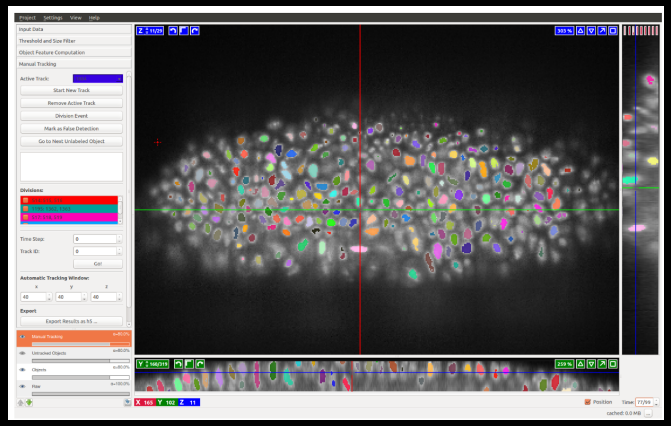
ilastik

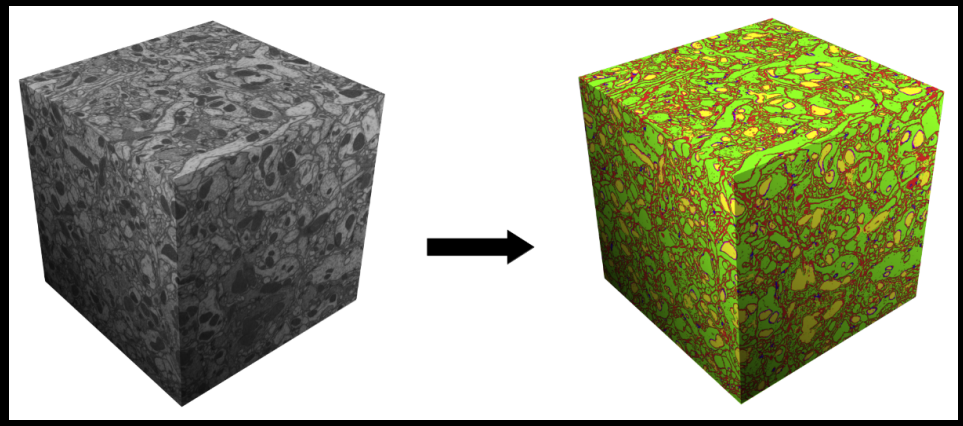
- Developed by Anna Kreshuk et al @EMBL Heidelberg
- User-friendly platform for the analysis and visualization of multi-dimensional image data.
- provides a range of tools for tasks such as object classification, object tracking, and image segmentation.

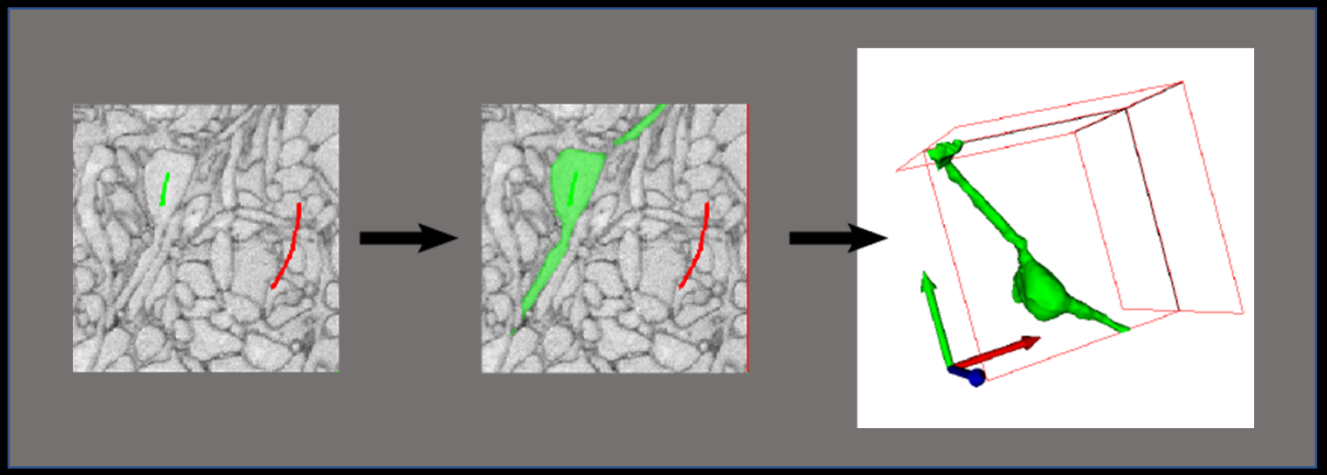
Demo: Pixel Segmentation & Classification

