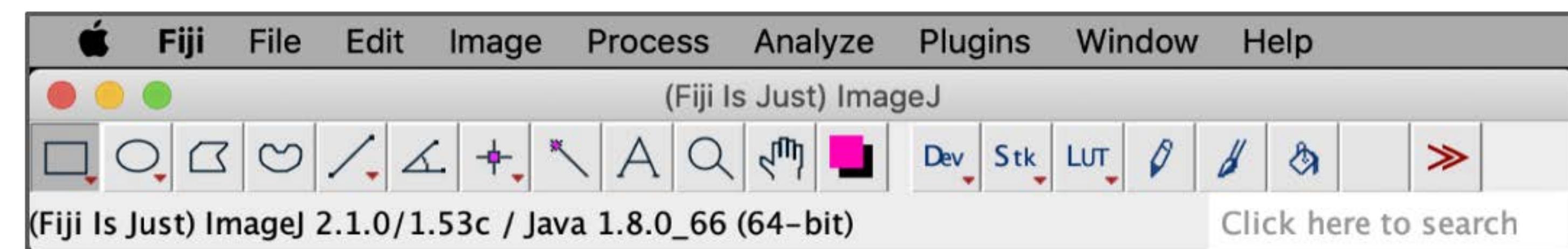


# Introduction to image analysis using ImageJ/Fiji



Federico Gasparoli, Ranit Karmakar, Antoine Ruzette, Maria Theiss & Simon F. Nørrelykke  
*Image Analysis Collaboratory, HMS*

# Get the course materials

<https://hms-iac.github.io/fiji-workshop>

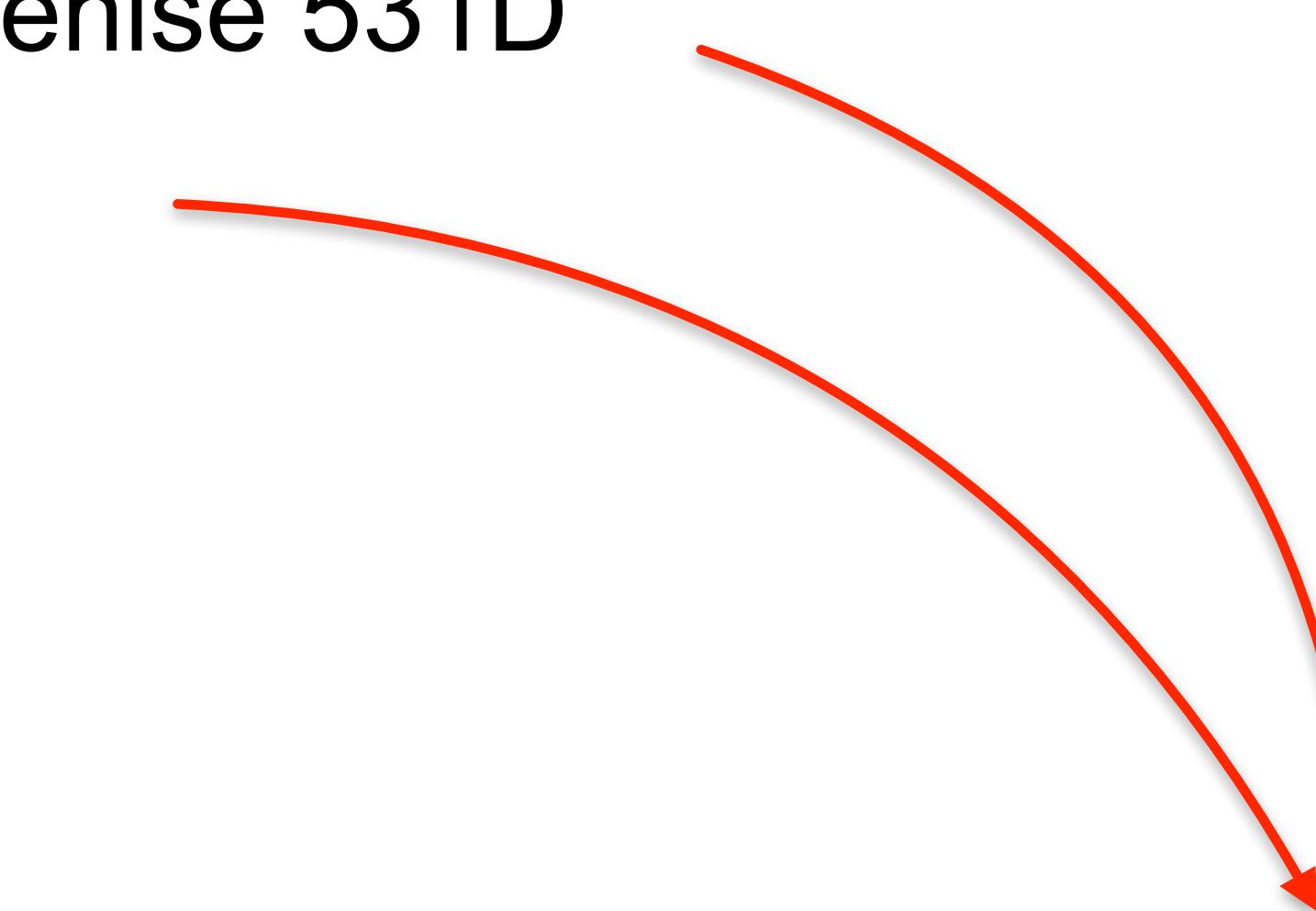
One-stop resource for everything we'll cover today

- Download all slides (PDF)
- Download all exercises (PDF)
- Download all images (ZIP)

# I.A.C. (Image Analysis Collaboratory)

@ HMS since Sep '22

- **Support and collaboration:** for image and data analysis projects
- **Teaching:** at HMS and internationally
- **IT:** access to relevant solutions, open- and closed-source
- **R&D:** centered on methods and tools
- **Locations:** LHRRB 105 & Armenise 531D
- **Consults:** Contact Simon
- **[simon@hms.harvard.edu](mailto:simon@hms.harvard.edu)**



<https://iac.hms.harvard.edu/>

# Who we are, currently



Antoine A. Ruzette

Associate



Ranit Karmakar

Specialist Postdoc



Maria Theiss

Specialist Postdoc



Federico Gasparoli

Research Associate



Simon F. Nørrelykke

Director, Lecturer

## Backgrounds in

Physics

Biology

Microscopy

Bioengineering

Computer Engineering

# Learning objectives

1. **Motivate** the use of algorithms in image analysis
2. **Introduce** some image-analysis nomenclature
3. **Learn** to use Fiji effectively and reproducibly

# Reasons for Learning About Image Processing

- Make pretty pictures (processing)
  - publications, talks, websites, ...
- Get numbers out of pictures (analysis)
  - cell sizes, vessel lengths, GFP expression level, ...
- Make experiment possible (automation)
  - whole-genome screen: millions of images
- Objectivity and Reproducibility
  - in science, just do it!

# Reasons for Not Learning About Image Processing

- ...

**none**

# Acknowledgements

- Szymon Stoma, ETH Zurich, Switzerland
  - slides and workflows
- Peter Bankhead, Edinburgh University, Scotland
  - slides and graphics

# Self-introductions

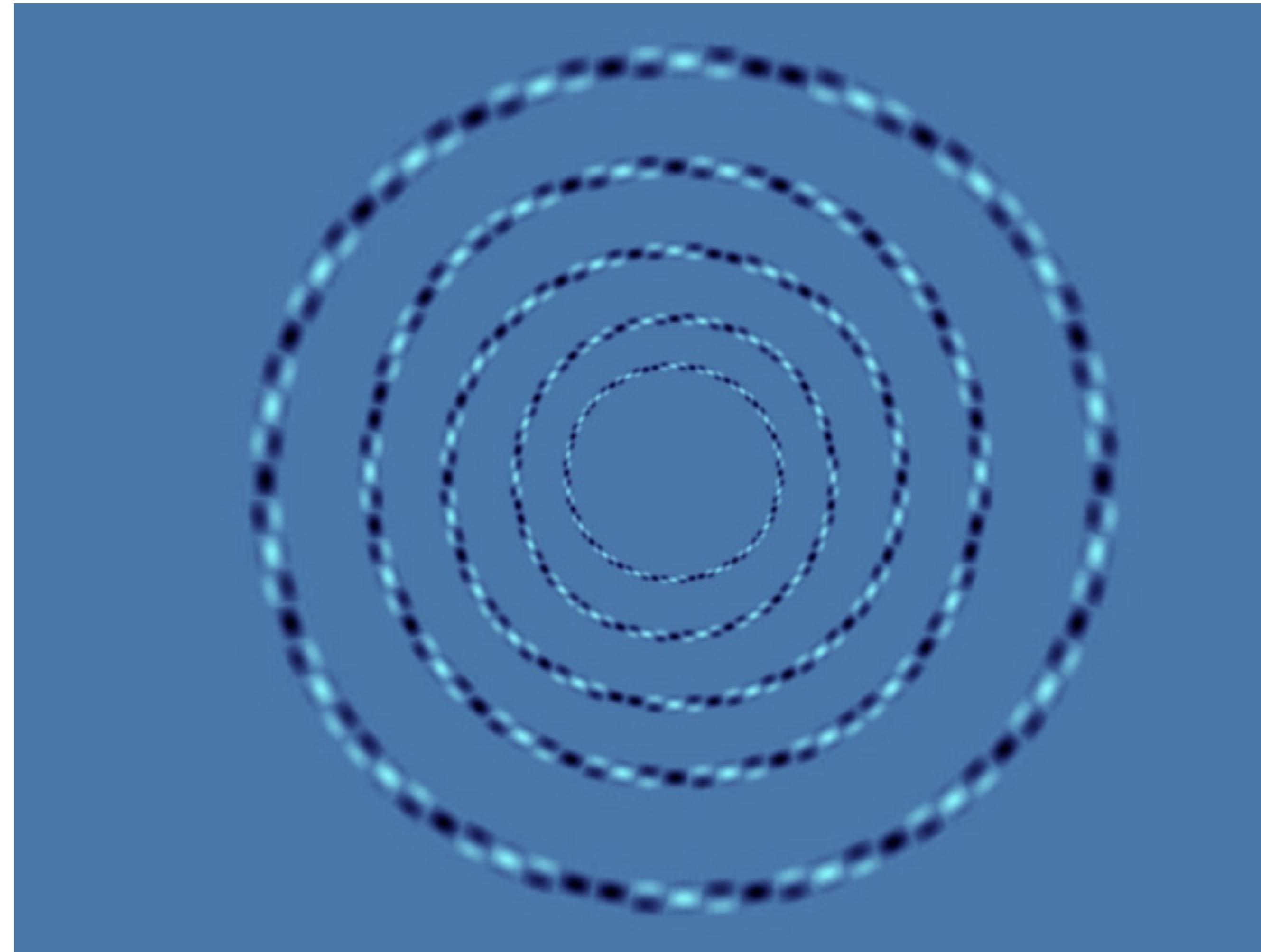
1. My **name** is *Donald Duck*
2. My **position** is as a *10th year postdoc*
3. My **lab** is *Mad City Labs*
4. My **model system** is *blue whales*
5. I have **data** from *confocal microscopy*

# Optical Illusions

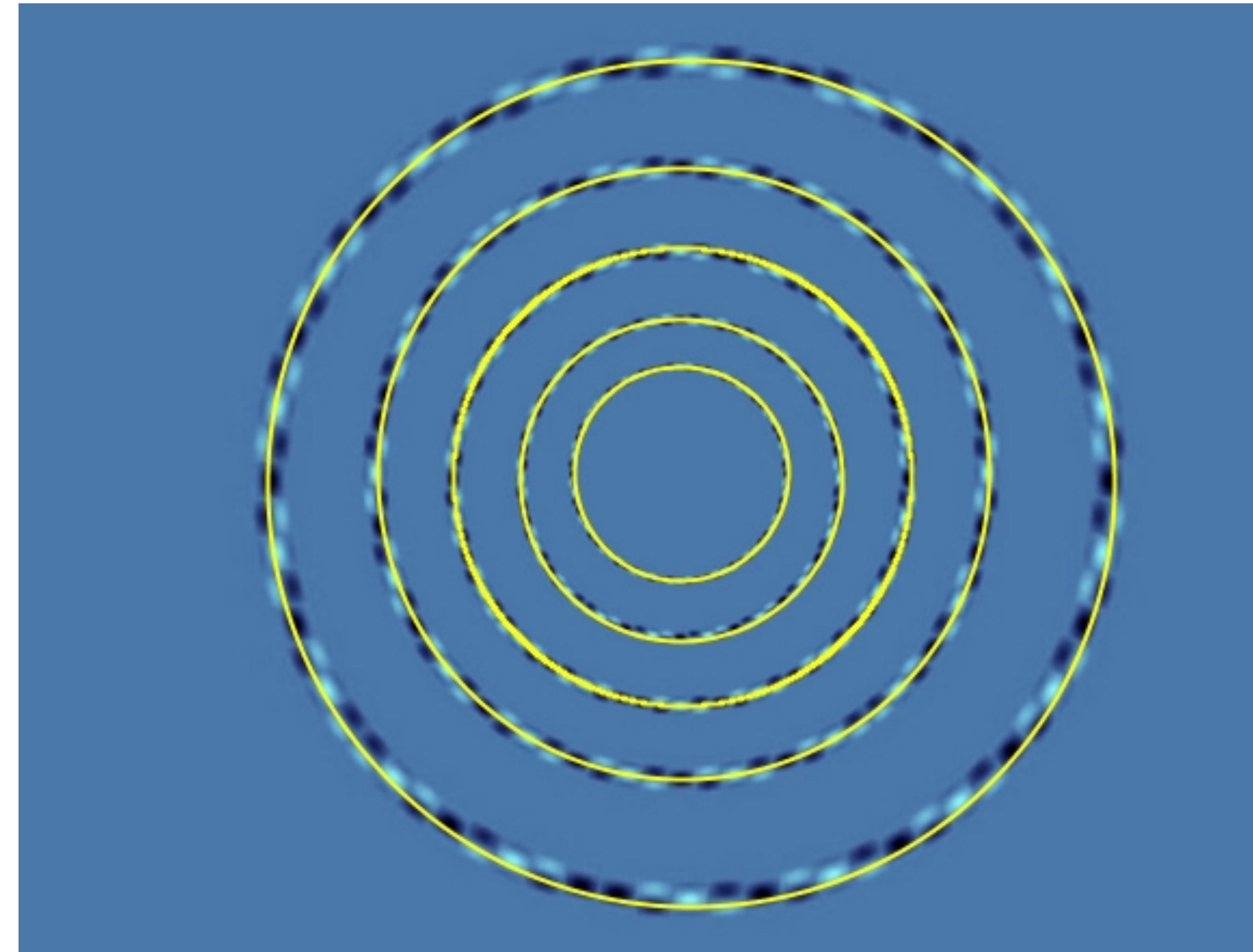
---

Why should you analyze images  
with computers

# Color perception and pattern recognition is individual – science less so

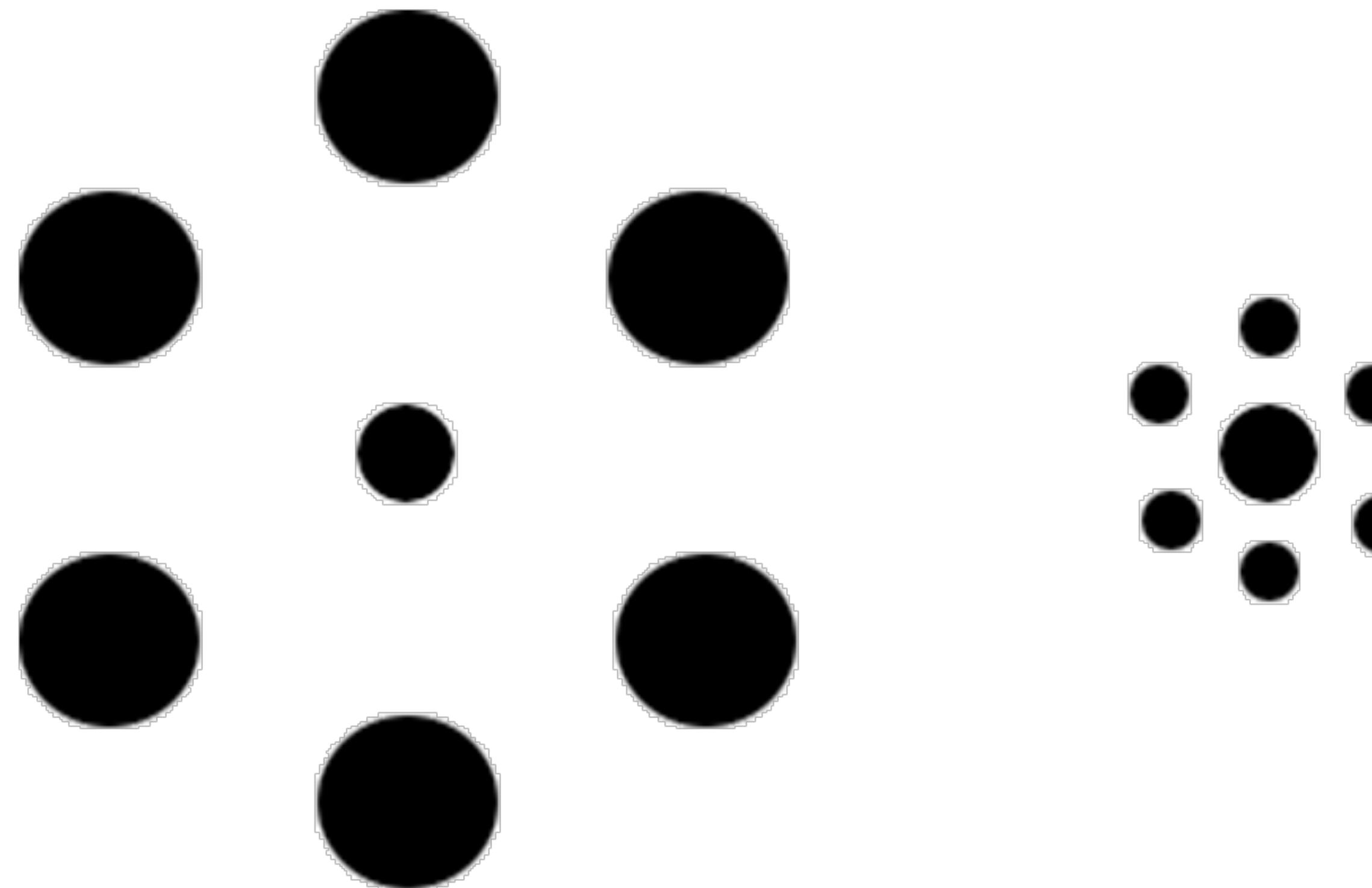


# Concentric circles !?



<https://www.moillusions.com/perfect-circles-optical-illusion/>

# Identical central discs?



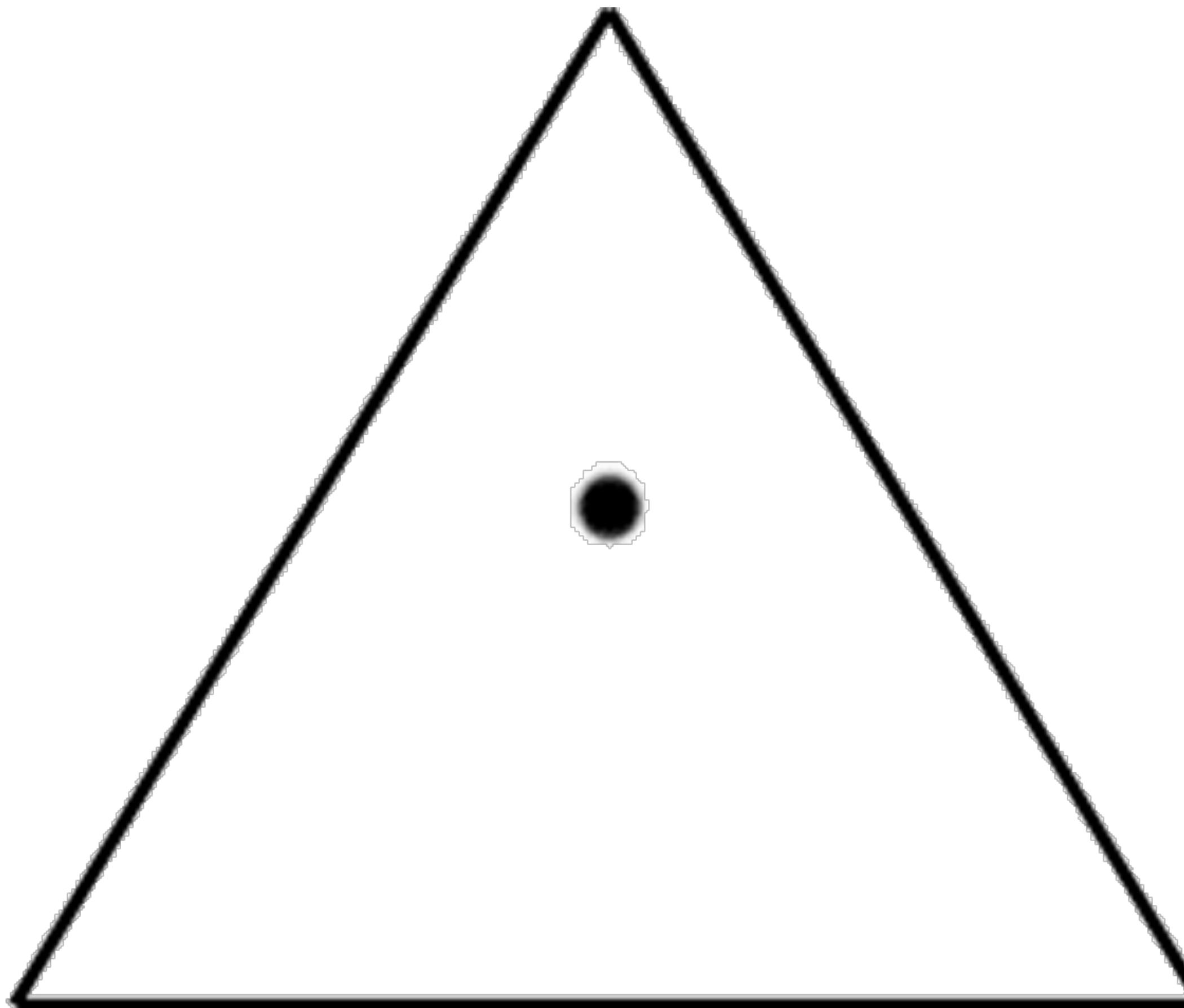
Our size estimate is strongly influenced by the local neighbourhood

# Identical central discs?



Yes, the discs are identical

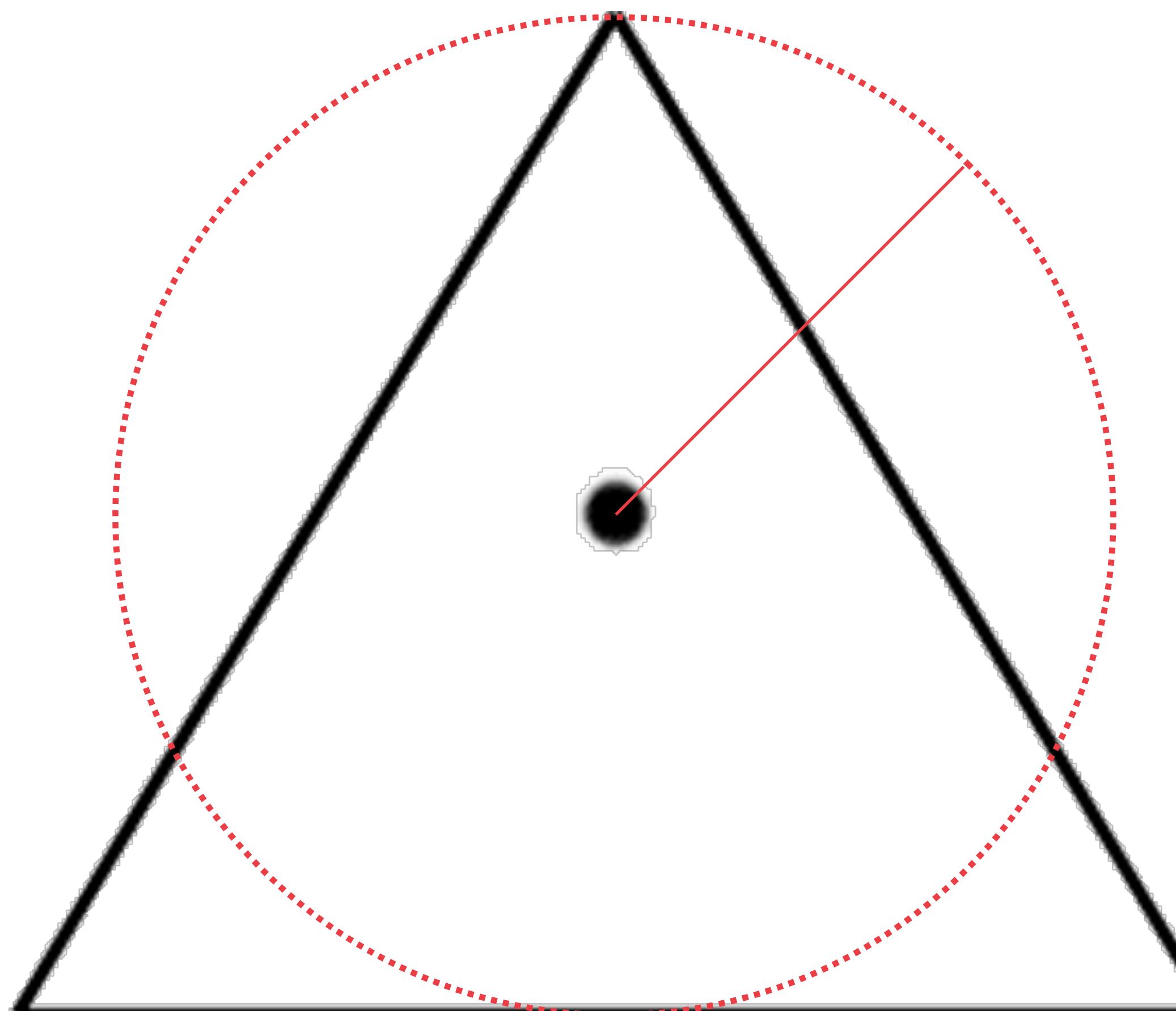
# Is the dot half-way up?