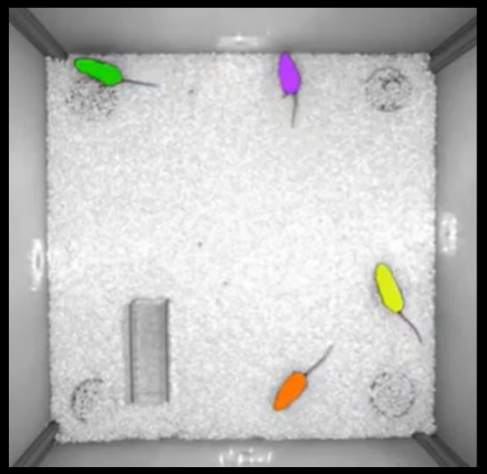


Other things ilastik can do

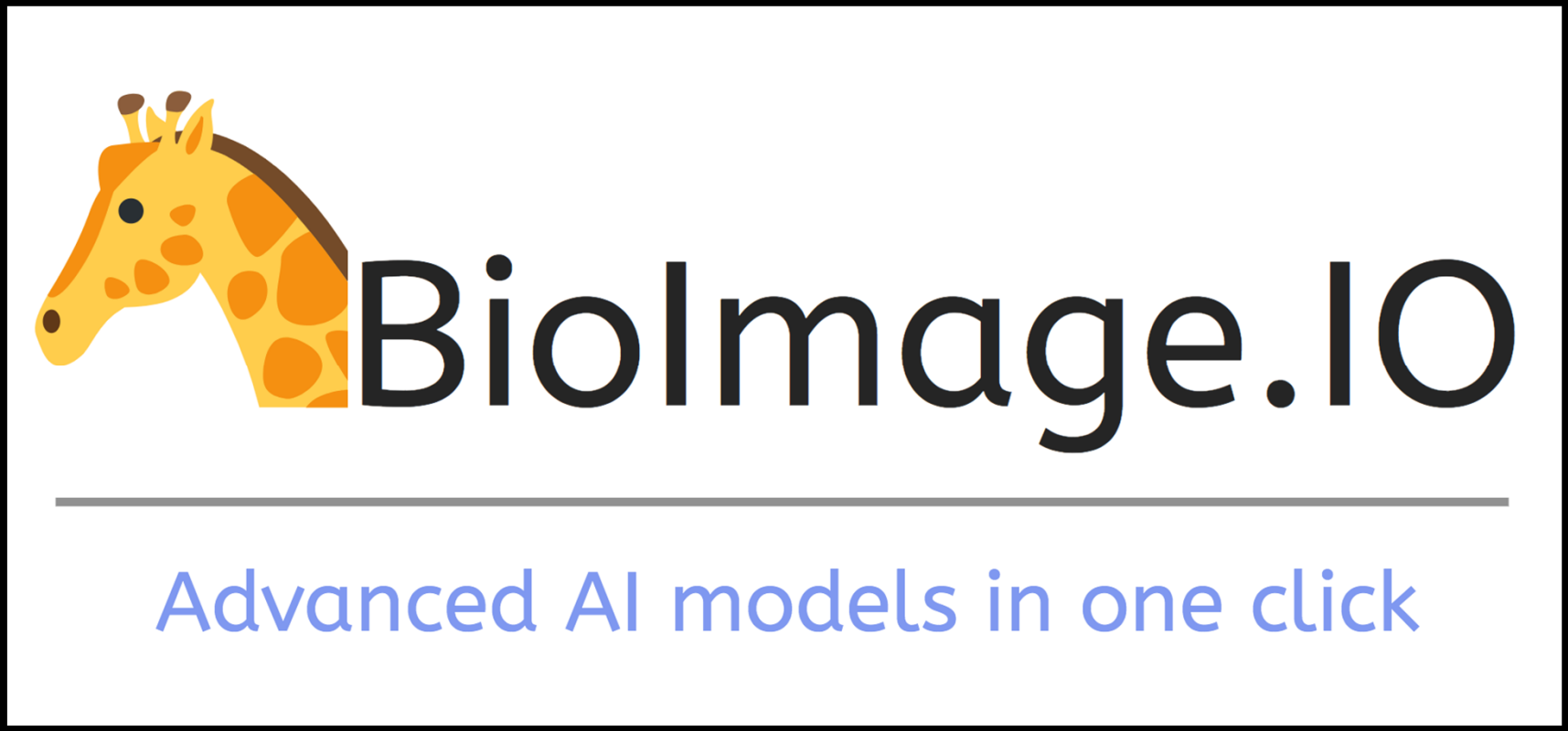








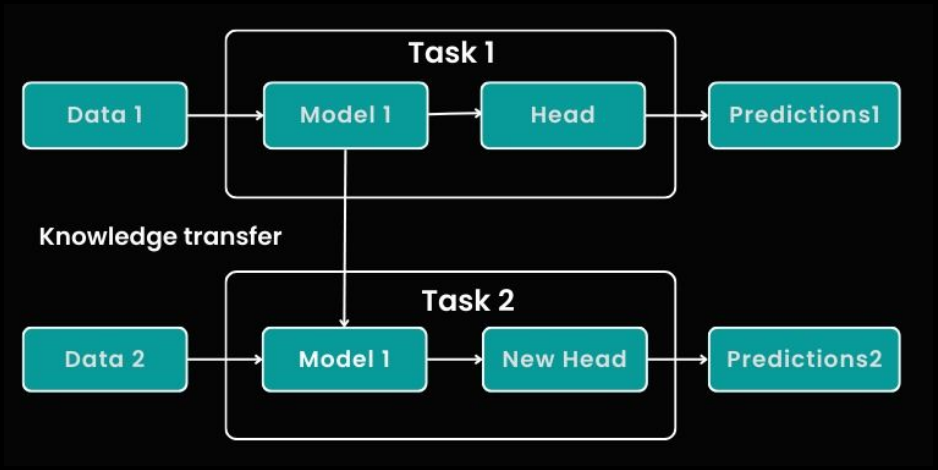
BioImage.IO

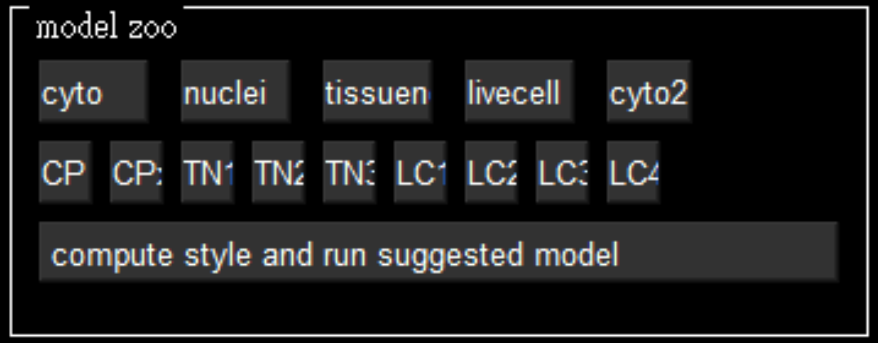