Our sense of distance depends on neighbourhood

<http://www.brainbashers.com>

# Is the dot half-way up?



Yes it is!

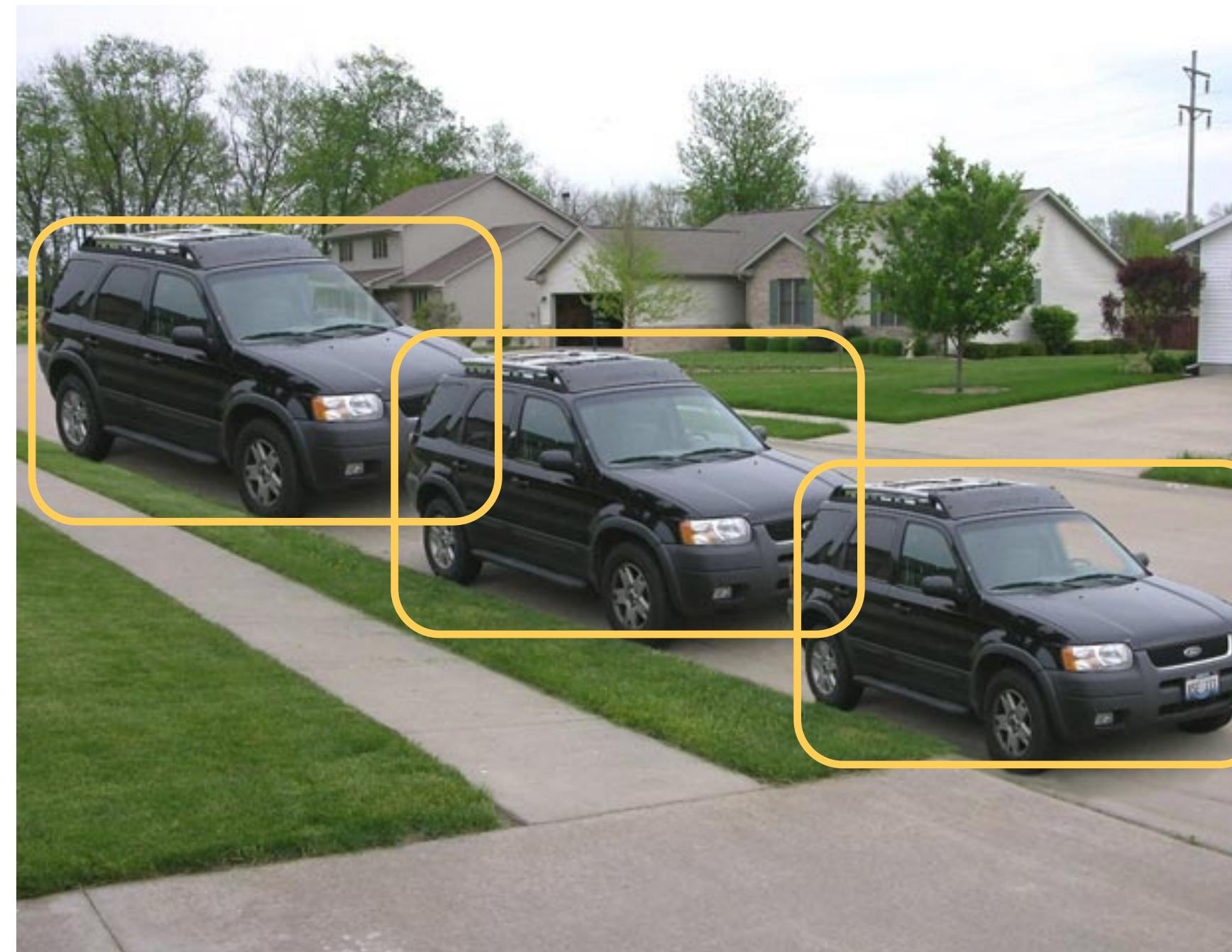
# Which car is bigger?



# Which car is bigger?

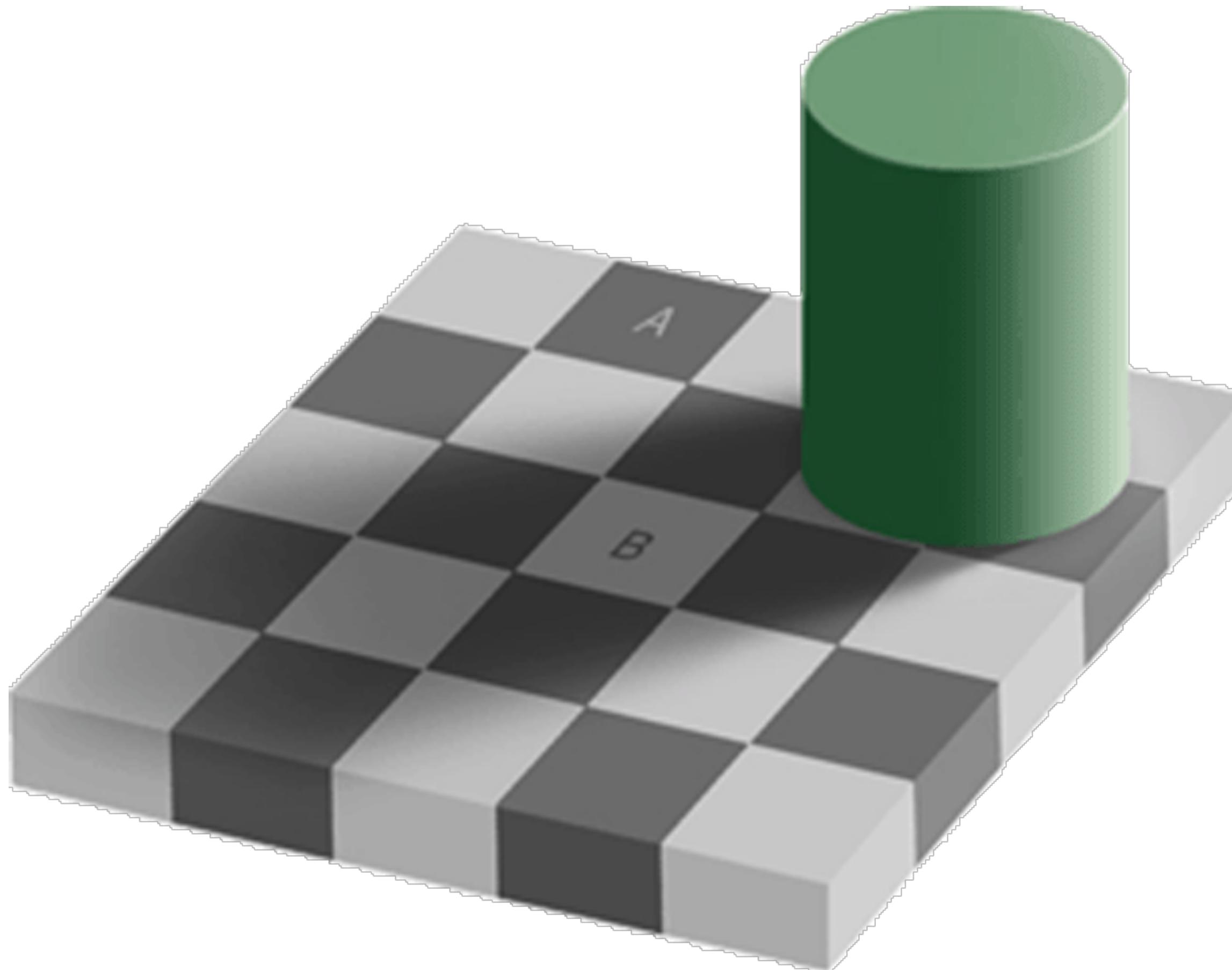


# Which car is bigger?



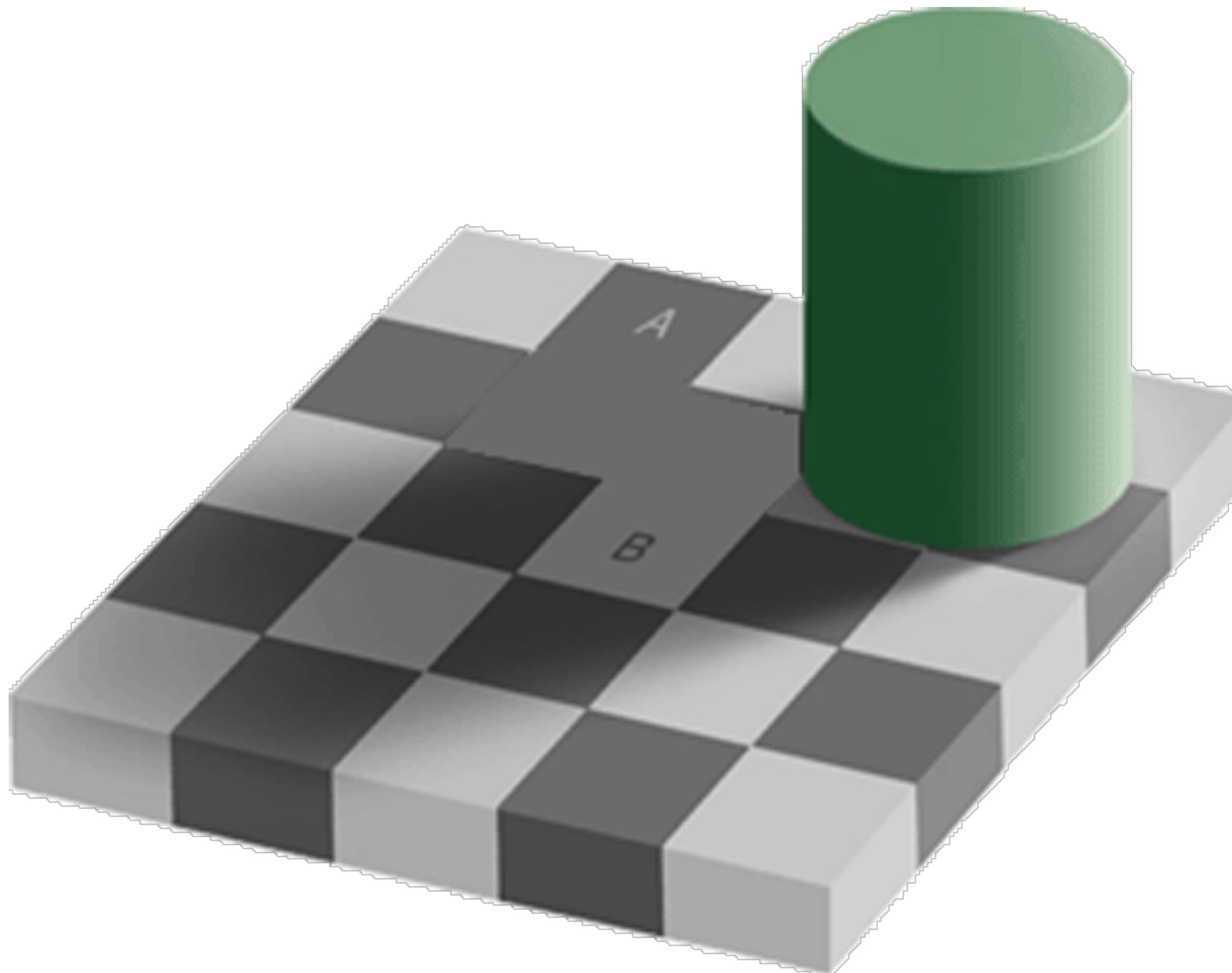
All cars are same size  
(Erroneous, application of subjective/perceptual constancy algorithm by your brain)

# Are A and B equally grey?



Intensity perception depends strongly on neighbourhood

# Are A and B equally grey?



Yes they are!

# Spot the animal

---

Why you should still use your brain  
(a pre-trained neural network)

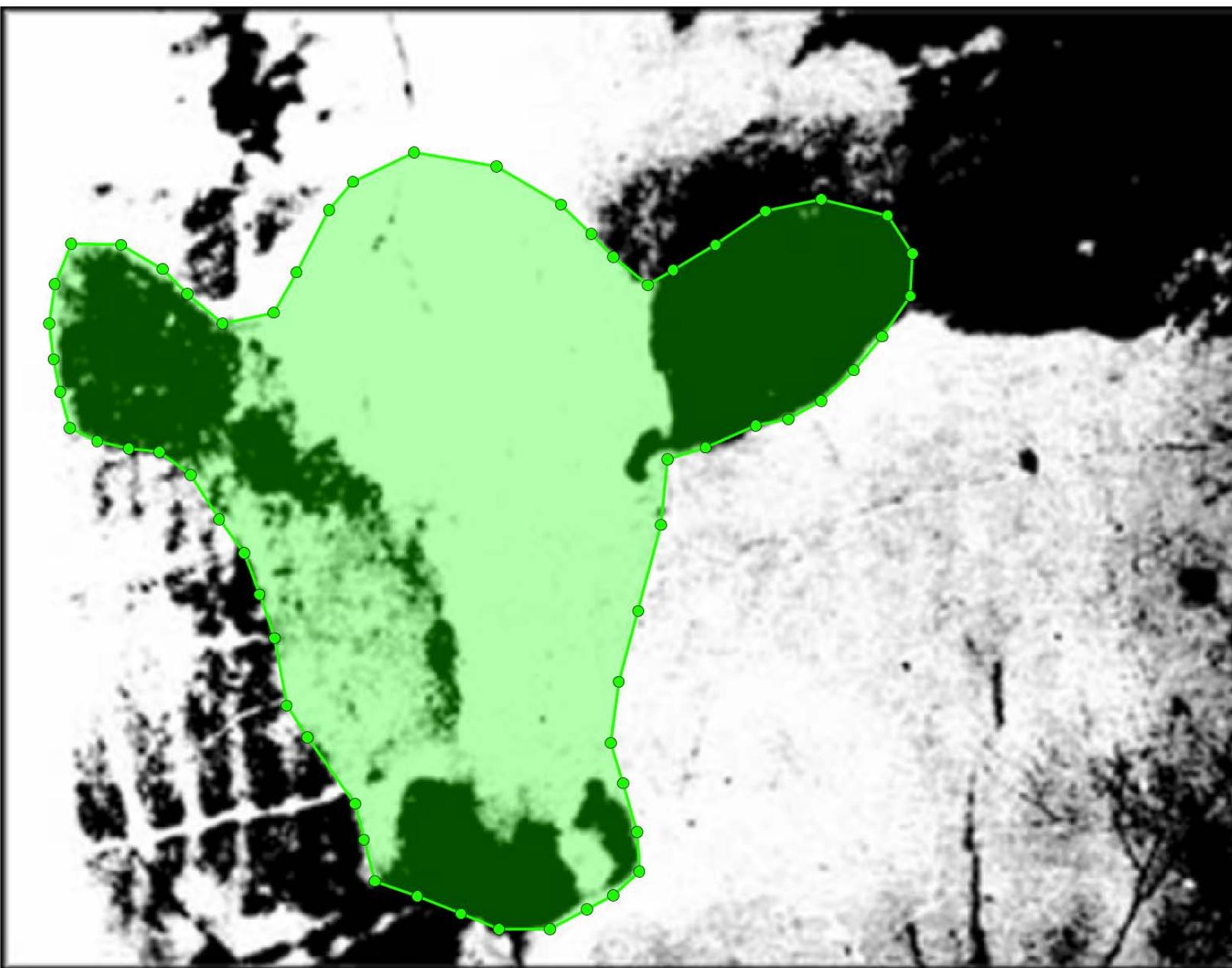
# Pattern Recognition



Yes, that is a cow!

<http://www.brainbashers.com>

# Pattern Recognition



Yes, that is a cow!

<http://www.brainbashers.com>

# Using Prior Knowledge



Keyword: Dalmatian dog!

<http://www.brainbashers.com>

# Using Prior Knowledge



Keyword: Dalmatian dog!

<http://www.brainbashers.com>

# A brief history of image analysis

---

ImageJ centric

# 37 years ago...

**NIH IMAGE**

## What is ImageJ?



[ImageJ](#) is an open source image processing and image analysis [software](#) for multi-dimensional image data with a focus on scientific imaging.

1987 - [NIH Image](#): written in Pascal for the Macintosh by [Wayne Rasband](#)

1997 - [ImageJ](#) 1.x: written in Java and also compatible with Windows PC

Since then, [ImageJ](#) has been maintained and upgraded (e.g. [ImageJ2](#), [Fiji](#), ...)

[ImageJ GitHub](#)

[Fiji GitHub](#)





# What is Fiji?



Fiji stands for Fiji Is Just ImageJ

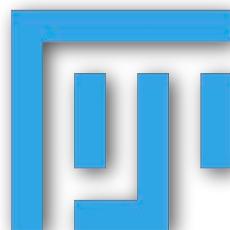
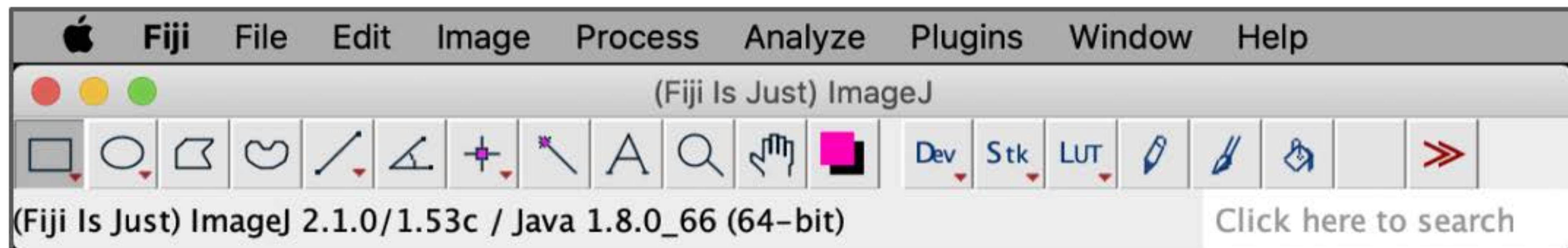
It is a “batteries included” distribution of [ImageJ2](#)

Easy to install (“portable” application)

Include a collection of Plugins

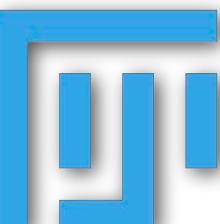
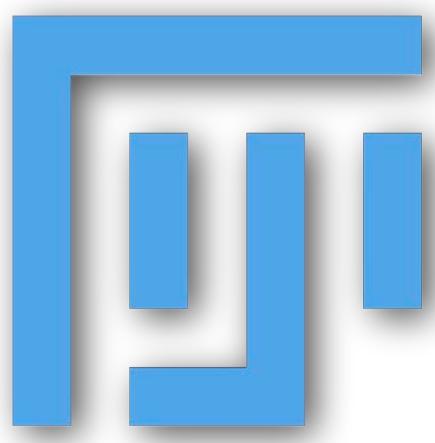
Easy to update (<https://imagej.net/list-of-update-sites/>)

Documentation



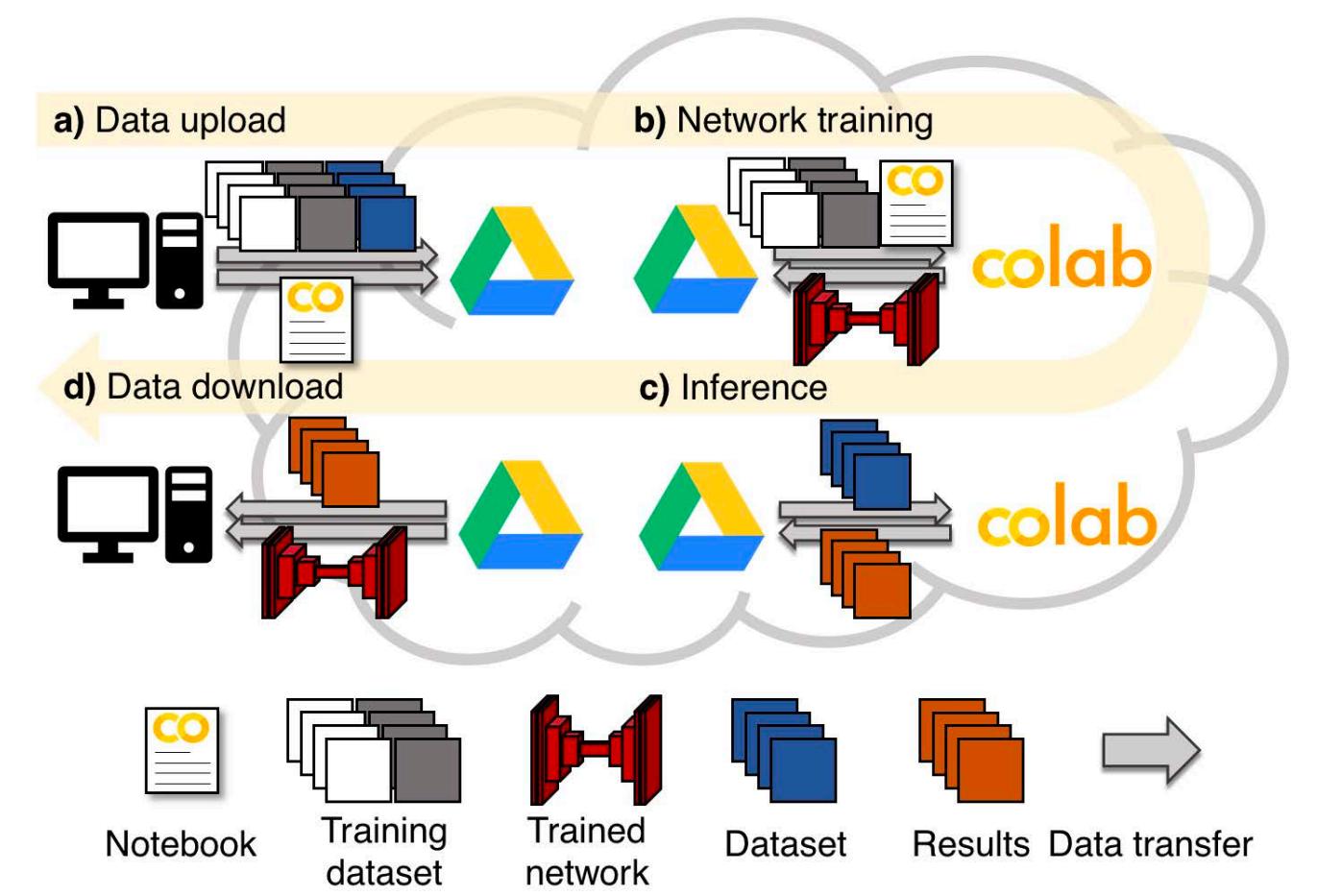
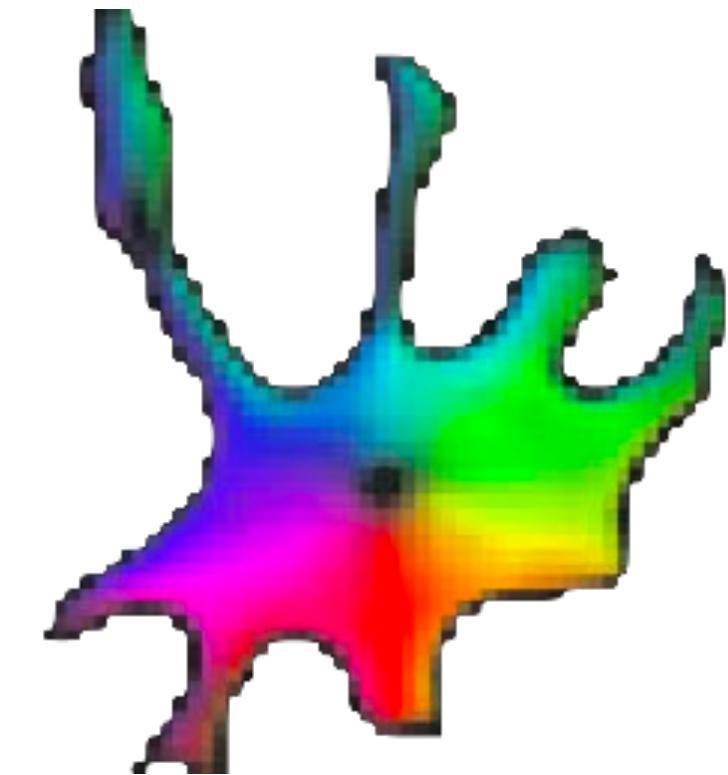
# 12 years ago...