Cellpose

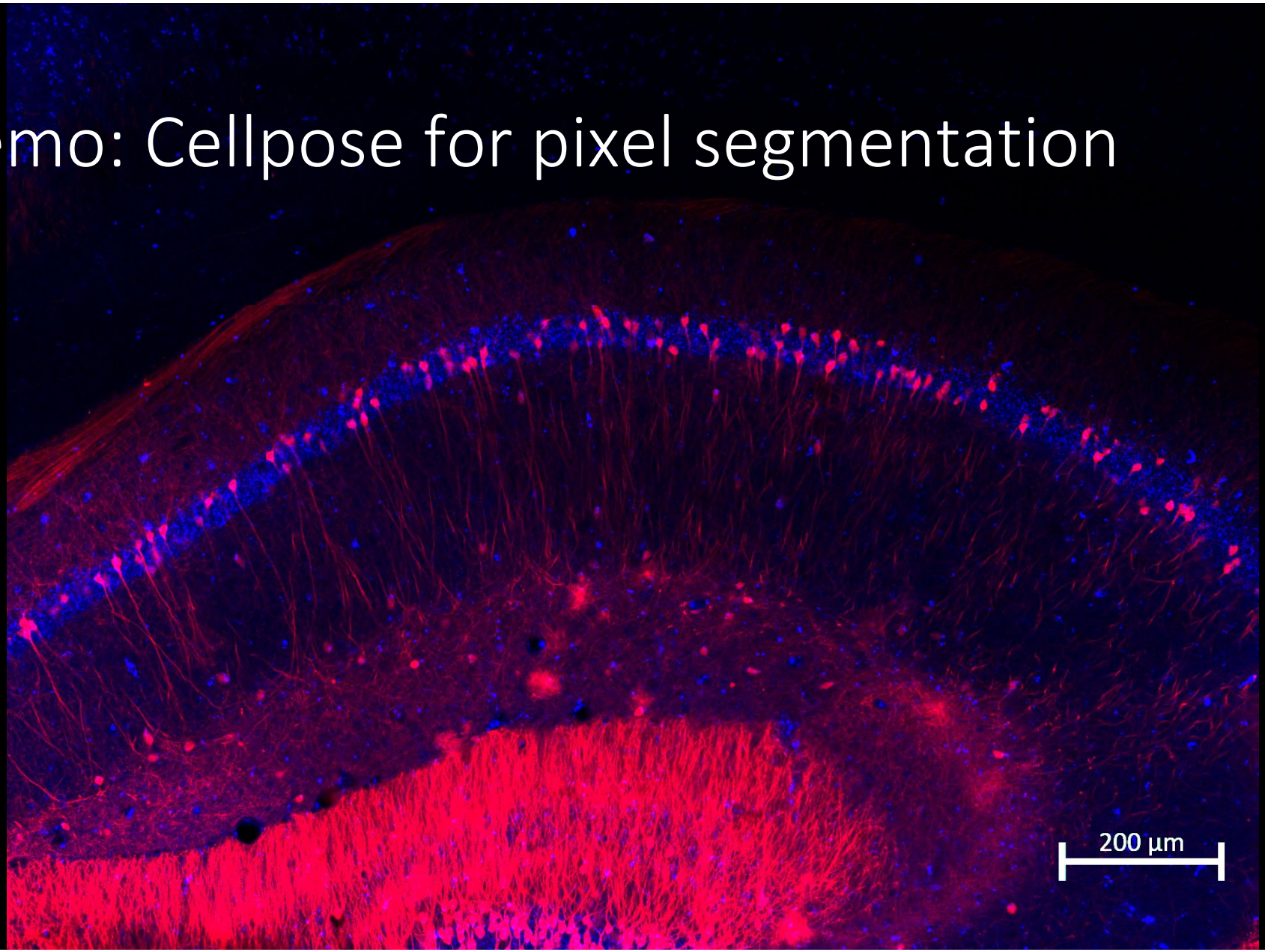
- Developed by Carsen Stringer and Marius Pachitariu @ HHMI Jenelia
- designed to accurately segment and classify cells in images
- provides a range of options for customizing the analysis, including the **ability to choose between different models** and to set various parameters for the segmentation process
- **leverages the power of Transfer Learning**