- AlexNet is published (2012)
  - Wins the *ImageNet Large Scale Visual Recognition Challenge*
  - Error of 15.3%, more than 10.8 percentage points better than #2
  - Possible due to GPUs
- CellProfiler has been around for 6 years (2006)
- Fiji (Fiji is just ImageJ) has been around for 5 years (2007)
- scikit-image released 3 years ago (2009)
- Bitcoin grows from \$5 to \$13 (remember Silk Road?) (2012)
- U-Net, GANs, and Jupyter will appear in 2-3 years (2014/15)
- AlphaGo will beat Lee Sodol in 4 years (2016)
- QuPath is still 4 years in the future (2016)



# 4 years ago...

- CellPose is out (2020)
  - “Cellpose: a generalist algorithm for cellular segmentation”
  - Trained on highly varied images of cells, over 70,000 segmented objects
  - Cells don’t have to be star-shaped
  - Web-platform and Jupyter notebooks
  - <https://www.nature.com/articles/s41592-020-01018-x>
- ZeroCostDL4Mic available
  - [Paper](#) still a year in the future
  - Implementation of common DL technologies to microscopy imaging
  - Relies on GPUs and other infrastructure provided by Google Colab.
  - <https://github.com/HenriquesLab/ZeroCostDL4Mic/wiki>
- *Vision Transformers* (2020)
  - “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”, Dosovitskiy et al.
  - <https://arxiv.org/abs/2010.11929>

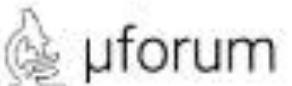


# Today...

# Image Analysis Software

All Topics	AGAVE	AICSImageIO	Arkitekt	Ay	BAND
BIAFLows	BiaPy	BIII	BiofilmQ	Bio-Formats	BioImageIO
BoneJ	BrainGlobe	Cellpose	CellProfiler	CLIJ	CytoMAP
Cytomine	DAIM	DeepLabCut	Fiji	FLIMLib	GerBI
Icy	IDR	ilastik	ImageJ	ImageJ2	ImgLib2
ImJoy	ImSwitch	iRODS	JIPipe	JuliaImages	Mars
MCMICRO	MIA	MIB	μManager	MoBIE	ModularImageAnalysis
MorphoGraphX	MorphoNet	napari	NEUBIAS	NFDI4BIOIMAGE	OME
OMERO	OmeSliCC	OpenIRIS	OpenSPIM	Orbit	Piximi
PolusAI	PYME	Python-Microscope	QUAREP-LiMi	QuPath	scenery
SCIFIO	scikit-image	sciview	SmartMicroscopy	SR-Tesseler	StarDist
TeamTomo	TissUUmaps	vedo			
... Your Icon Here					

**RELATED COMMUNITIES**

**Citation Details**

 Article  
**Fiji: an open-source platform for biological-image analysis**  
Nature Methods, 2012, Volume 9, Number 7, Page 676  
Johannes Schindelin, Ignacio Arganda-Carreras, Erwin Frise, Verena Kaynig, Mark Longair, Tobias Pietzsch, Stephan Preibisch, Curtis Rueden, Stephan Saalfeld, Benjamin Schmid, Jean-Yves Tinevez, Daniel James White, Volker Hartenstein, Kevin Eliceiri, Pavel Tomancak, Albert Cardona

4680 ITEMS CITE THIS ARTICLE WITH YEAR: 2022

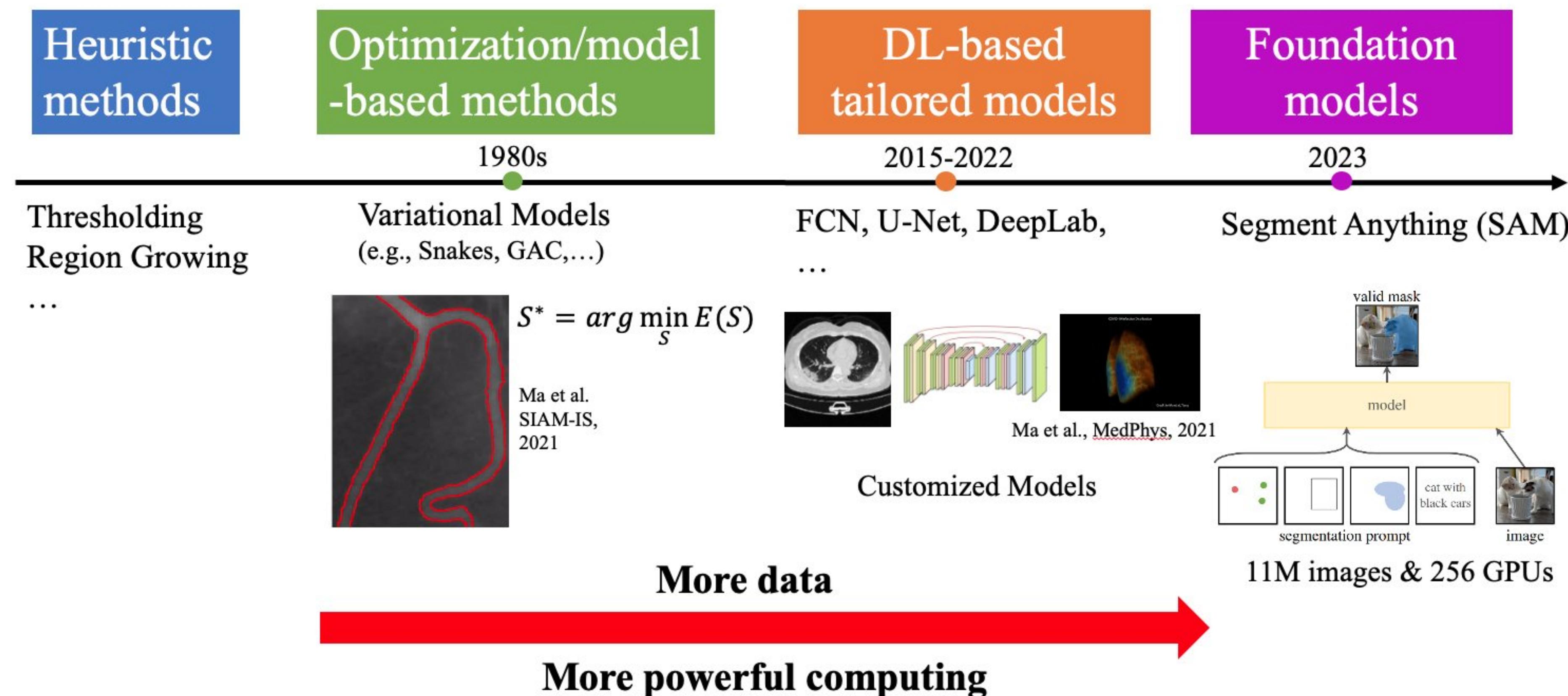
**Crossref** 30162 ITEMS CITE THIS ARTICLE

**WEB OF SCIENCE™** 33431 ITEMS CITE THIS ARTICLE

**Crossref** 39921 ITEMS CITE THIS ARTICLE

# History of Segmentation

## Segmentation Paradigm Over the Past Half Century



# English as the next programming language

---

LLMs, chatbots and foundation  
models for bio image analysis



GPT-3.5

GPT-4

# ChatGPT

**Create a workout plan**  
for resistance training

**Design a database schema**  
for an online merch store

**Explain why popcorn pops**  
to a kid who loves watching it in the microwave

**Write an email**  
to request a quote from local plumbers

Please write an ImageJ macro that uses StarDist to segment an image from fluorescence microscopy



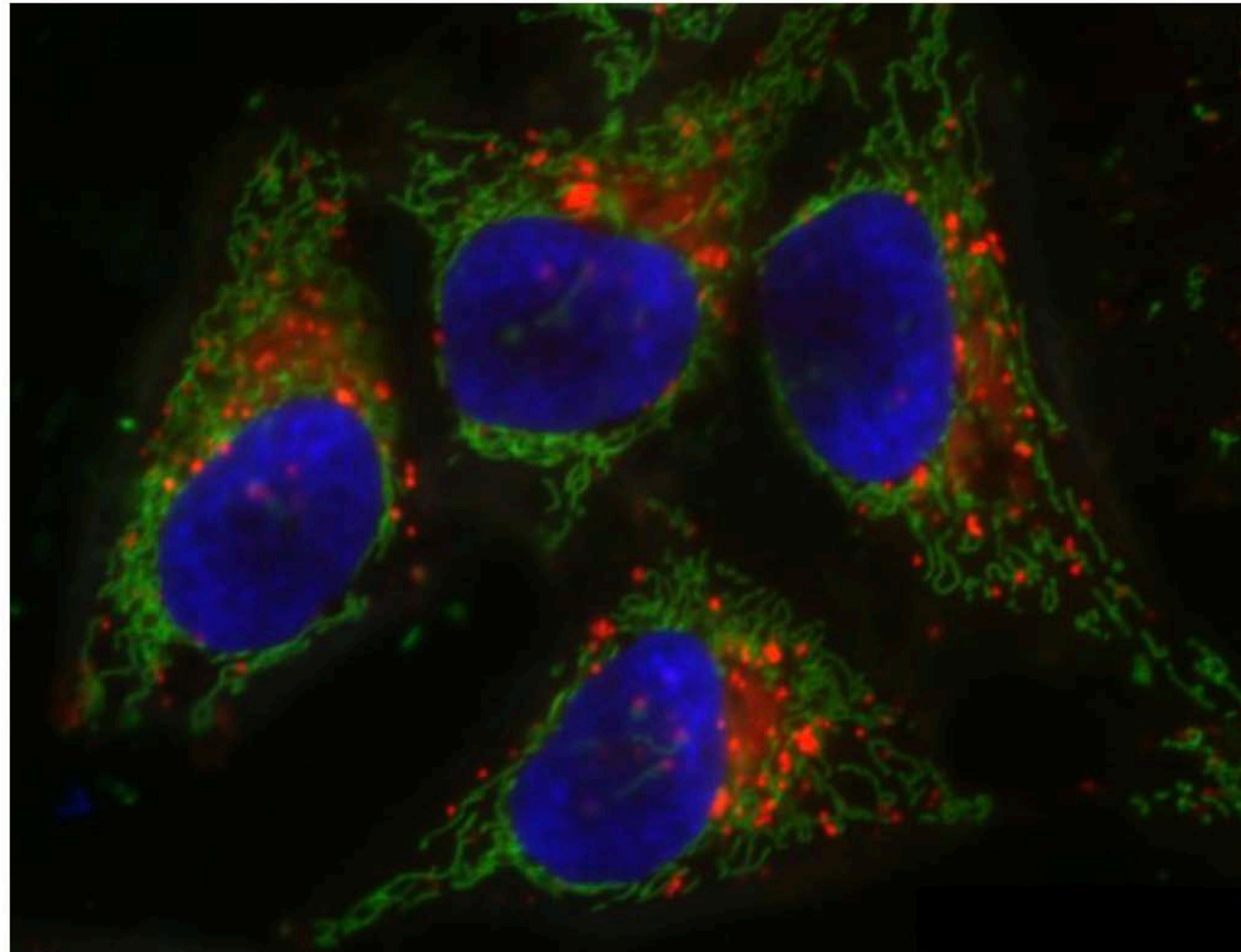
?

Free Research Preview. ChatGPT may produce inaccurate information about people, places, or facts. [ChatGPT September 25 Version](#)

# Using ChatGPT for Image Analysis

SI

You

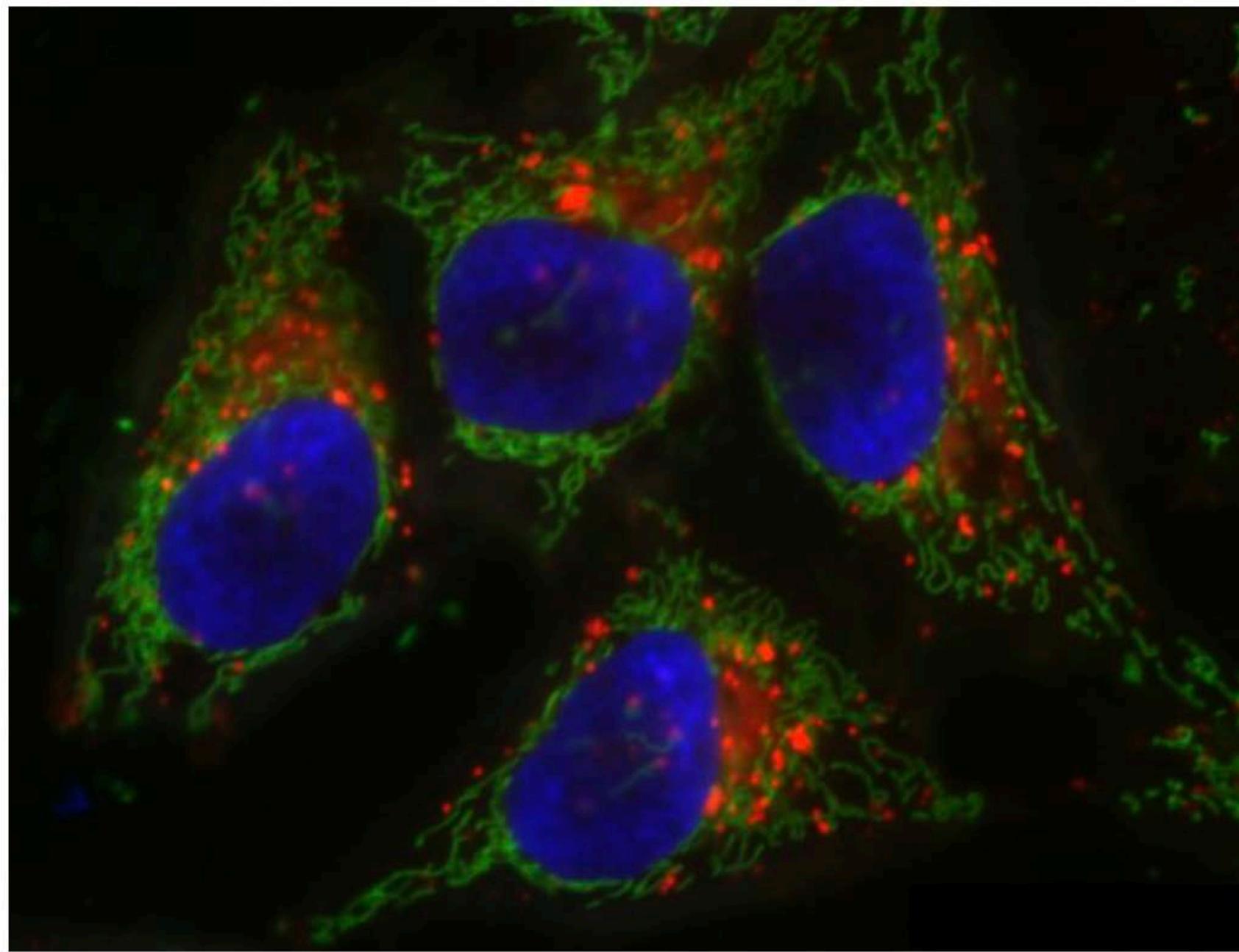


please segment the nuclei and show me the relevant code you used

# Using ChatGPT for Image Analysis

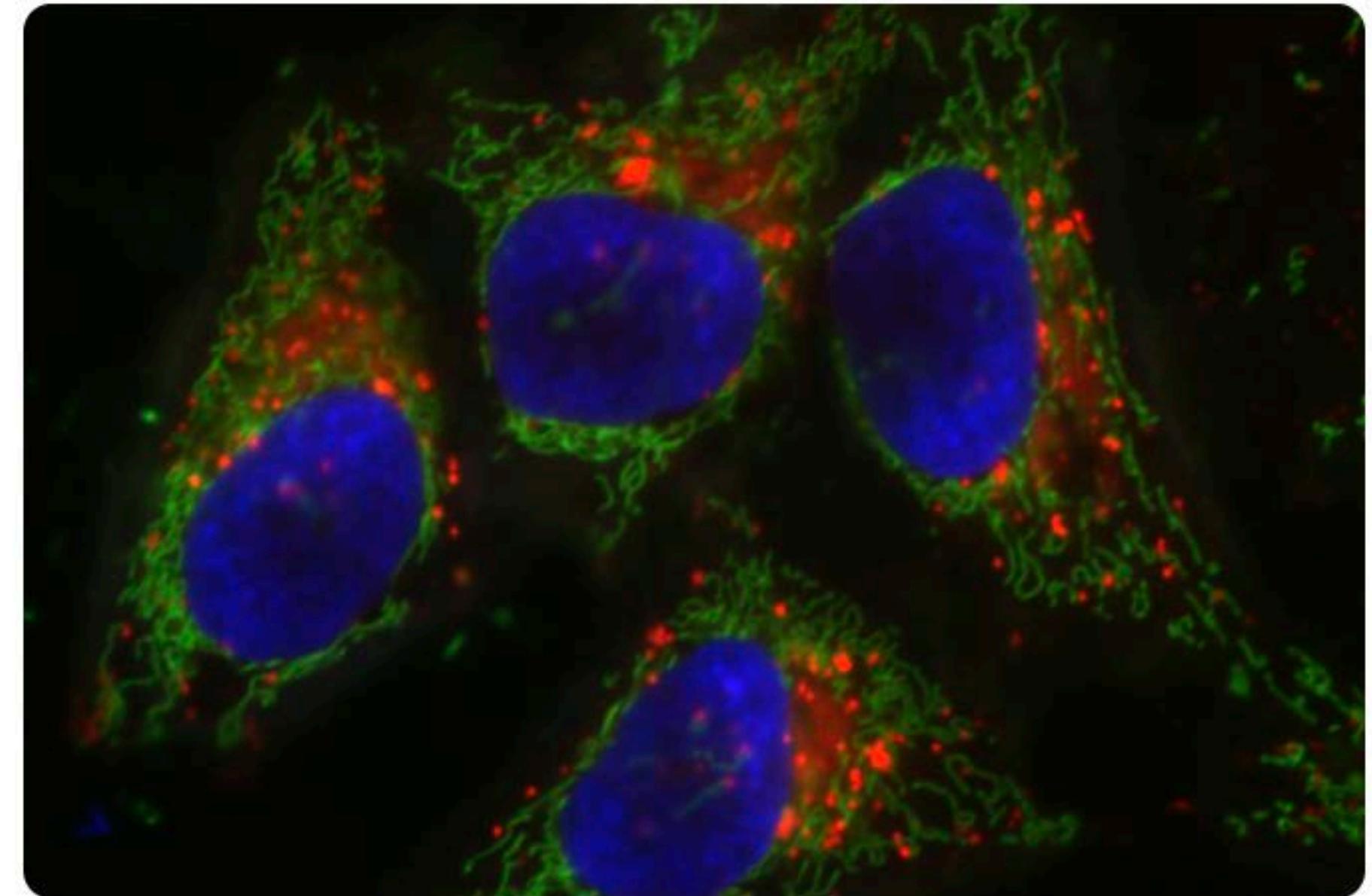
GPT-4, May 2024

You



please segment the nuclei and show me the relevant code you used

GPT-4o, Oct. 2024



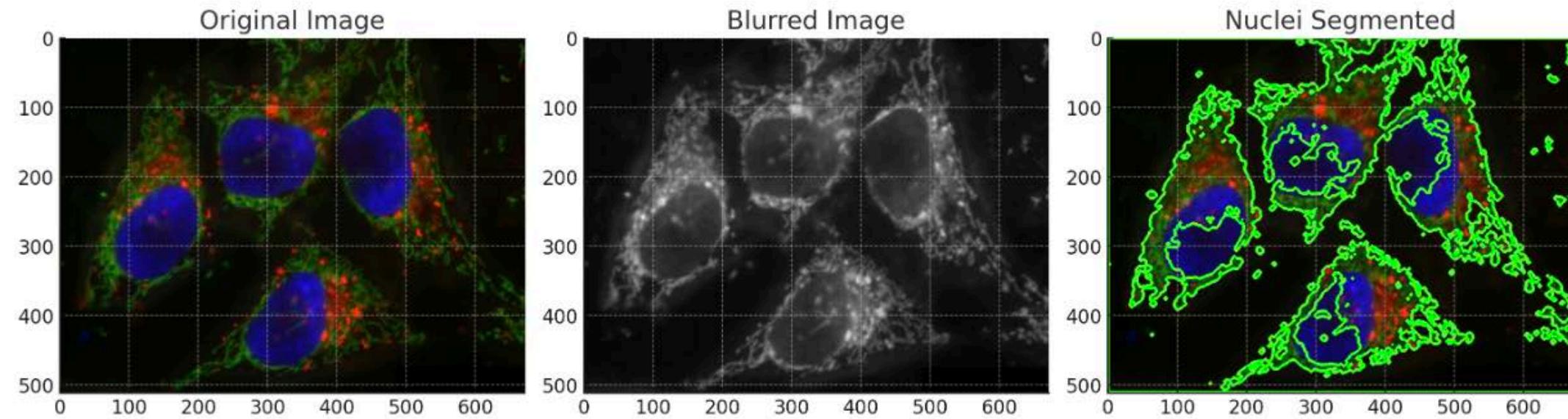
please segment the nuclei and show me the relevant code you used