Transfer Learning





Demo: Cellpose for pixel segmentation



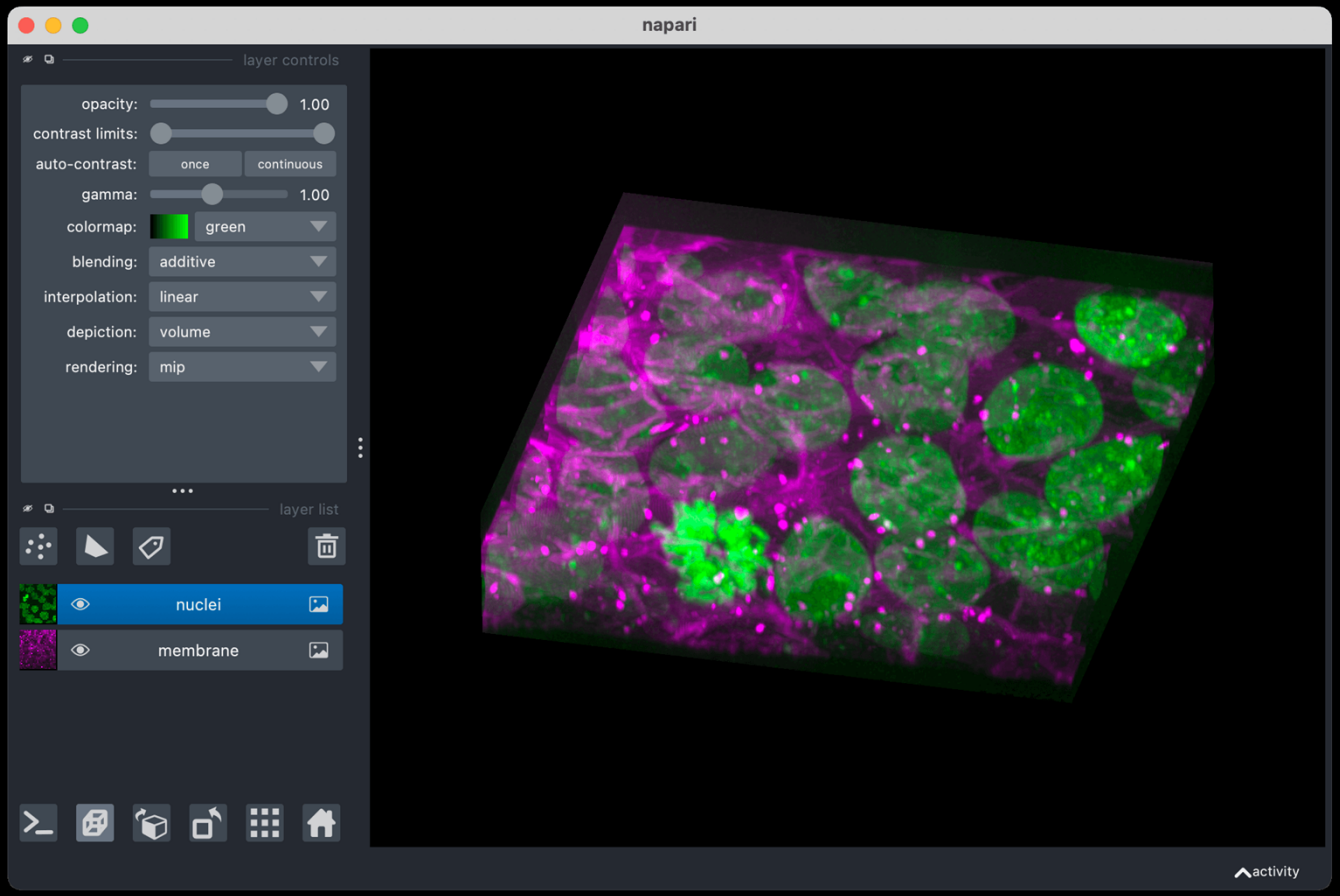
Cellpose or ilastik or …?

- There are more tools: DeepImageJ, StarDist, etc. etc.
- Depends on your use case
- Try your sample with one of them, tweak parameters a bit, see how it perform
- **My very naïve advice**: try cellpose first, if doesn't work well, go ilastik
- **Or**: try to be very familiar with one of the "General Purpose" tools, so every time when you got an image, try that first

napari

- Developed by Robert Haase et al @ TU Dresden
- for visualizing and interacting with multi-dimensional image data.
- a user-friendly platform for the exploration and analysis of images, and it provides a range of features for tasks such as annotation, measurement, and image processing.
- “Becoming” the python imageJ

Demo: 3D data visualization with Napari

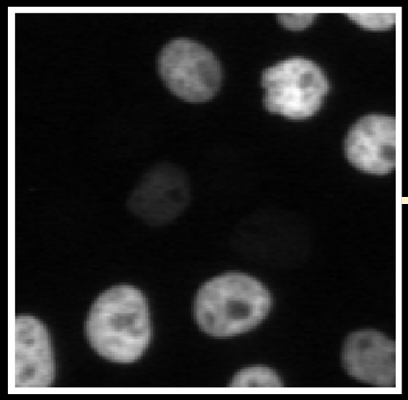


Bio-statistics

Important… but I won’t cover it in this workshop

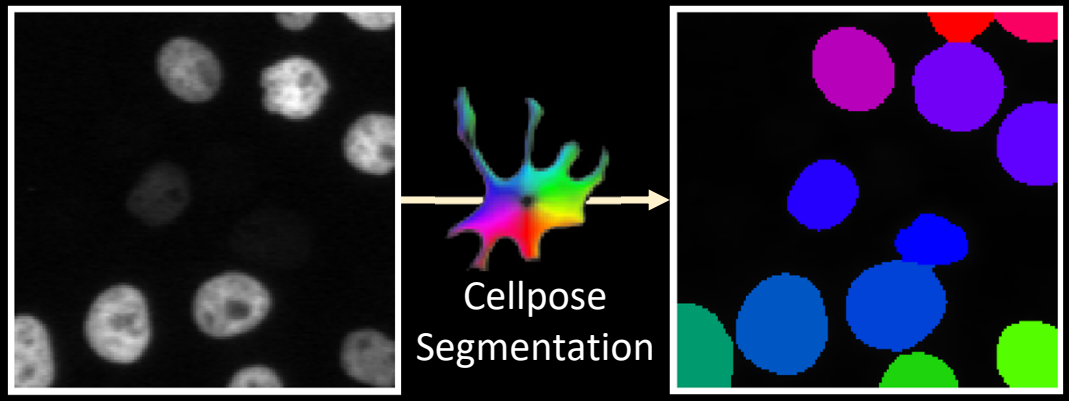
Codelab

In the last codelab...



Adaptive Filtering
Distance Transform
Connected Components Analysis
Watershed Transform
...

In this code lab…



Cellpose is GPU supported

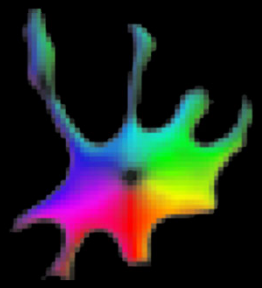
- Don’t have a GPU? Try **Google colab!**
- Cloud-based jupyter notebook like environment, supported by Google
- Free access to GPUs!

- Still better to have your own *local* GPUs
- Google colab can be served as a test-bench

In this session, we will…

- Run Cellpose on Google colab to segment features in our images
- My suggestion:

colab

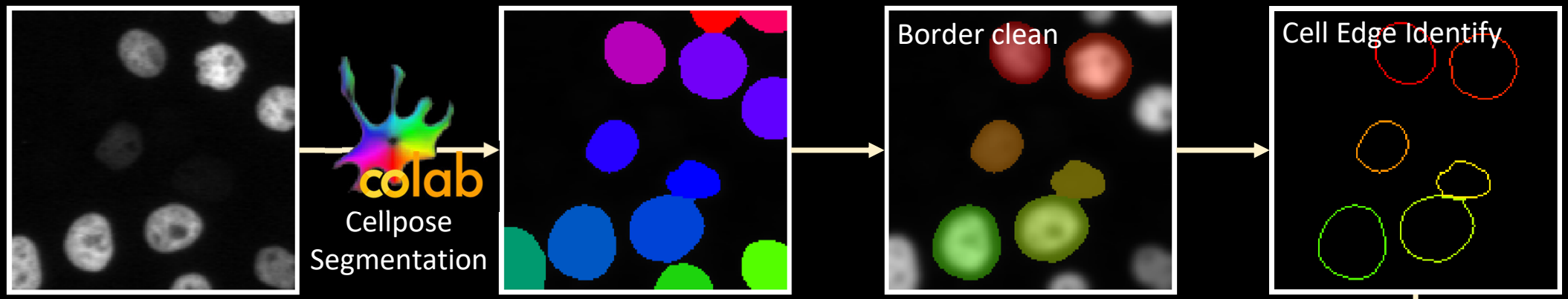


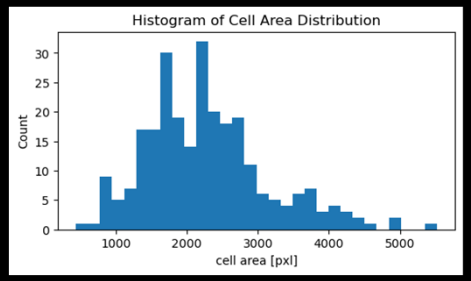
Visualization
Manual Labelling

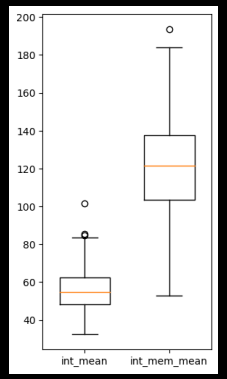
Model training
Segmentation

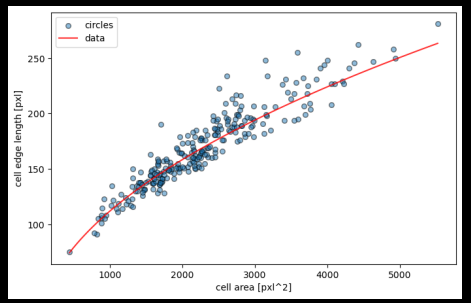
* You can run Cellpose locally if your computer is ok to run it