# Using ChatGPT for Image Analysis

GPT-4, May 2024



ChatGPT

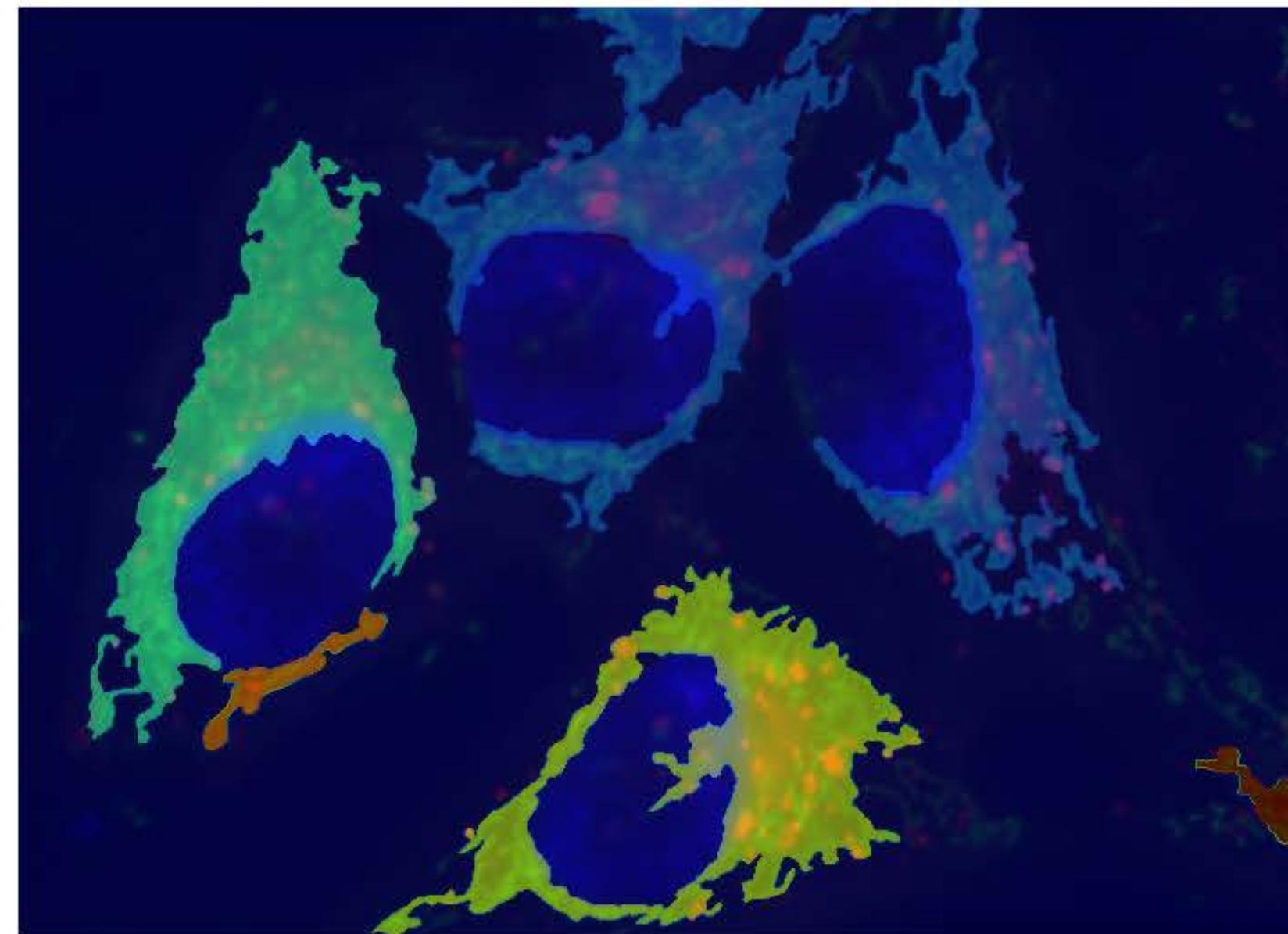


The nuclei in the image have been segmented using the following steps:

- The image was converted to grayscale to simplify the analysis.
- Gaussian Blur was applied to reduce noise and detail in the image, which aids in thresholding and contour detection.
- Otsu's thresholding was performed to convert the grayscale image into a binary image, which helps in distinguishing the nuclei from the background.
- Contours were found in the binary image, which correspond to the boundaries of the nuclei.
- These contours were then drawn onto the original image to visually segment the nuclei.

GPT-4o, Oct. 2024

Segmented Nuclei



# Using ChatGPT for Image Analysis

GPT-4, May 2024

```
python
Copy code

import cv2
import numpy as np
from matplotlib import pyplot as plt

# Load the image
image_path = '/mnt/data/hela-cells.jpg'
image = cv2.imread(image_path)

# Convert the image to gray scale
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

# Apply GaussianBlur, which is good for edge detection
blurred = cv2.GaussianBlur(gray, (5, 5), 0)

# Perform Otsu's thresholding
_, thresh = cv2.threshold(blurred, 0, 255, cv2.THRESH_BINARY_INV + cv2.THRESH_OTSU)

# Find contours which will give us the nuclei boundaries
contours, _ = cv2.findContours(thresh, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)

# Draw the contours on the original image
contoured_image = image.copy()
cv2.drawContours(contoured_image, contours, -1, (0, 255, 0), 3)

# Save the output image with segmented nuclei
output_path = '/mnt/data/hela-cells-segmented.jpg'
cv2.imwrite(output_path, contoured_image)
```

GPT-4o, Oct. 2024

```
python
Copy code

from skimage import io, filters, measure, morphology, color
import matplotlib.pyplot as plt
import numpy as np

# Load the image
image_path = '/mnt/data/hela-cells.png'
image = io.imread(image_path)

# Convert to grayscale
gray_image = color.rgb2gray(image)

# Apply Gaussian filter to smooth the image
smoothed = filters.gaussian(gray_image, sigma=1)

# Use Otsu's threshold to create a binary mask of the nuclei
threshold_value = filters.threshold_otsu(smoothed)
binary_mask = smoothed > threshold_value

# Remove small objects and fill holes to improve segmentation
cleaned_mask = morphology.remove_small_objects(binary_mask, min_size=500)
cleaned_mask = morphology.remove_small_holes(cleaned_mask, area_threshold=500)

# Label the segmented regions
labeled_nuclei = measure.label(cleaned_mask)
```

# Using ChatGPT for Image Analysis

GPT-4, May 2024



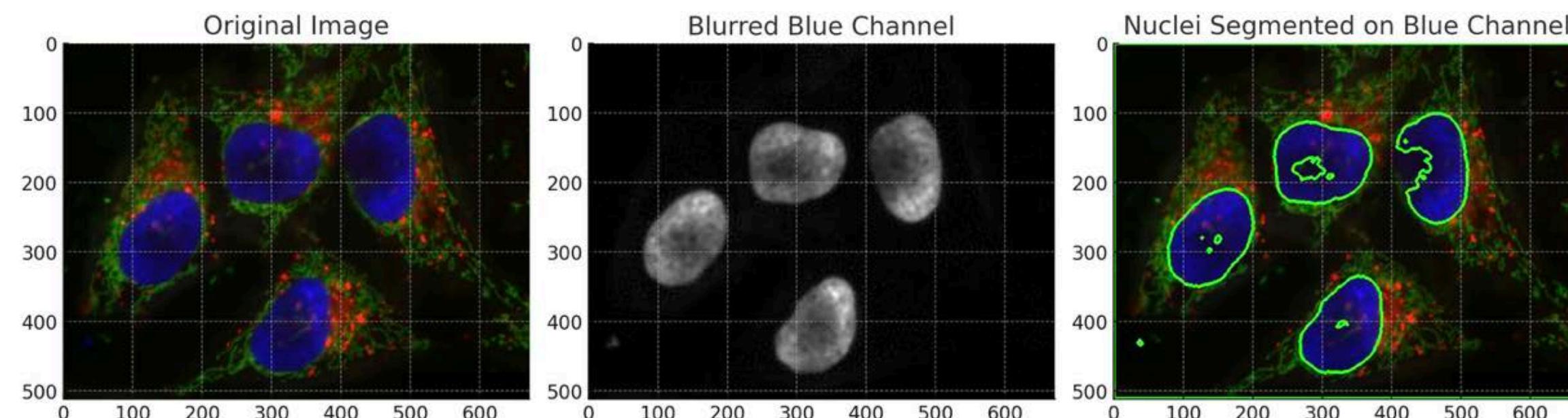
SI

You

please use the blue channel for segmenting the nuclei



ChatGPT

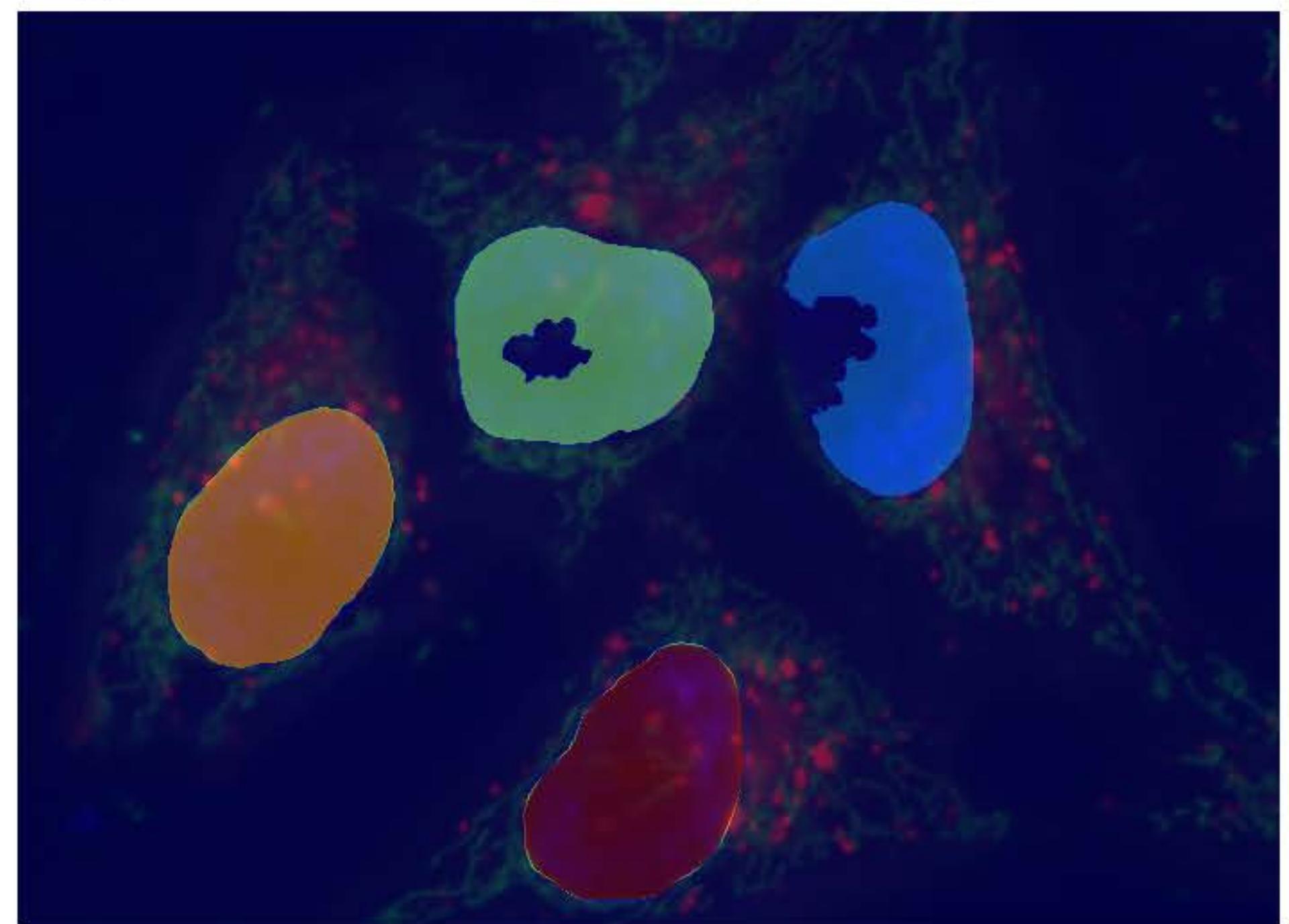


The nuclei segmentation has been performed using the blue channel of the image with the following steps:

- The blue channel was extracted from the original image since the nuclei appear to be stained blue.
- Gaussian Blur was applied to the blue channel to smooth the image, which helps in reducing noise for thresholding.
- Otsu's thresholding was then used on the blurred blue channel to create a binary image where the nuclei are distinguished from the background.

GPT-4o, Oct. 2024

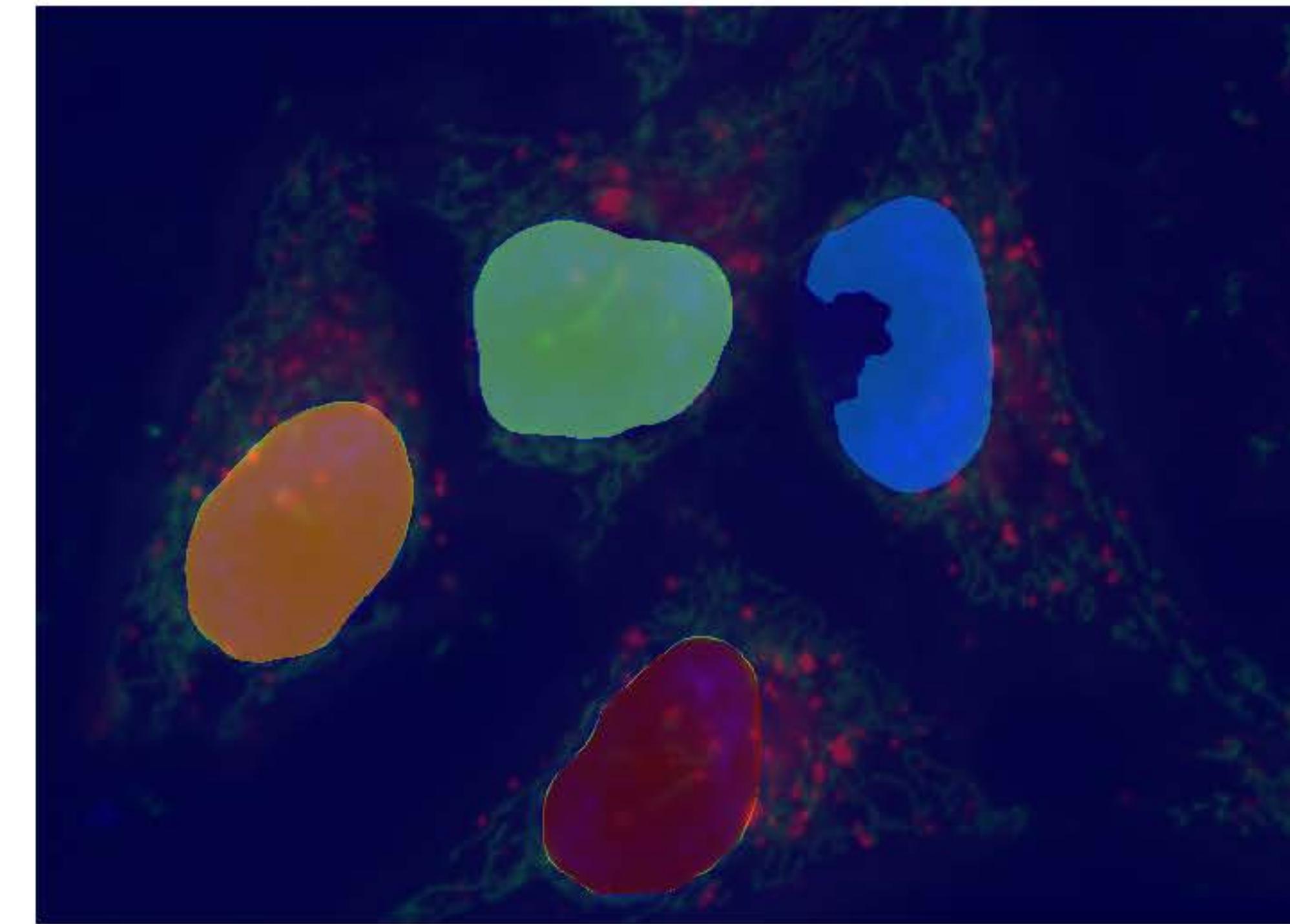
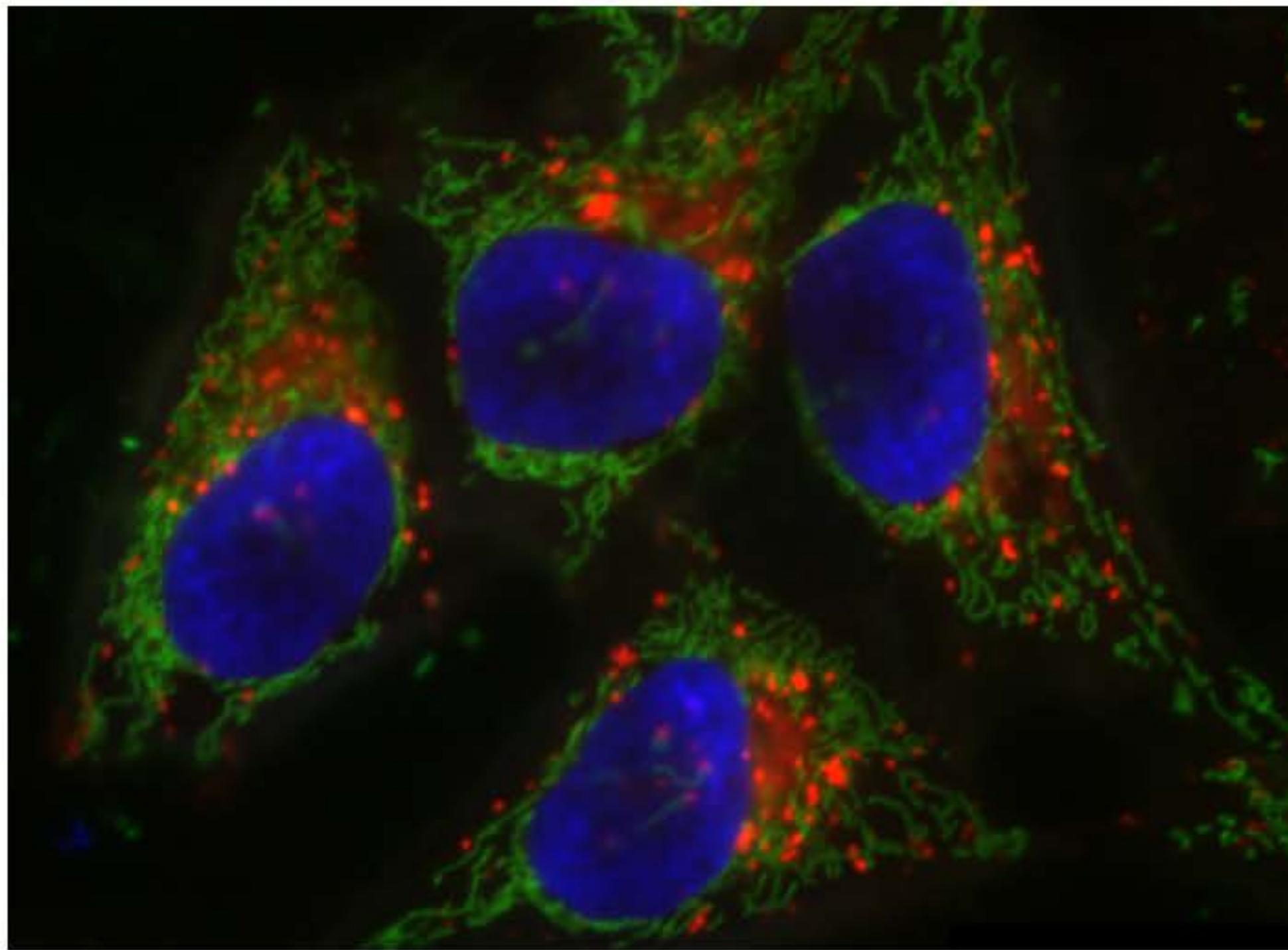
Segmented Nuclei (Blue Channel)