In this code lab…









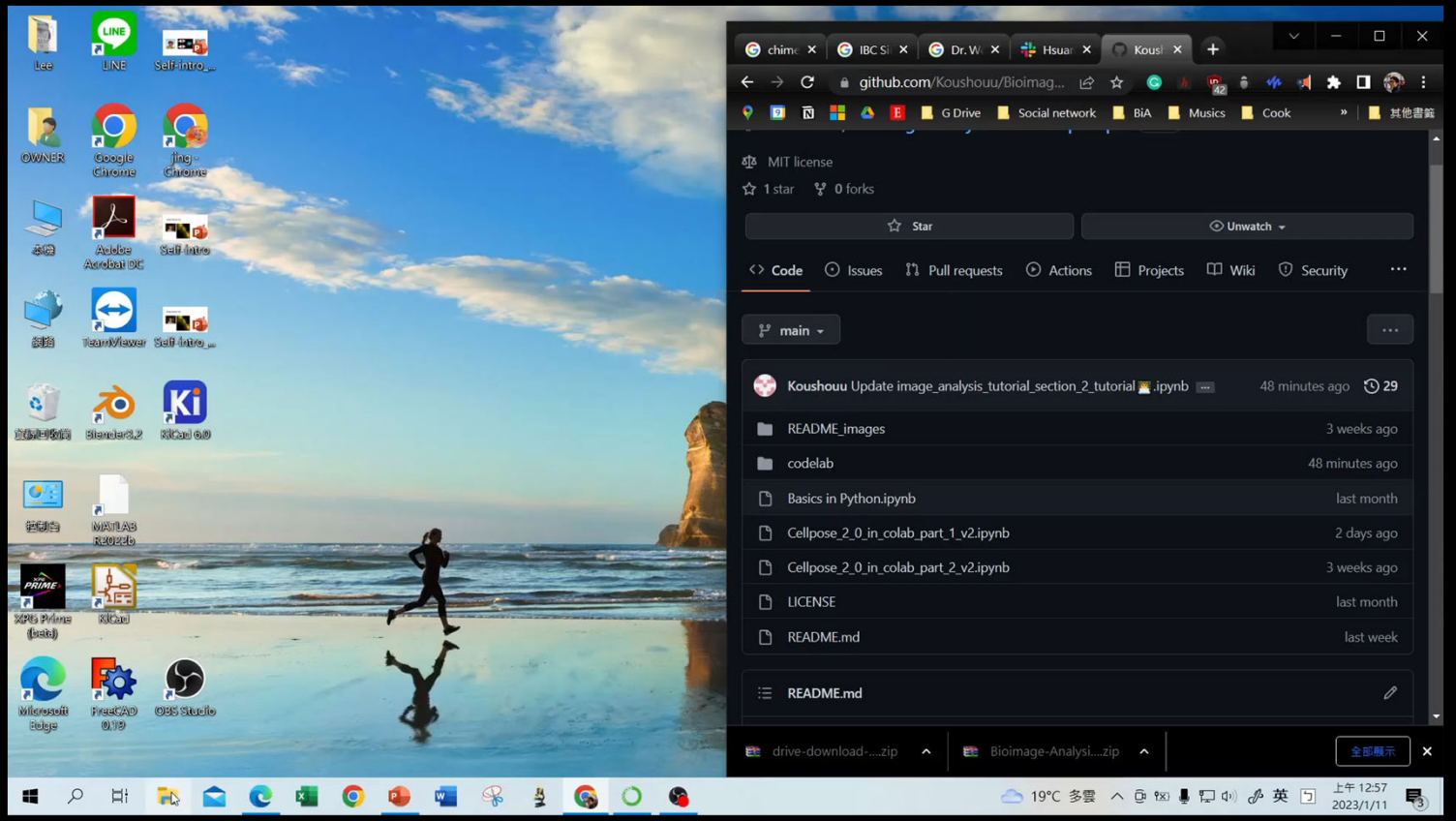
Biostatistics
with python

Coming next…

How to think like a bioimage analyst

DEMO: Cellpose on Google Colab

Alternatively, you can use Cellpose GUI



bit.ly/bioimage2023-cellpose