# Using ChatGPT-4o for Image Analysis

GPT-4o, Oct. 2024 - prompted to close larger holes

make sure to close larger holes, and increase smoothing sigma to 3

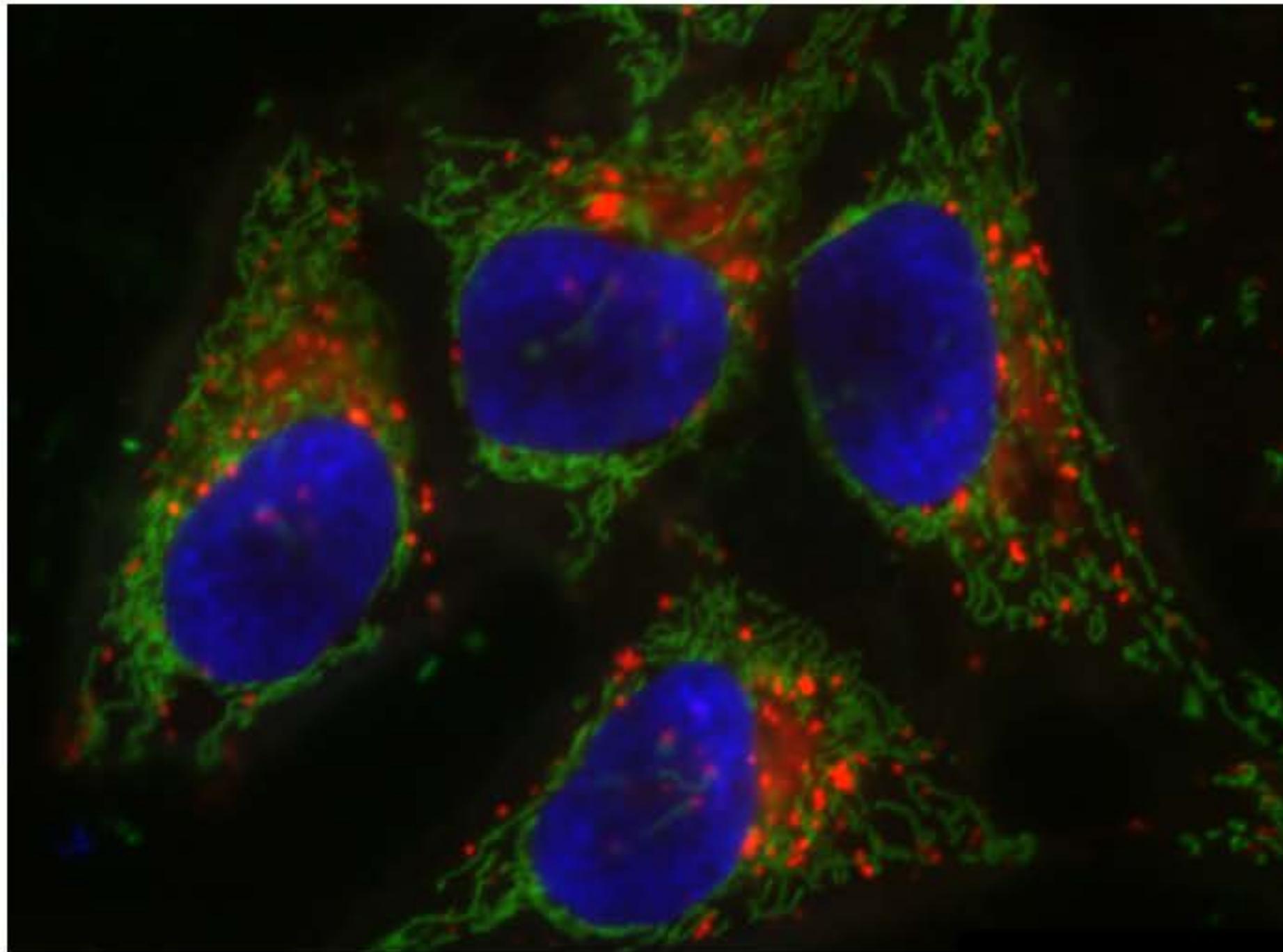


# Using ChatGPT-4o for Image Analysis

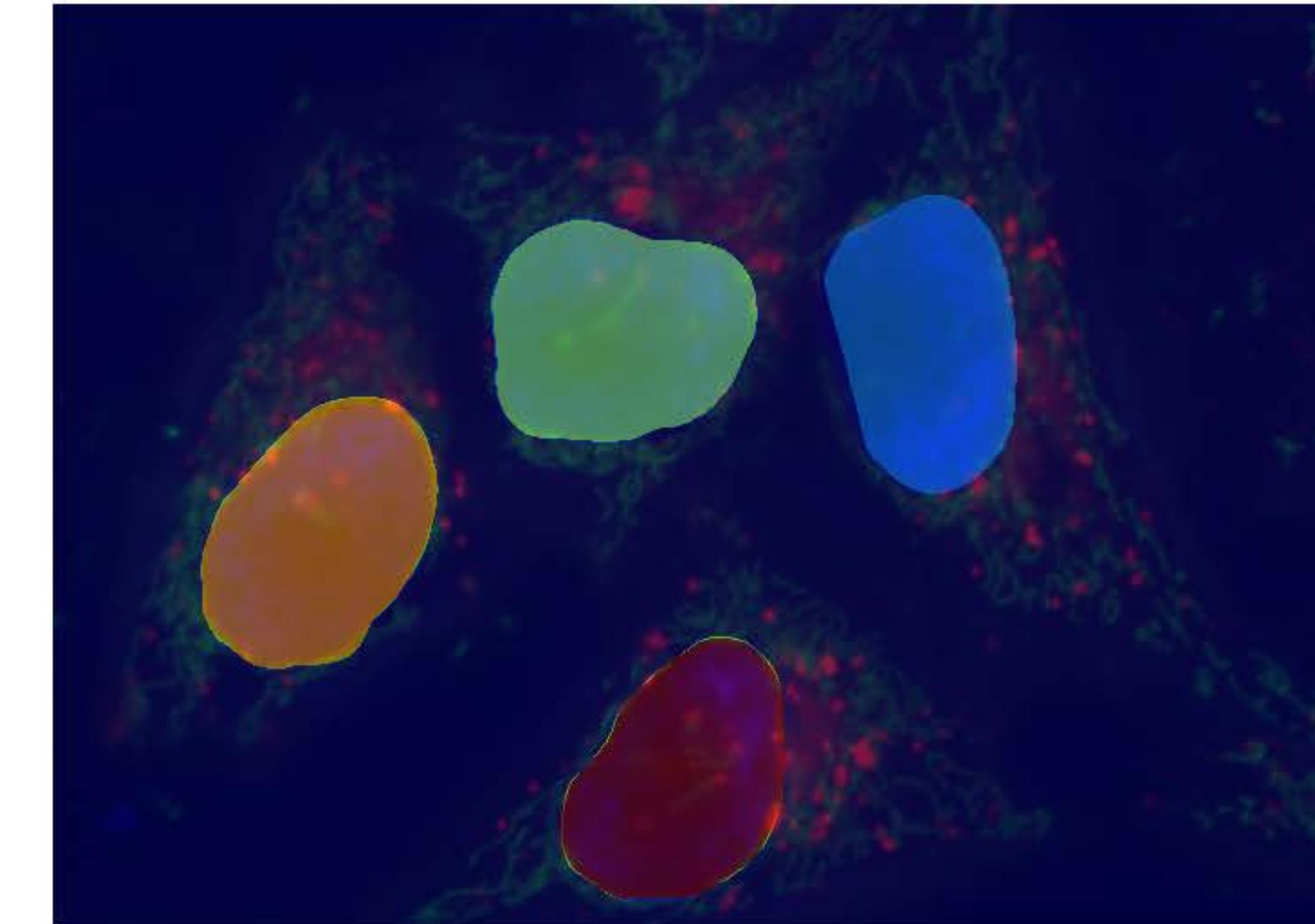
GPT-4o, Oct. 2024 - it helps to know what to ask

compute the convex hull for shapes with a low circularity, to  
approximate the characteristic shape of a nuclei

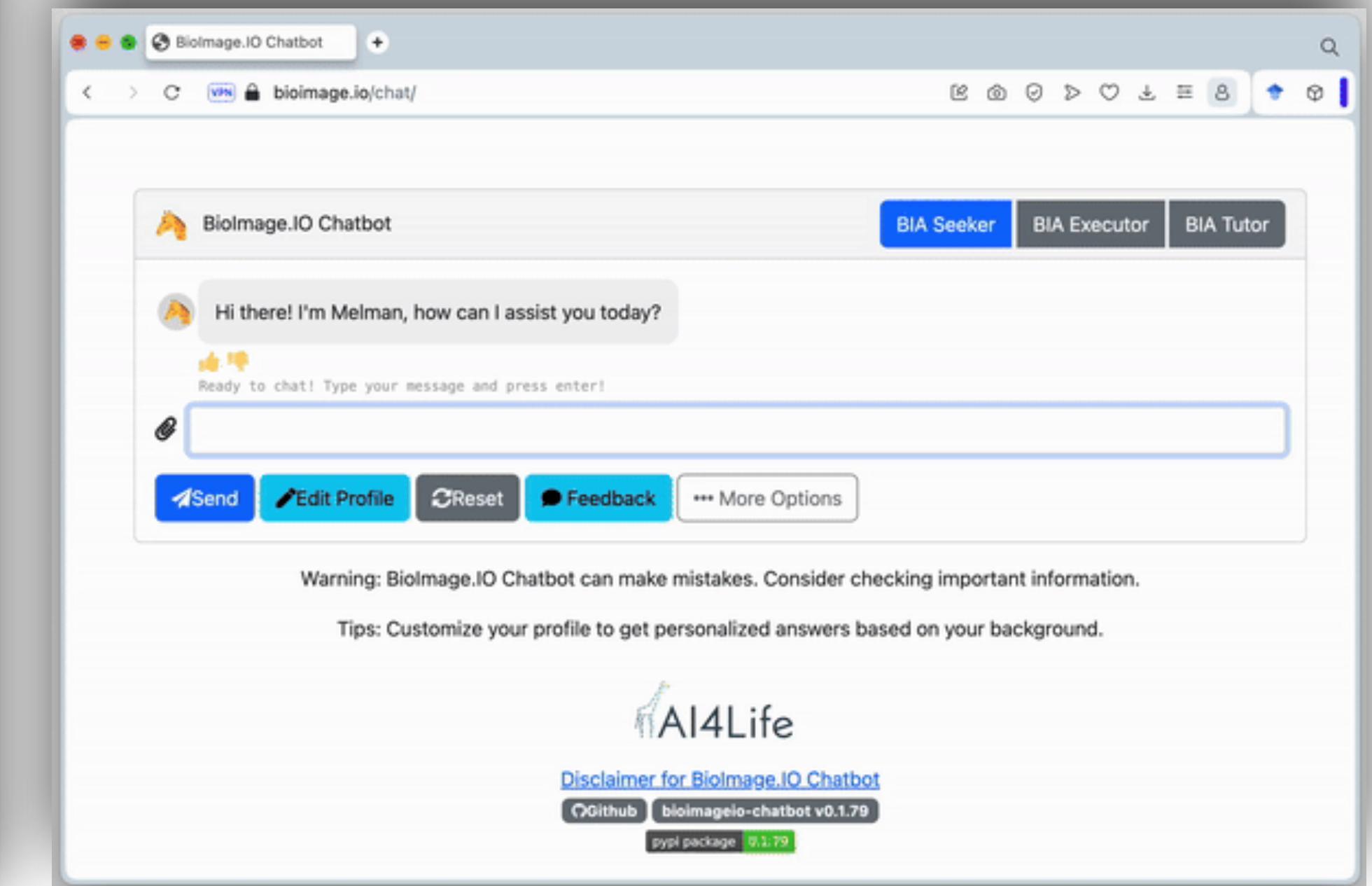
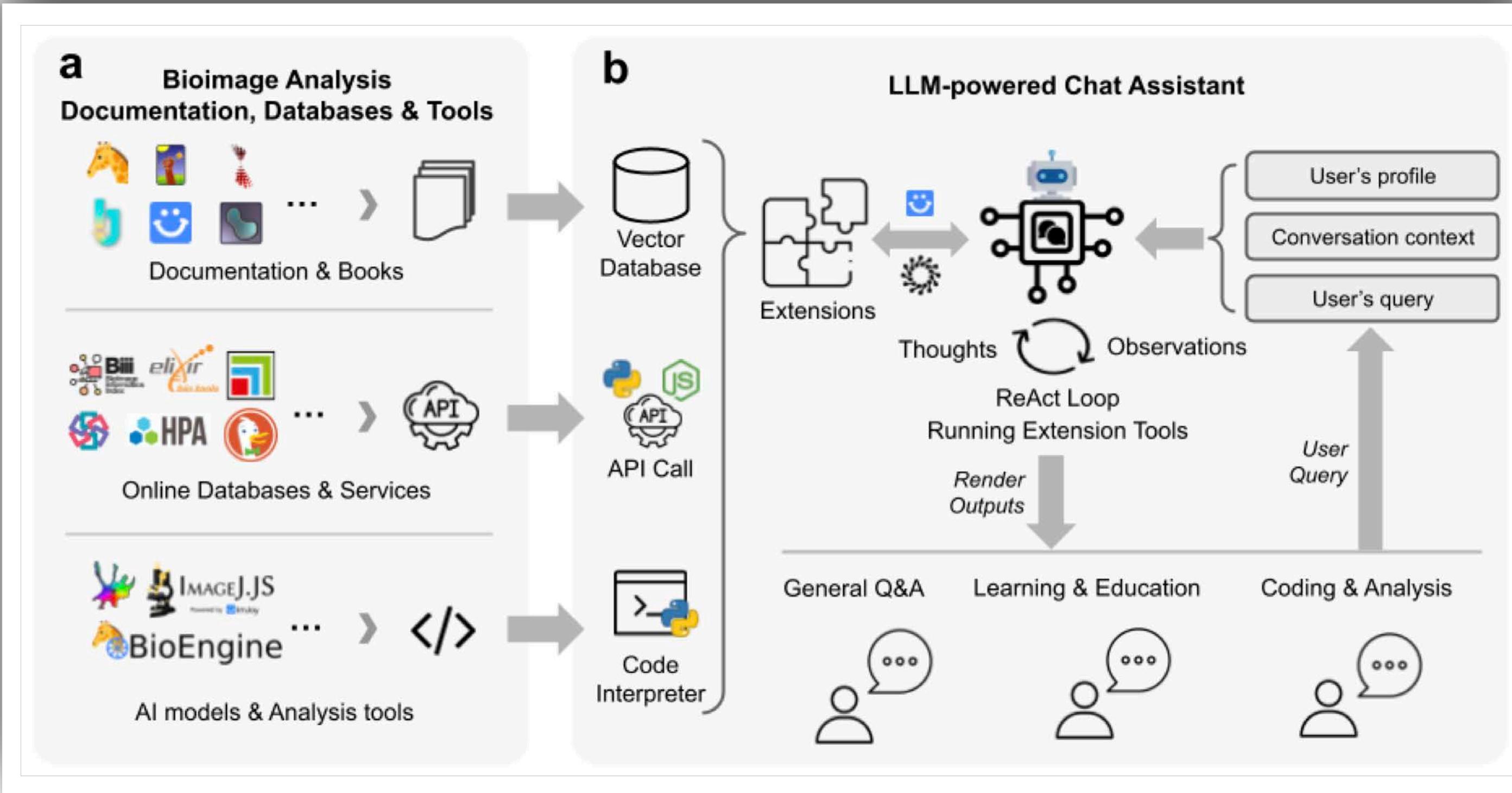
Original Image



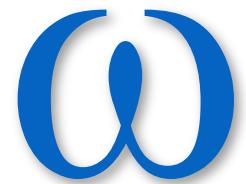
Convex Hull for Low Circular Nuclei



# BioImage Chatbot



# napari-omega



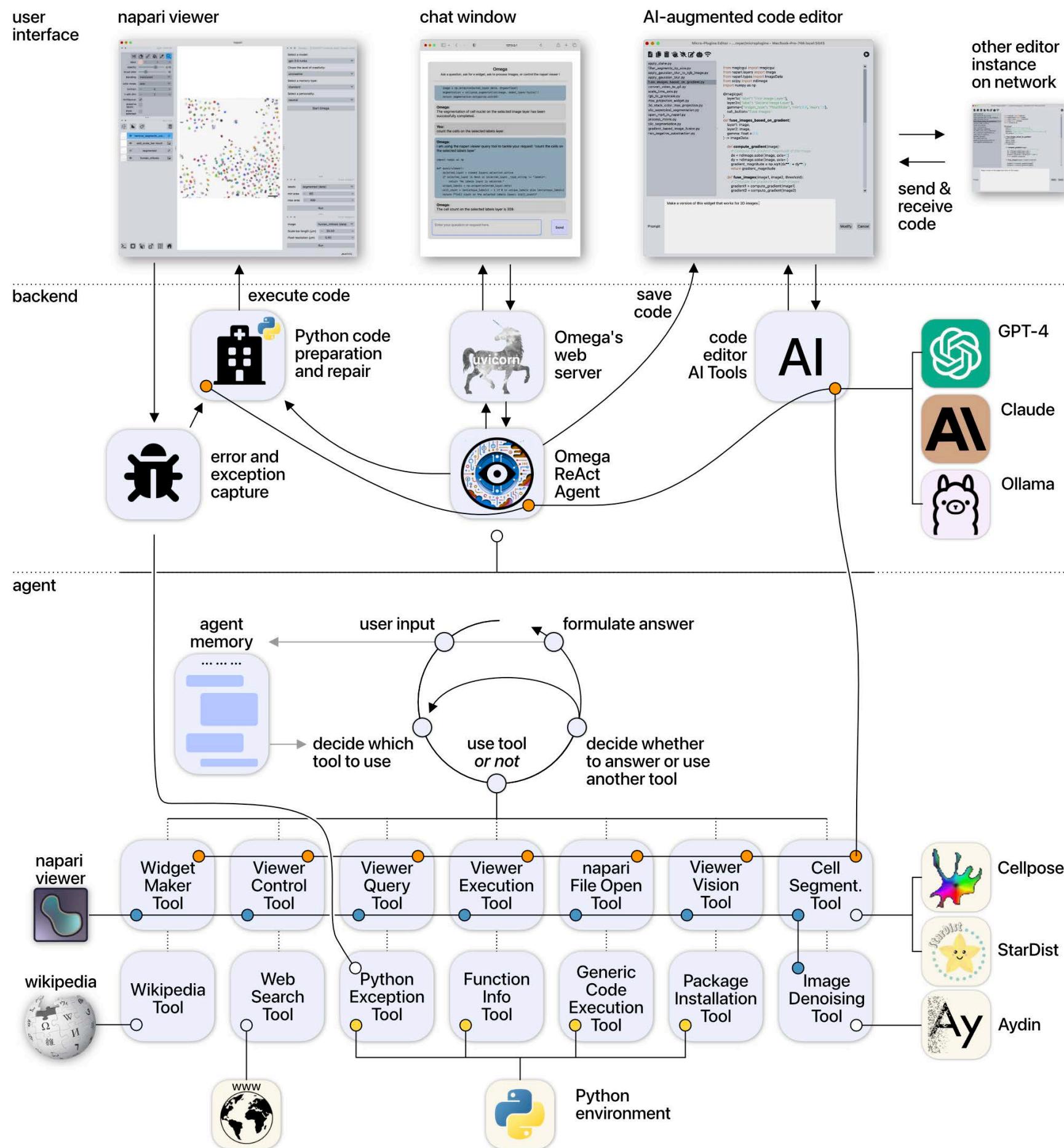
## Omega – Harnessing for Bioimage Analysis

Loïc A. Royer<sup>1,\*</sup>

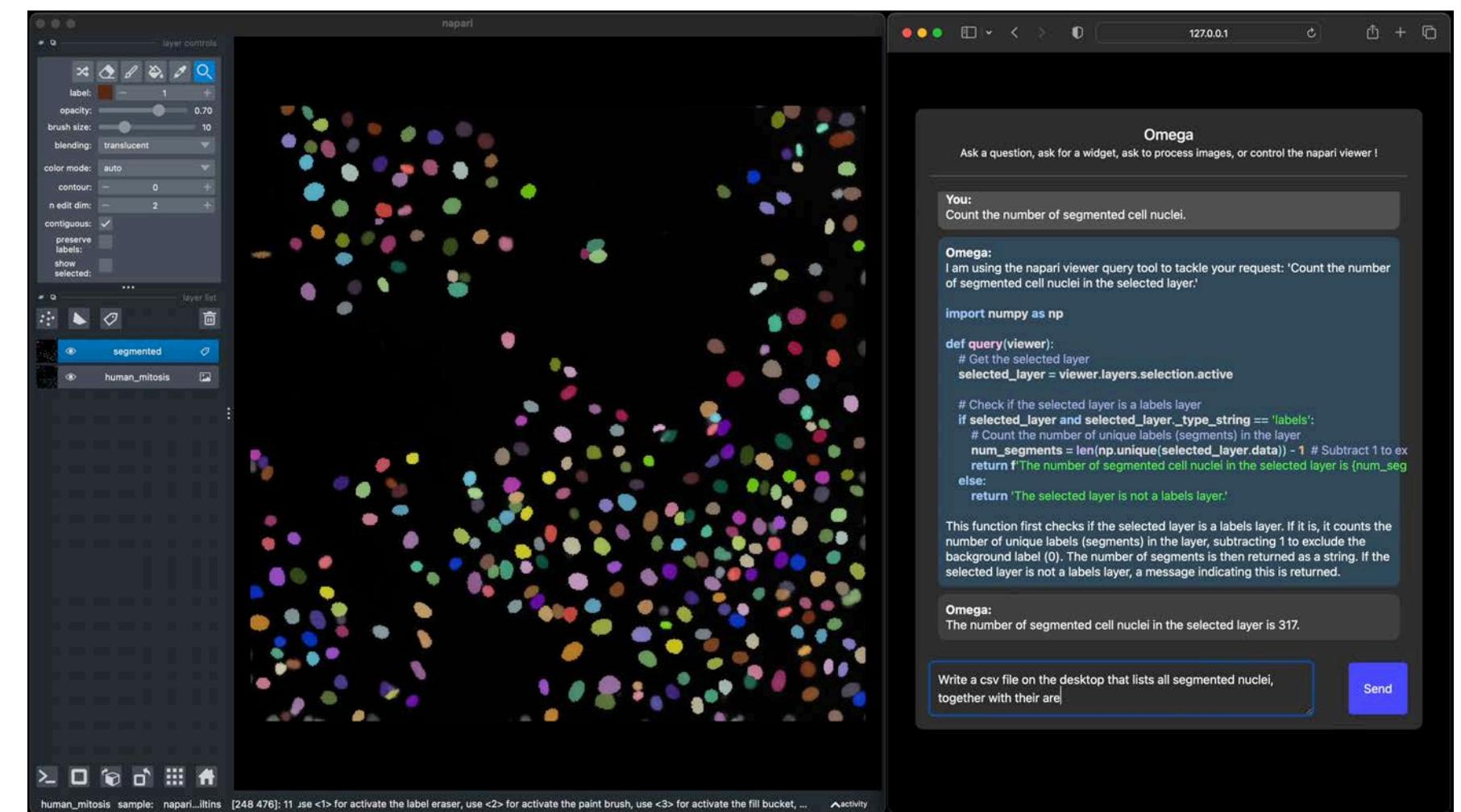
<sup>1</sup>Chan Zuckerberg Biohub, San Francisco, USA.

\*Correspondence: [loic.royer@czbiohub.org](mailto:loic.royer@czbiohub.org)

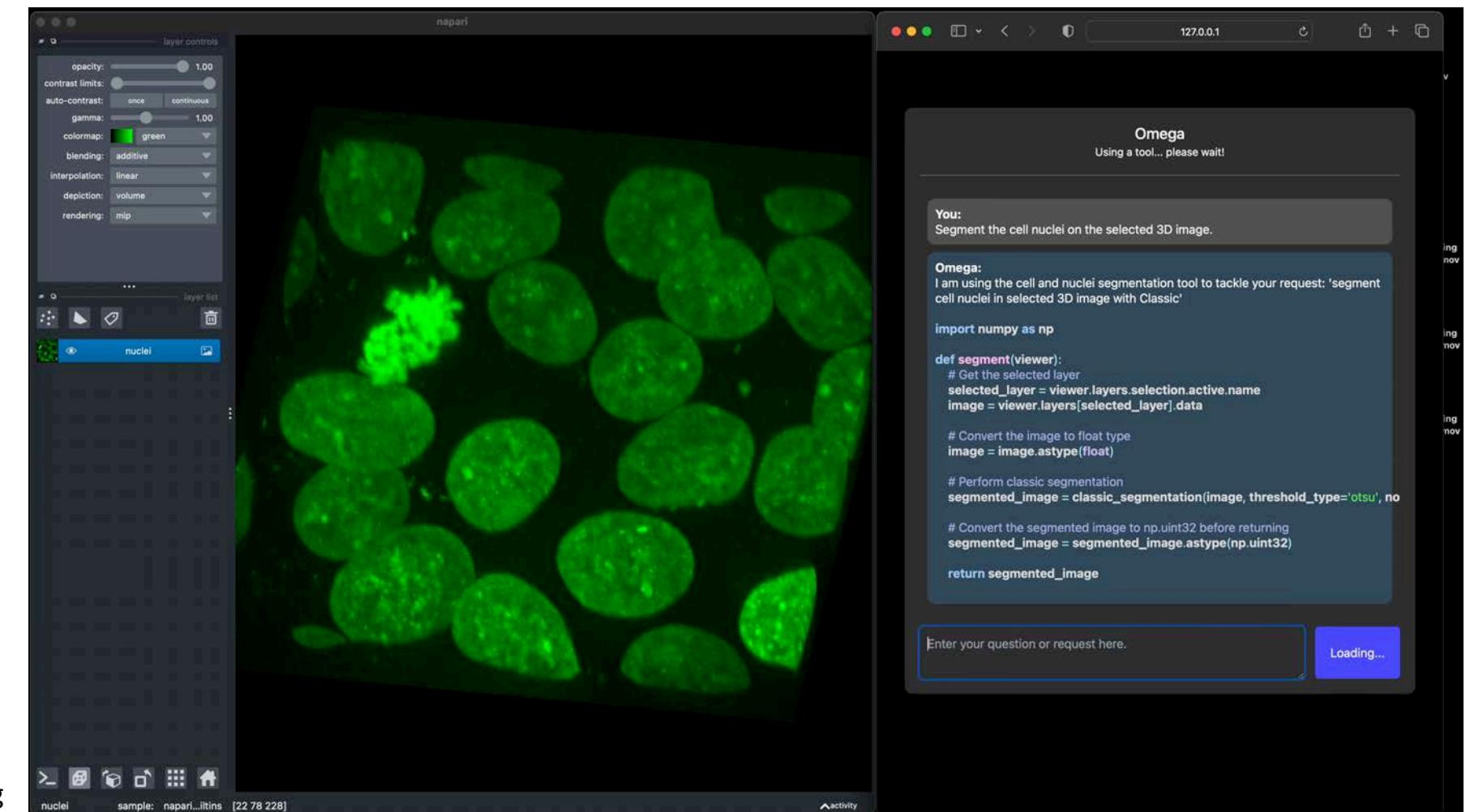
[10.5281/zenodo.10828225](https://doi.org/10.5281/zenodo.10828225)



**Supplementary Figure 3.** Omega's System Architecture. Diagram illustrating the Omega system architecture, encompassing both the user interface and backend components. The user interface is displayed at the top, consisting of the napari viewer, chat window, and AI-enhanced code editor. Key backend elements include the Omega ReAct agent, web server, AI tools within the editor, and the Python code repair module. The architecture employs a cascaded design, with a central dialog loop orchestrated by a primary language model. This model coordinates a suite of specialized tools, each powered by a secondary tool-specific language model that can access online resources, interact with the napari viewer, and integrate functionalities from third-party libraries such as Cellpose and StarDist.



**Supplementary Video 1.** Omega can segment nuclei with StarDist and perform follow-up analysis. The video showcases Omega's ability to segment cell nuclei in a 2D image using [StarDist](#). Omega successfully segments the nuclei and adds a label layer to the napari viewer. With further instructions, Omega can count the segmented nuclei and create a CSV file on the desktop folder of the machine. This file contains coordinates and areas of all segments, sorted by decreasing area, with one segment per row. Omega also opens the file using the system's default CSV viewer. The video has been sped up by a factor of 2.



**Supplementary Video 2.** Omega can segment nuclei in a 3D image. This video shows how Omega segments the nuclei in a 3D image displayed in the napari viewer. Omega uses a specialized tool for cell and nuclei segmentation and employs a 'classic' approach that combines single thresholding, specifically [Otsu](#), with watershed splitting to prevent under-segmentation. After segmentation, Omega adds a labels layer to the viewer, and we inquire about the number of segments detected. The response is 27. The video has been sped up by a factor of 2.



HARVARD  
MEDICAL SCHOOL

Human red blood cells  
DIC microscopy

## Tools

Upload Gallery

### Hover & Click

Click an object one or more times. Shift-click to remove regions.



Add Remove  
Mask Area

Res~~ndo~~redo

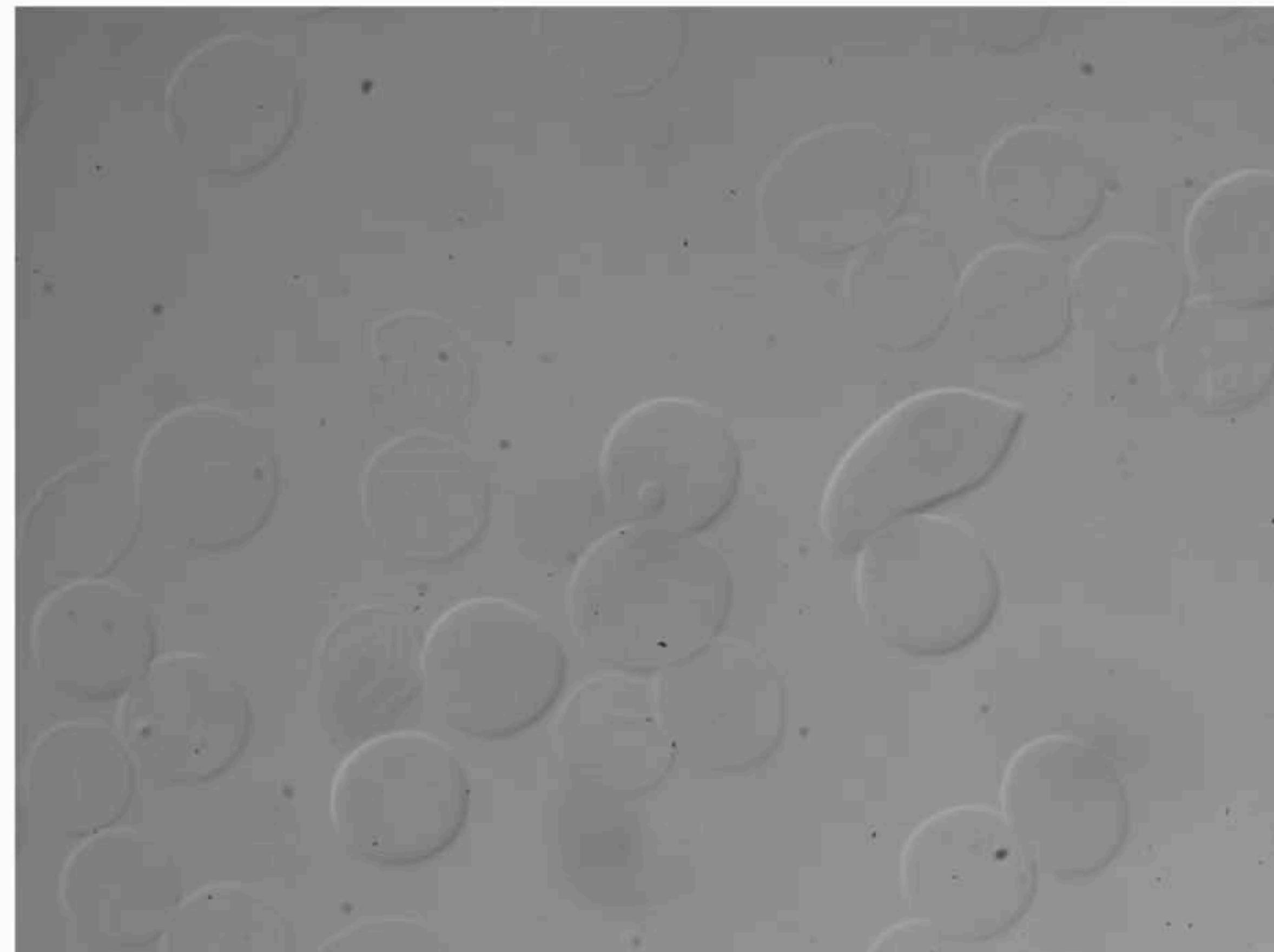
Multi-mask

Cut out object

Box

Everything

Cut-Outs



"[BBC009v1](#) from the Broad Bioimage Benchmark Collection [[Ljosa et al., Nature Methods, 2012](#)]."

# Segment Anything

Research by Meta AI

[Home](#)[Demo](#)[Dataset](#)[Blog](#)[Paper](#)

## Tools

Upload Gallery

### Hover & Click

Click an object one or more times. Shift-click to remove regions.

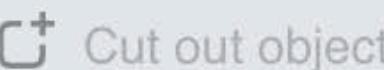


Add Remove  
Mask Area

Reset Undo Redo



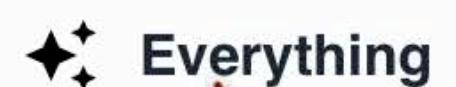
Multi-mask



Cut out object



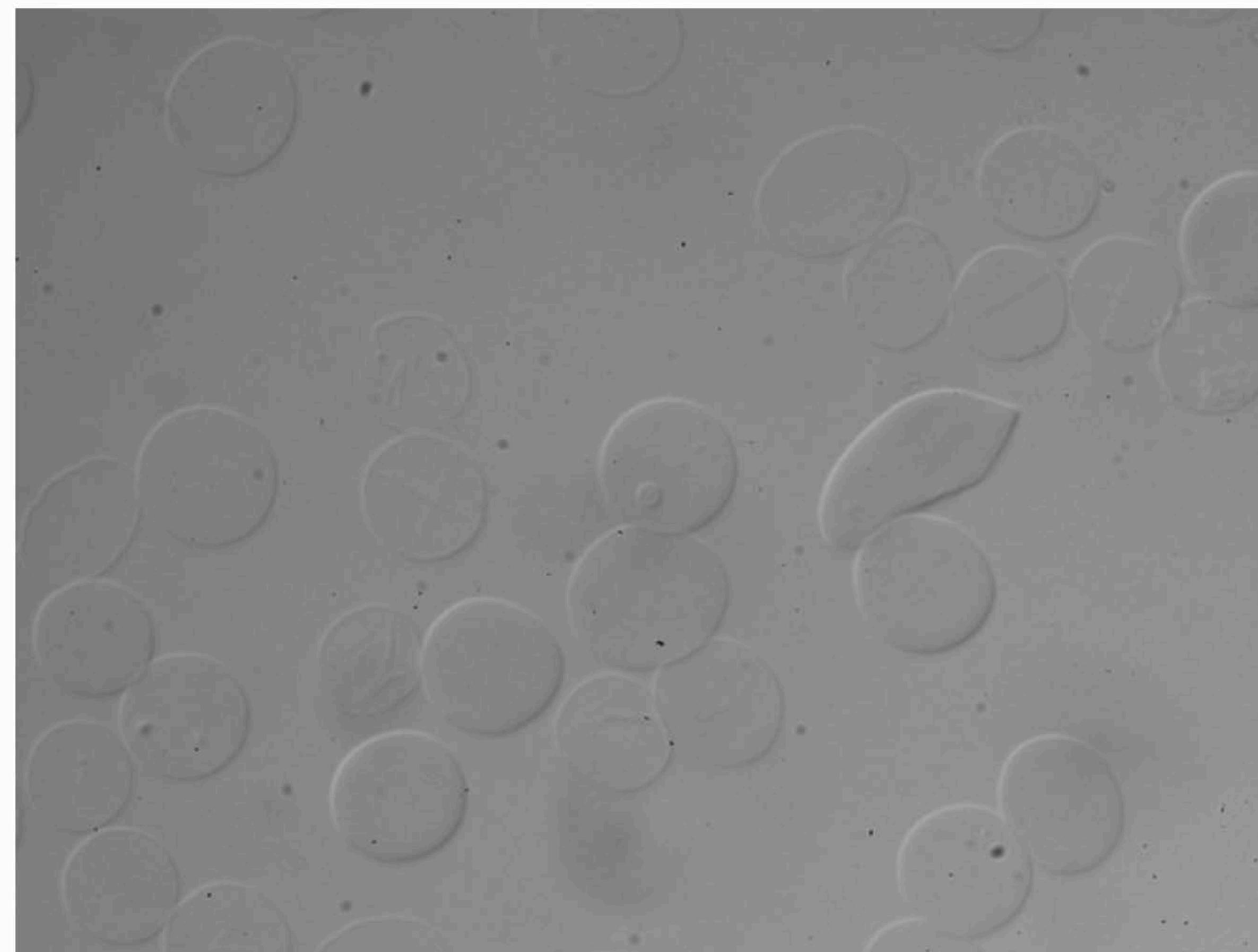
Box



Everything

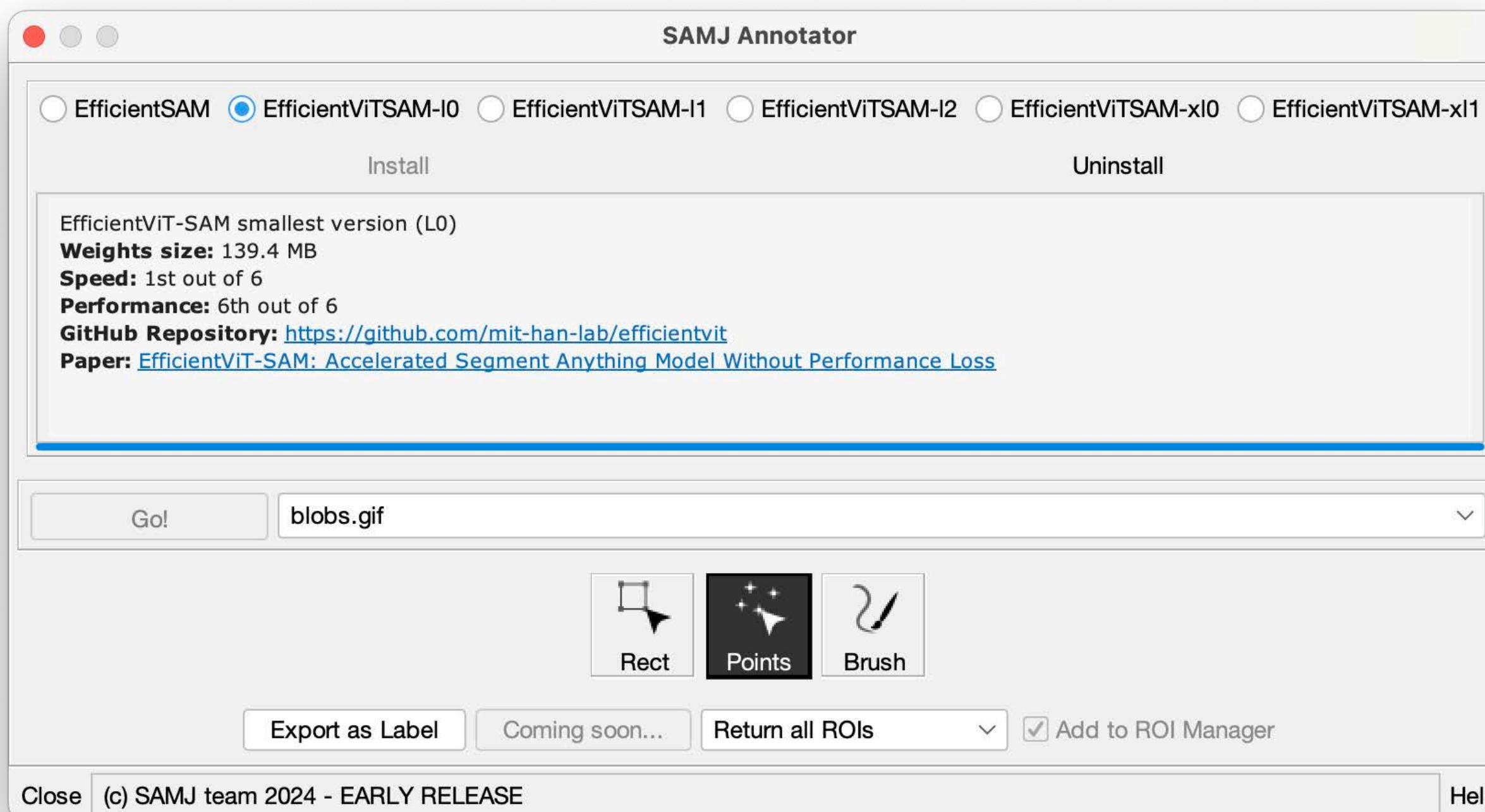


Cut-Outs

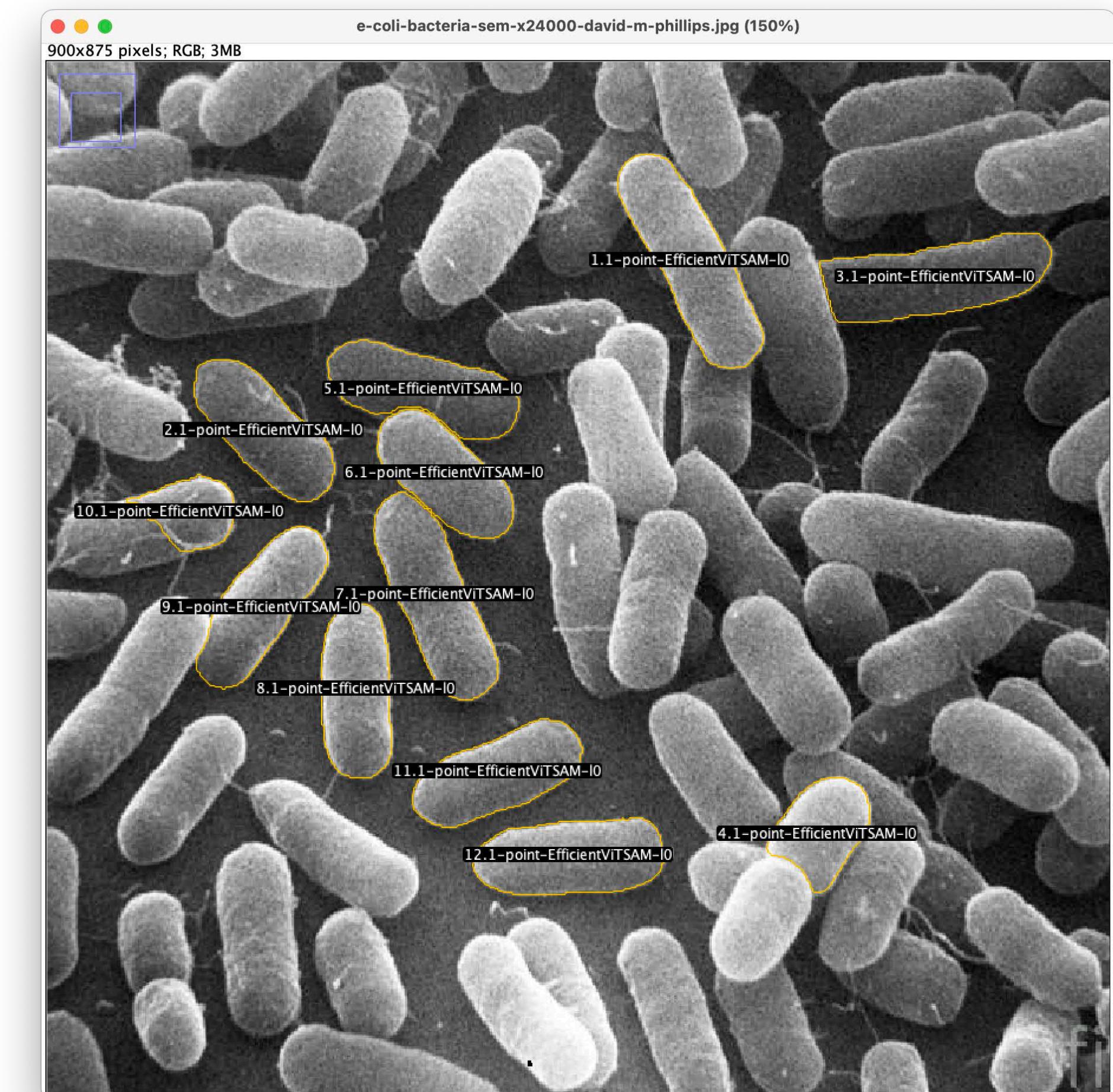


# Segment Anything in ImageJ/Fiji

Released around March 14th, 2024  
 Graphical User Interface  
 No coding required



Example of point annotations



<https://github.com/segment-anything-models-java/SAMJ-IJ>

# Image Analysis Workflow

- There are typically *five* steps in an image analysis
- Often a good idea to structure work along these lines before starting



Think of this even ***before*** you acquire the images!

otherwise image analysis may become only a *post-mortem* on your experiment



- **Data-wrangling in general**
  - Image format conversion, incl. lossless compression
  - Renaming
- **Restoration and Reconstruction**
  - Tomographic reconstruction
  - Registration (3D stacks)
  - Stitching (tiled images)
  - Illumination correction
  - Intensity normalisation
  - Deconvolution
  - Filtering (smoothing, sharpening, etc)
  - Grayscale morphological operations
- **Color deconvolution**
- **Channel alignment (chromatic aberrations)**
- ...



- **Image Segmentation**
  - Thresholding
  - Pixel Clustering
  - Edge detection
  - Region growing
  - Level set methods
  - Watershed
  - Model based
  - Multi-scale
  - Trainable pixel-classification
- **Spot detection**
- **Tracking**
- ...



- **Filtering** detected objects based on
  - Size, shape, intensity
  - Texture, location, neighbours
- **Binary morphological operations**
  - Open/close
  - Skeletonise
  - Hit-or-miss transform
  - ...



- **Object measures**
  - Size, shape, intensity
  - Texture, positions
  - Local neighbourhood
- **Image measures**
  - Object count
  - Area covered
  - Total intensity
- **Co-localisation**
- ...

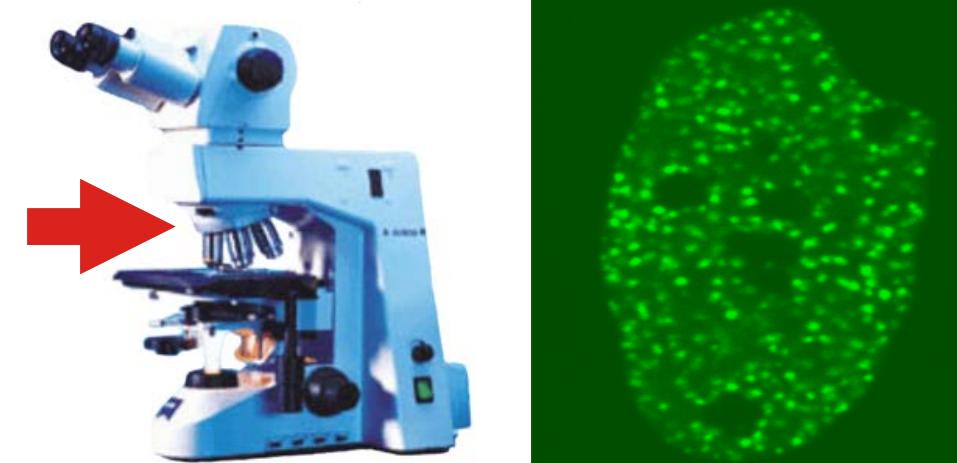


- Exploratory data analysis
- Filtering of numerical data based on measured values
- Summary statistics
- Plotting numerical values
- Making movies and montages of images
- Record entire work-flow for later reference
  - Write down what you did and why (each step)
  - Use a version control system
  - Document your code
  - ...

# Processing vs. Analysis

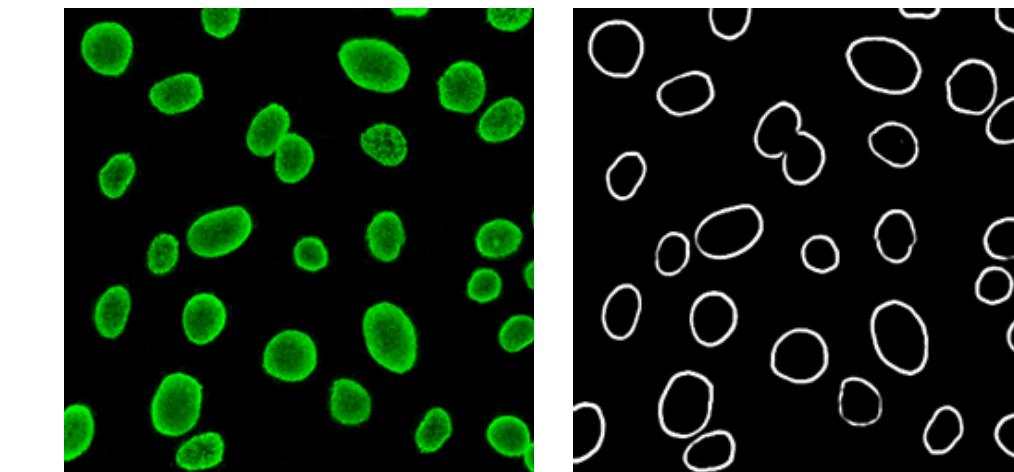
## Image Formation

object in → image out



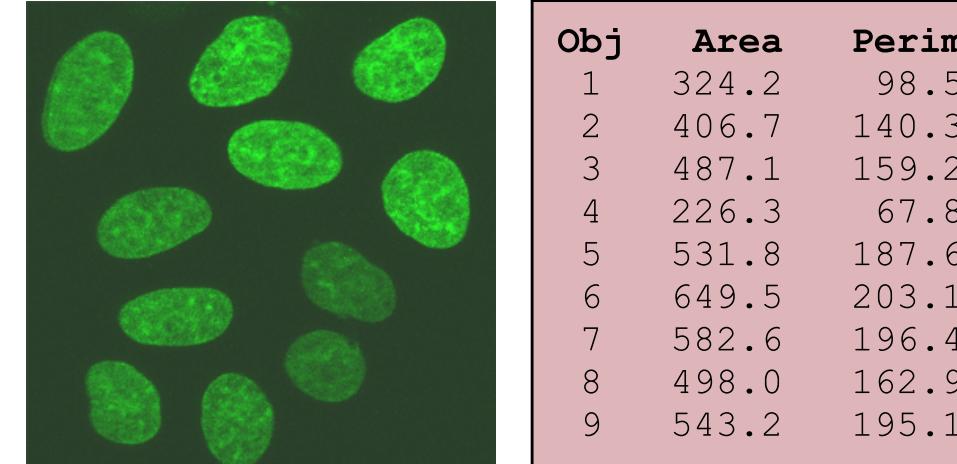
## Image Processing

image in → image out



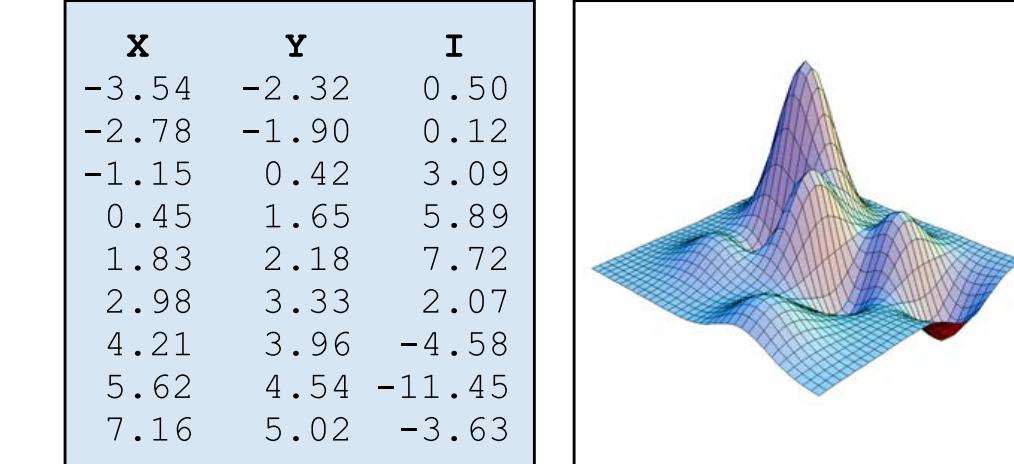
## Image Analysis

image in → features out



## Computer Graphics

numbers in → image out



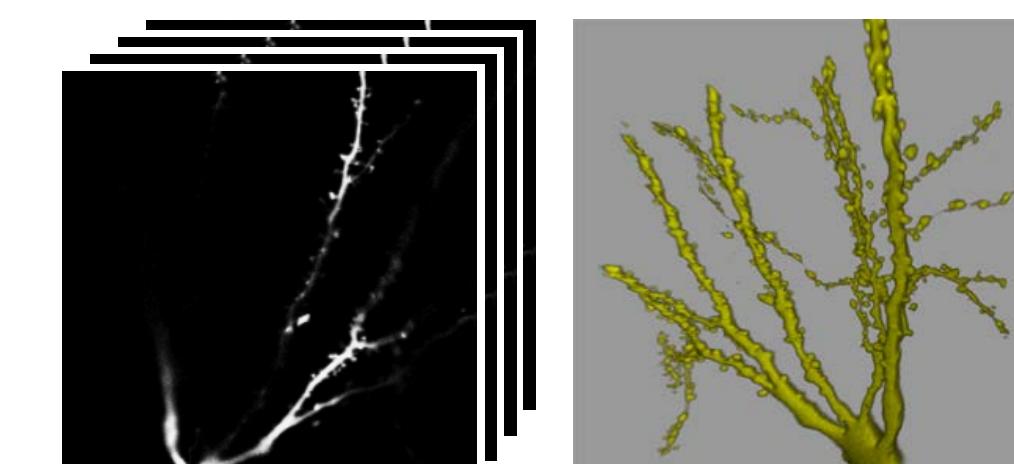
## Computer Vision

image in → interpretation out



## Visualization

image in → representation out



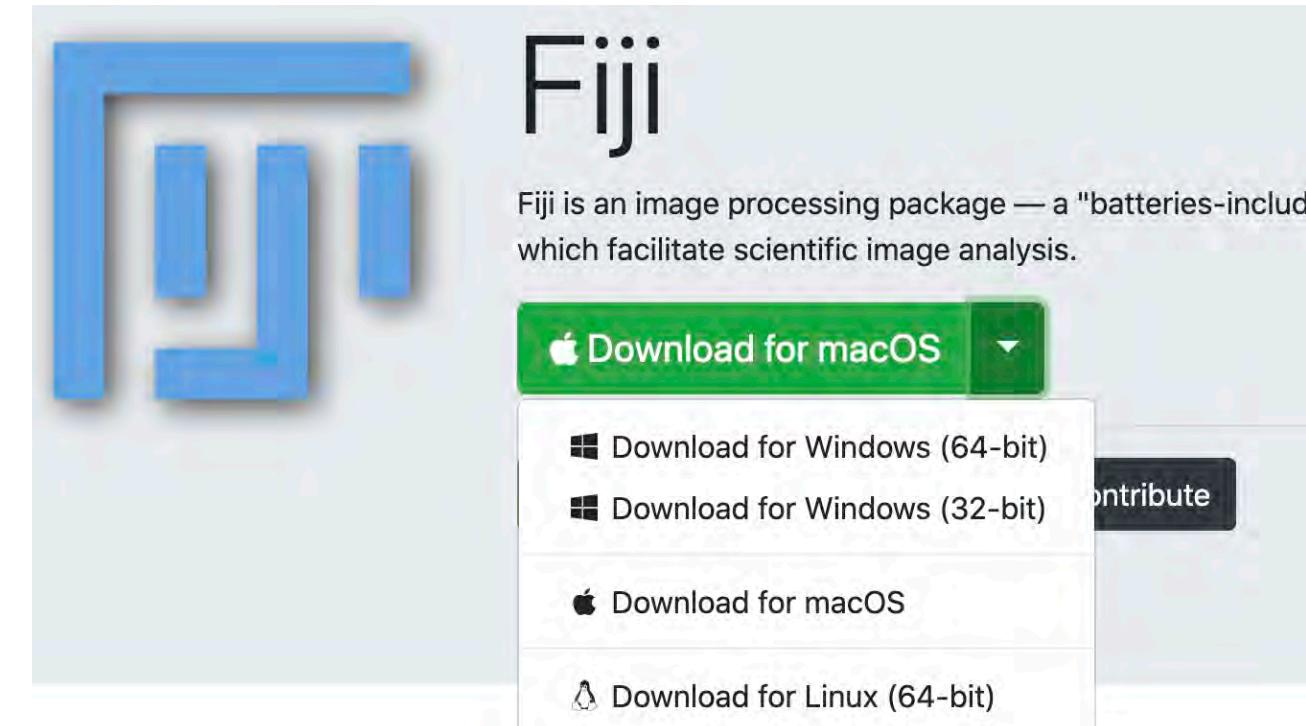
# Debugging Fiji and getting help — a workflow

1. **Restart Fiji**
2. **Ask your right neighbor**
3. **Ask your left neighbor**
4. **Ask an instructor**
5. **Search <https://forum.image.sc/>**

# Install and Update Fiji

# Download Fiji

<https://fiji.sc/>



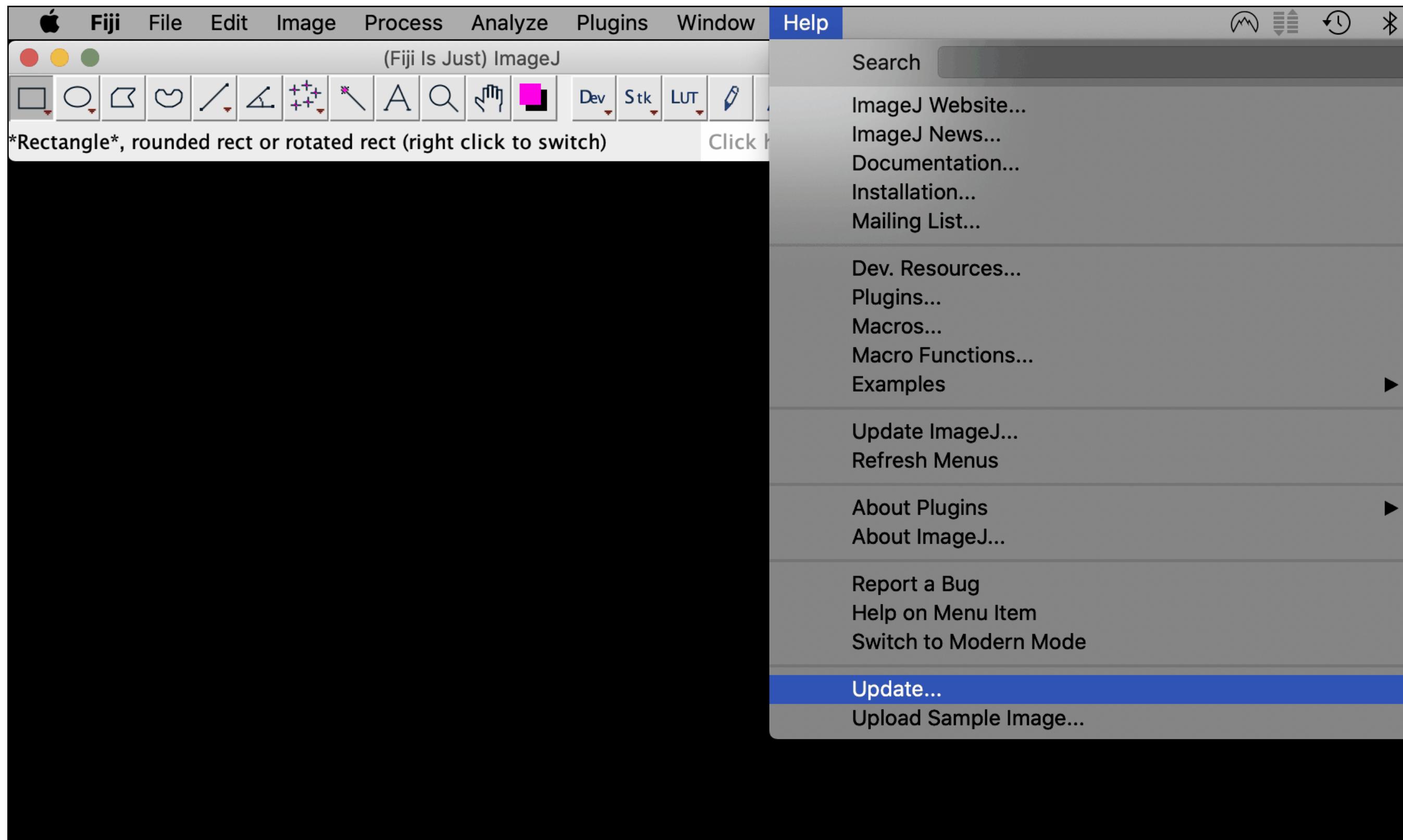
<https://imagej.net/Fiji>



Fiji is distributed as a **portable application**,  
which means that you do not have to run an installer.  
Just download (**zip file**), **unpack** and start it.

# Update Fiji

(and install/uninstall plugins)



**Help > Update...**

To update Fiji and/or the plugins.

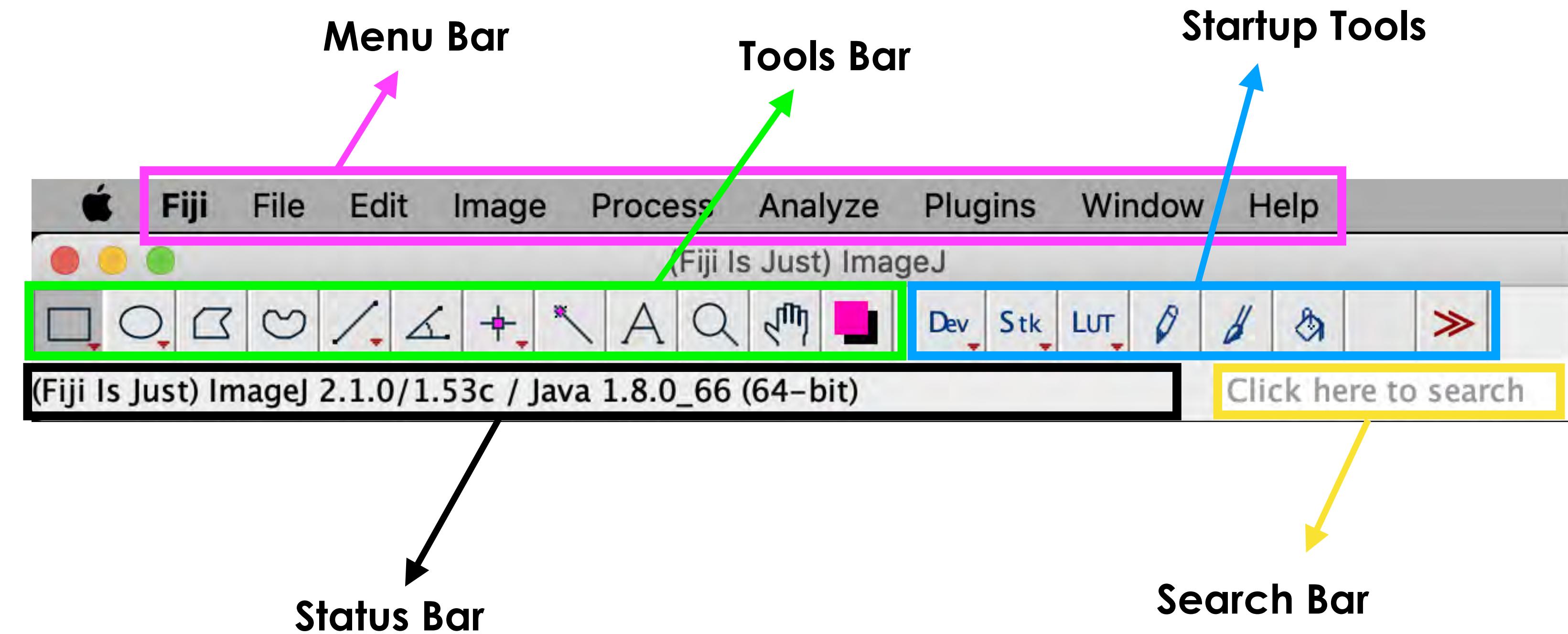
**Apply Changes:** Install/Update the listed plugins

**Manage Update Sites:** opens a list of plugins, you can select which one(s) to install in Fiji

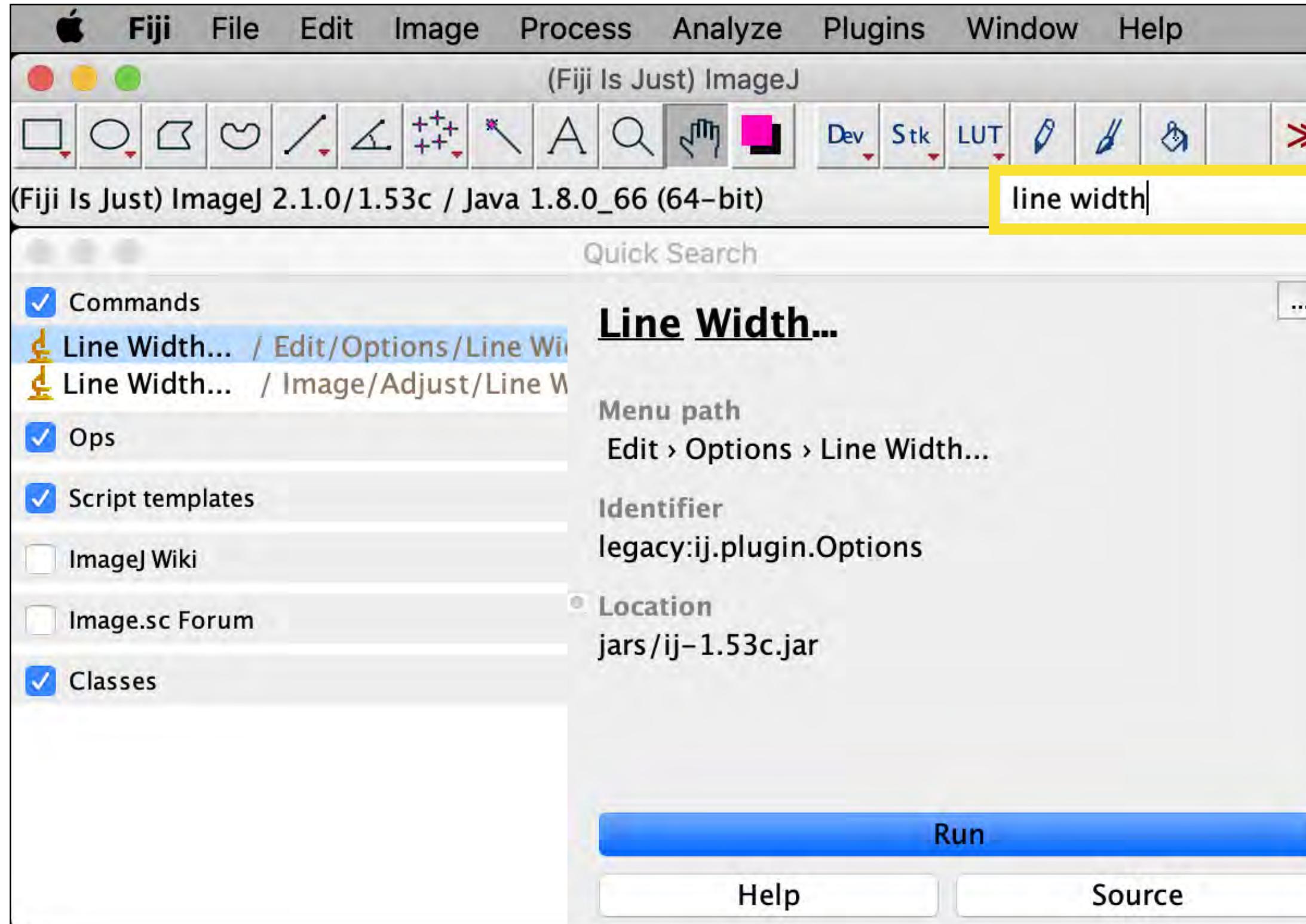


You can have more than one Fiji app!

# Graphic User Interface (GUI)



# Graphic User Interface (GUI)



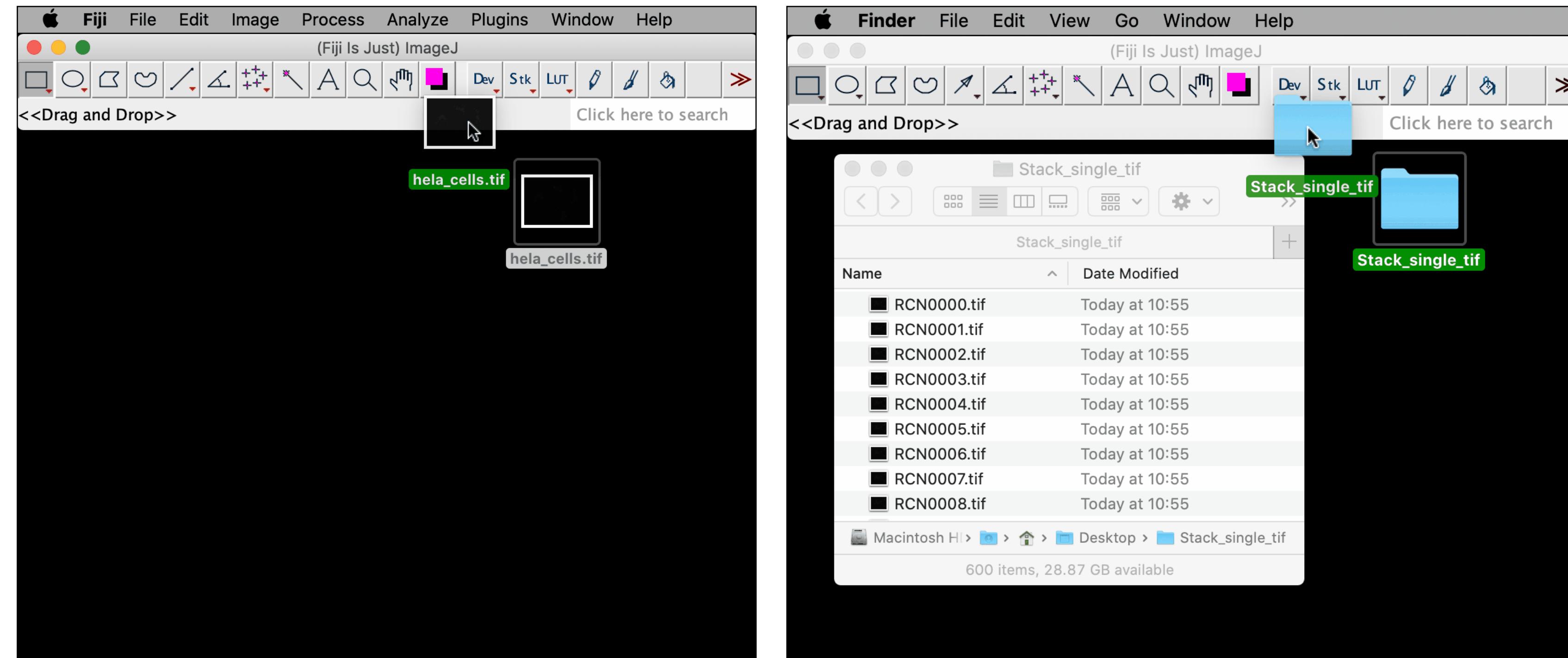
**Search Bar**  
(L key shortcut)

\***Plugins > Shortcuts > List Shortcuts - list of the default shortcuts**

\*\***Plugins > Utilities > Find Commands - search for Fiji Commands**

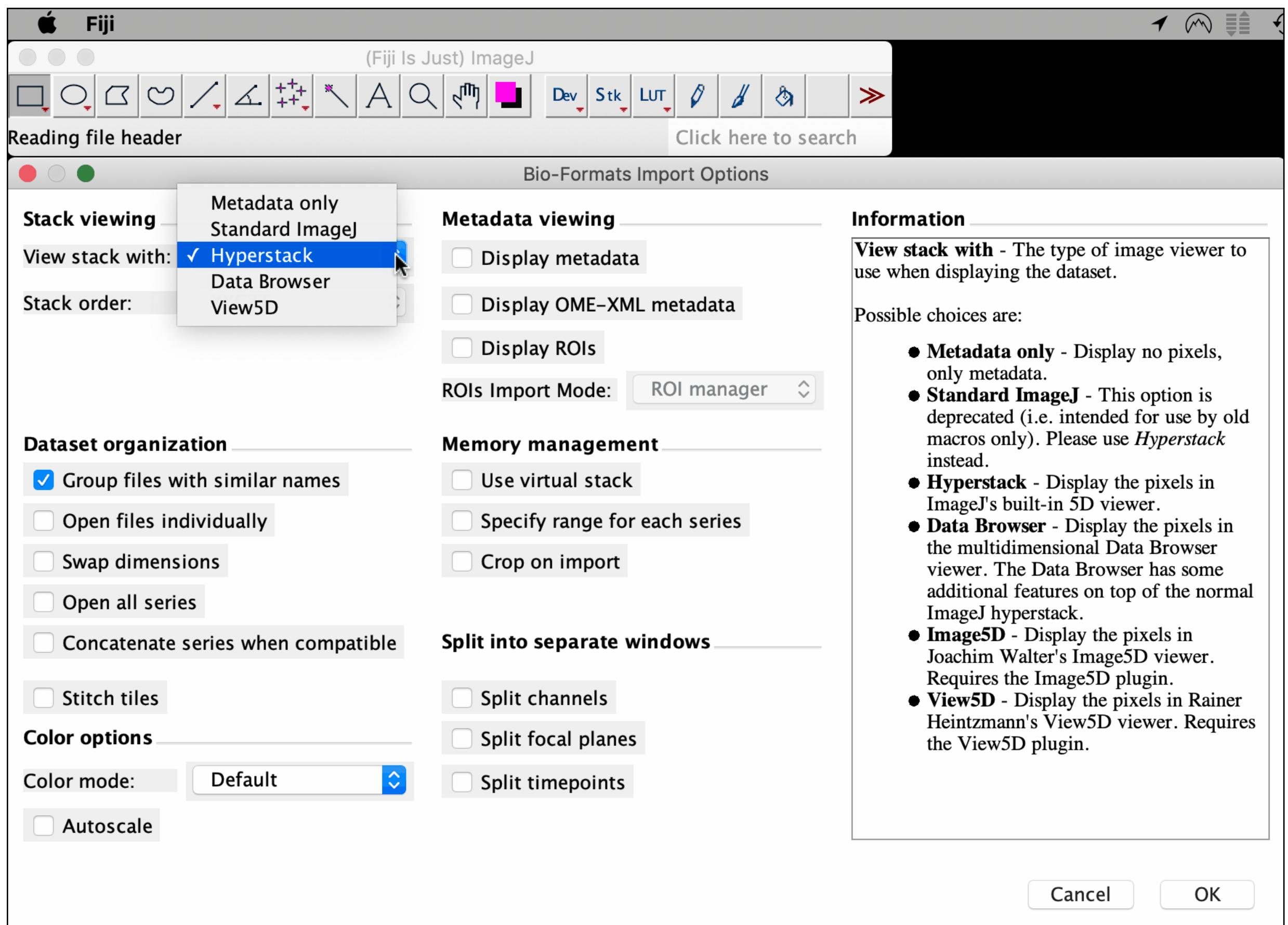
# Open a file in Fiji

***Drag and Drop the file you want to open onto the Status Bar.***



# Bio-Format Plugin

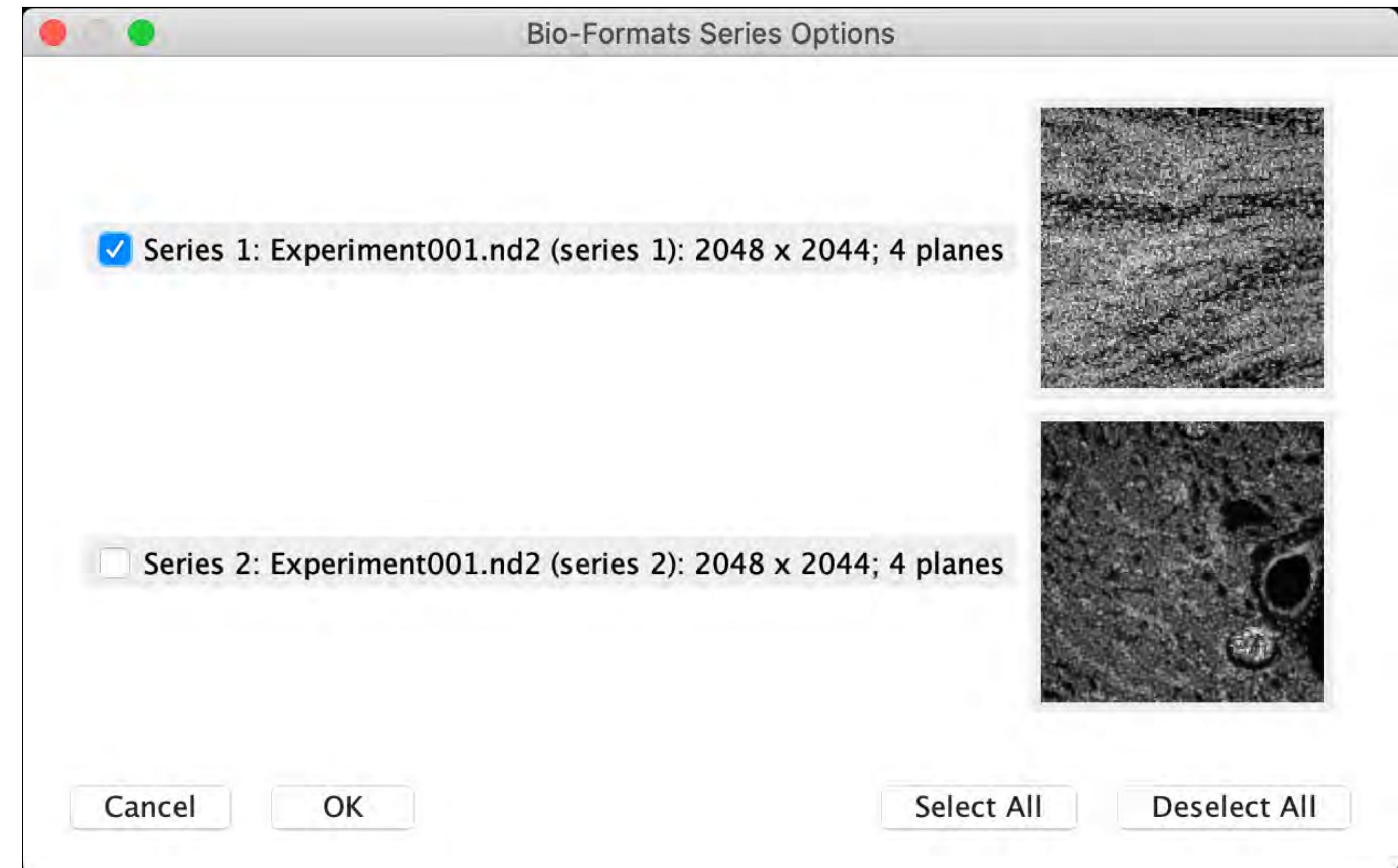
## import a file - options



**Import data from many life sciences file formats  
(e.g. @NIC .nd2 format)**

# Bio-Format Plugin

## import a file - options

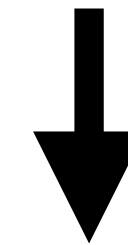


# Bio-Format Plugin

## import a file

Drag and Drop

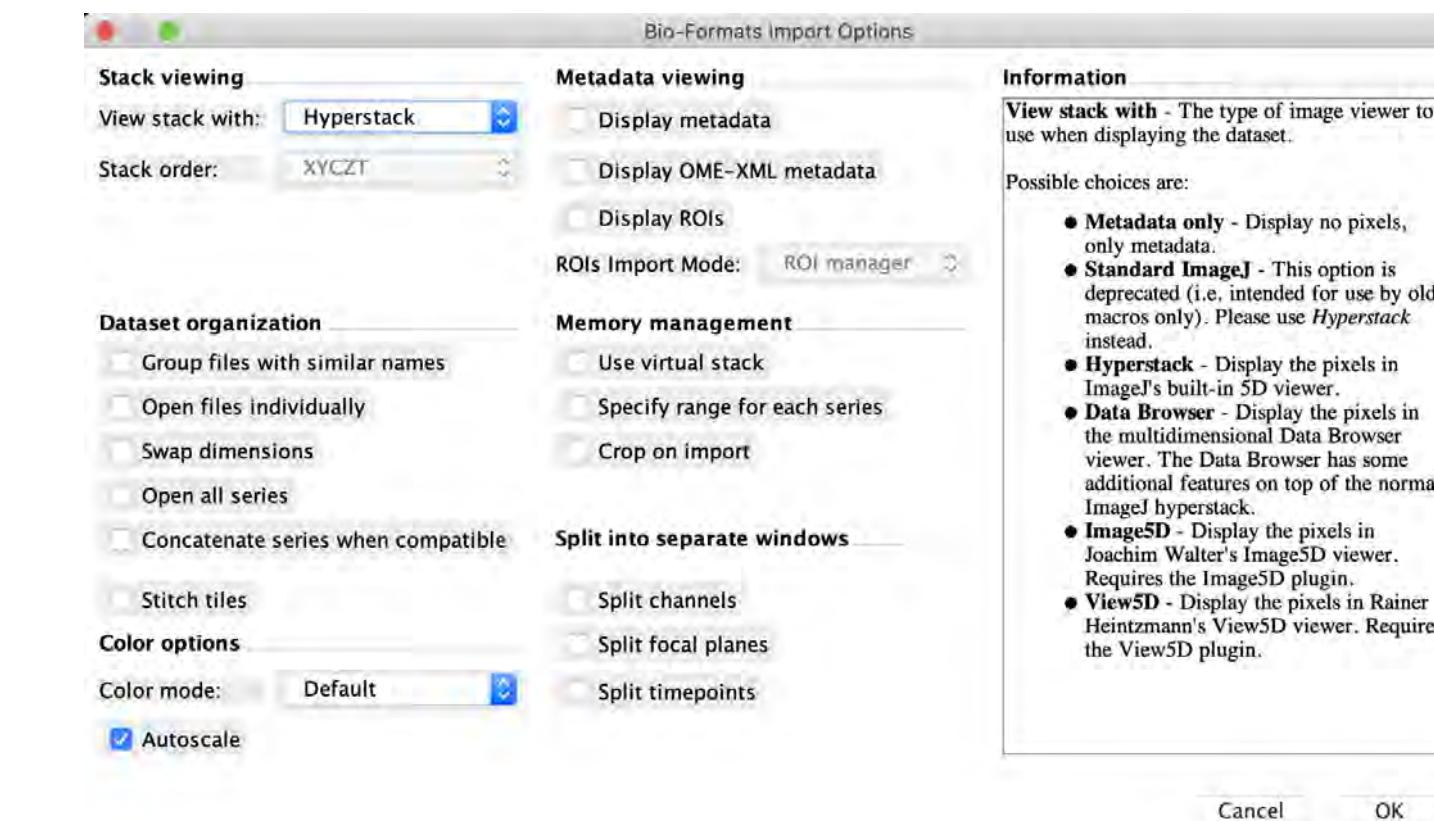
*If you have a **tiff** file, “**Drag and Drop**”  
does **not open Bio-Format Importer**.*



File > Import > Bio-Formats

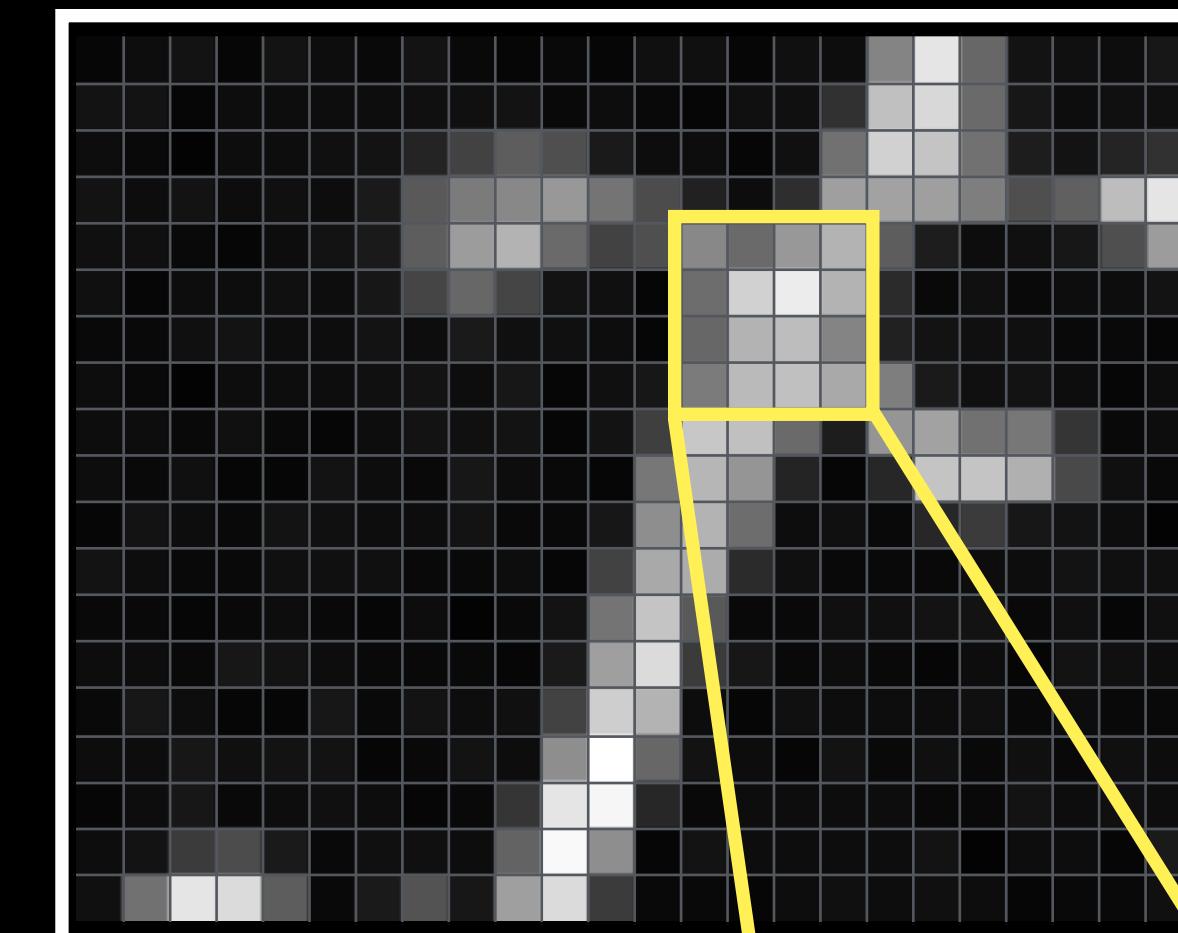
or

Plugins > Bio-Formats > Bio-  
Formats Importer



# what is an image?

# A digital image is a matrix of numbers!



=

6	13	19	6	19	13	9	19	9	6	9	6	16	16	6	16	13	132	229	103	19	16	13	23	9	9
19	19	6	13	13	13	13	16	16	19	9	13	9	6	16	16	49	192	216	106	23	13	16	16	23	13
13	9	4	13	13	16	19	36	66	93	79	26	13	13	6	16	113	209	196	113	29	19	36	49	36	33
19	13	19	13	16	13	26	89	123	136	152	116	76	33	13	46	159	162	159	126	79	96	189	229	226	212
16	16	9	6	13	19	26	93	156	179	106	66	79	136	106	152	179	93	29	13	16	23	79	156	123	49
16	6	13	13	16	13	23	69	103	69	19	16	6	109	209	236	179	43	9	16	9	13	13	19	13	13
9	9	16	19	13	13	19	13	26	16	16	13	6	103	179	189	132	33	19	16	16	9	9	6	6	6
13	9	4	13	13	13	16	19	13	23	6	16	23	123	186	192	169	126	26	16	19	13	6	13	16	13
13	13	9	16	9	6	13	19	16	19	6	19	63	199	192	106	29	149	162	113	119	53	9	13	6	13
13	9	16	6	6	19	13	9	23	13	9	6	119	182	149	36	6	39	196	196	176	73	16	9	9	9
6	19	13	9	19	16	13	13	19	9	9	23	142	179	109	13	16	9	39	59	23	19	13	4	9	9
19	13	9	9	16	16	16	9	9	13	6	66	169	172	43	16	9	9	9	13	13	19	16	16	16	9
9	9	6	9	13	9	6	13	4	9	19	116	196	89	9	9	16	16	19	19	9	16	6	16	9	9
13	13	9	23	19	13	9	9	9	6	26	159	219	59	23	9	13	9	6	13	6	19	16	13	16	13
9	23	13	6	6	23	9	19	13	16	66	206	179	13	6	16	13	13	13	16	9	13	9	9	16	13
13	13	23	16	19	19	6	9	19	13	142	255	103	19	13	6	19	9	16	9	16	9	16	13	23	9
6	13	23	9	13	16	13	6	9	53	229	246	39	9	13	13	13	13	9	9	19	13	16	13	13	13
13	19	59	76	26	9	16	16	13	99	249	142	6	19	13	13	13	13	19	4	13	13	6	26	9	13
16	113	229	219	93	9	26	83	23	159	219	59	9	9	6	13	16	13	16	13	6	9	9	16	23	9

=

136	106	152	179
109	209	236	179
103	179	189	132
123	186	192	169

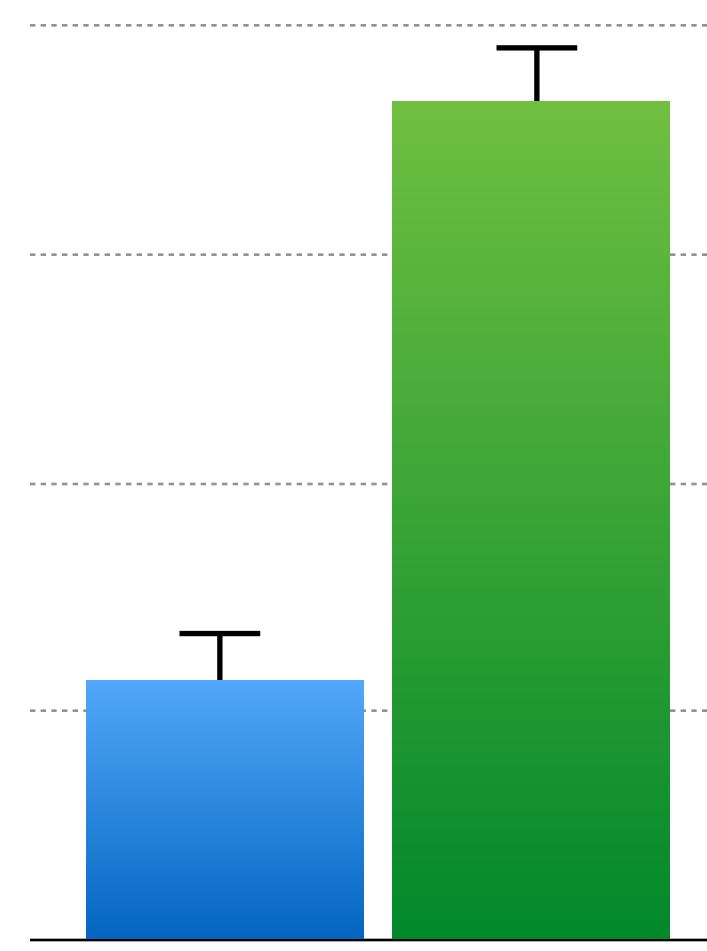
## Pixel = Picture Element

Images in publications and presentations  
should be used to **communicate** a finding...  
not **be** the finding

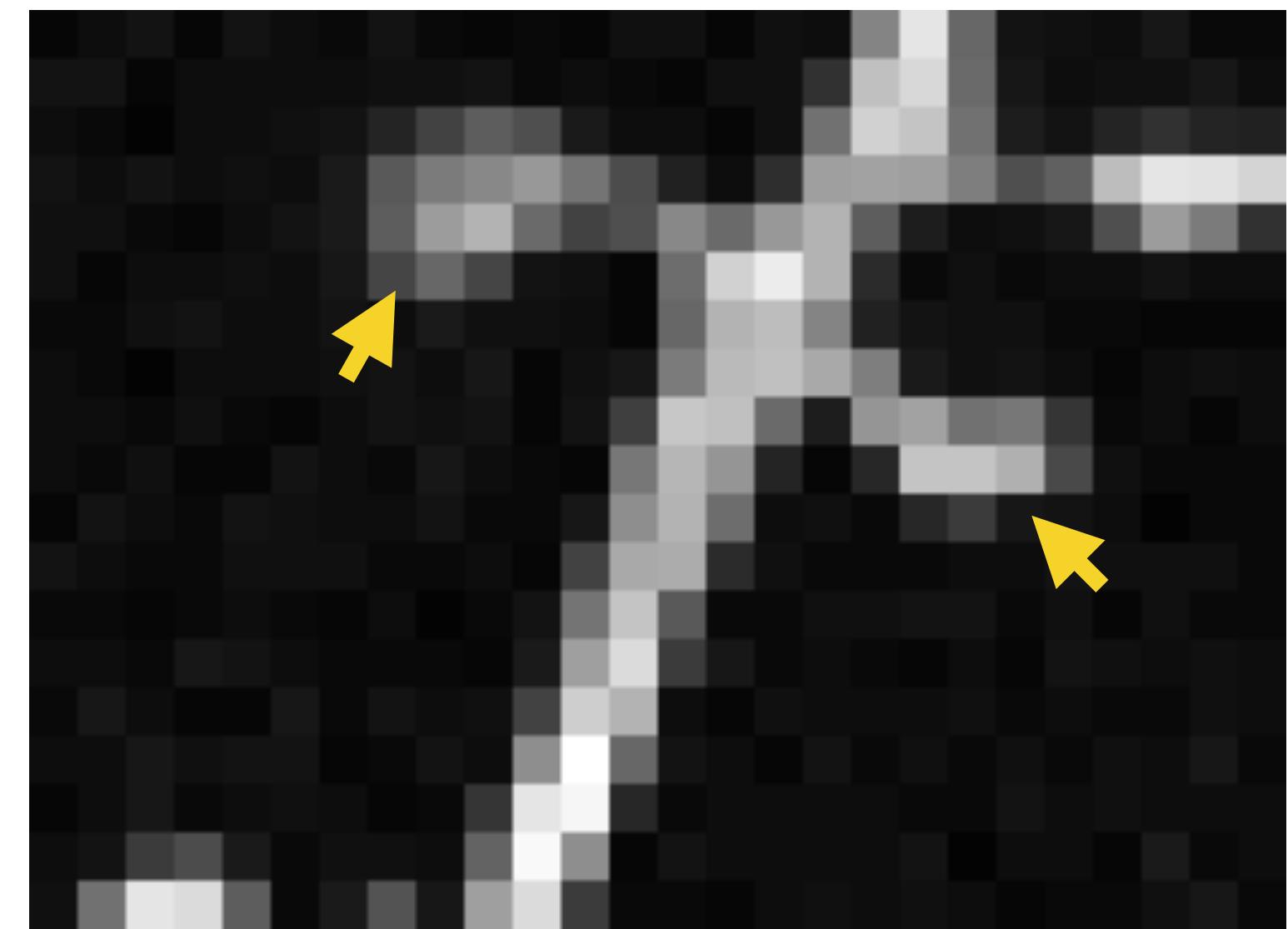
this is your **data**

6	13	19	6	19	13	9	19	9	6	9	6	16	16	6	16	13	132	229	103	19	16	13	23	9	9	
19	19	6	13	13	13	13	16	16	19	9	13	9	6	16	16	49	192	216	106	23	13	16	16	23	13	
13	9	4	13	13	16	19	36	66	93	79	26	13	13	6	16	113	209	196	113	29	19	16	49	36	33	
19	13	19	13	13	16	13	26	89	123	136	152	116	76	33	13	46	159	162	159	126	79	96	189	229	226	212
16	16	9	6	13	19	26	93	156	179	106	66	79	136	106	152	179	93	29	13	16	23	79	156	123	49	
16	6	13	13	16	13	23	69	103	69	19	16	6	109	209	236	179	43	9	16	9	13	13	19	13	13	
9	9	16	19	13	13	19	13	26	16	16	13	6	103	179	189	132	33	19	16	16	9	9	6	6	6	
13	9	4	13	13	13	16	19	13	23	6	16	23	123	186	192	169	126	26	16	19	13	6	13	16	13	
13	13	9	16	9	6	13	19	16	19	6	19	63	199	192	106	29	149	162	113	119	53	9	13	6	13	
13	9	16	6	6	19	13	9	23	13	9	6	119	182	149	36	6	39	196	196	176	73	16	9	9	9	
6	19	13	9	19	16	13	13	19	9	9	23	142	179	109	13	16	9	39	59	23	19	13	4	9	9	
19	13	9	9	16	16	16	9	9	13	6	66	169	172	43	16	9	9	9	13	13	19	16	16	16	9	
9	9	6	9	13	9	6	13	4	9	19	116	196	89	9	9	16	16	19	19	9	16	6	16	9	9	
13	13	9	23	19	13	9	9	6	26	159	219	59	23	9	13	9	6	13	6	19	16	13	16	13	13	
9	23	13	6	6	23	9	19	13	16	66	206	179	13	6	16	13	13	16	9	13	9	9	16	13	13	
13	13	23	16	19	19	6	9	19	13	142	255	103	19	13	6	19	9	16	9	16	9	16	13	23	9	
6	13	23	9	13	16	13	6	9	53	229	246	39	9	13	13	13	13	9	9	19	13	16	13	13	13	
13	19	59	76	26	9	16	16	13	99	249	142	6	19	13	13	13	13	19	4	13	13	6	26	9	13	
16	113	229	219	93	9	26	83	23	159	219	59	9	9	6	13	16	13	6	9	9	16	23	9			

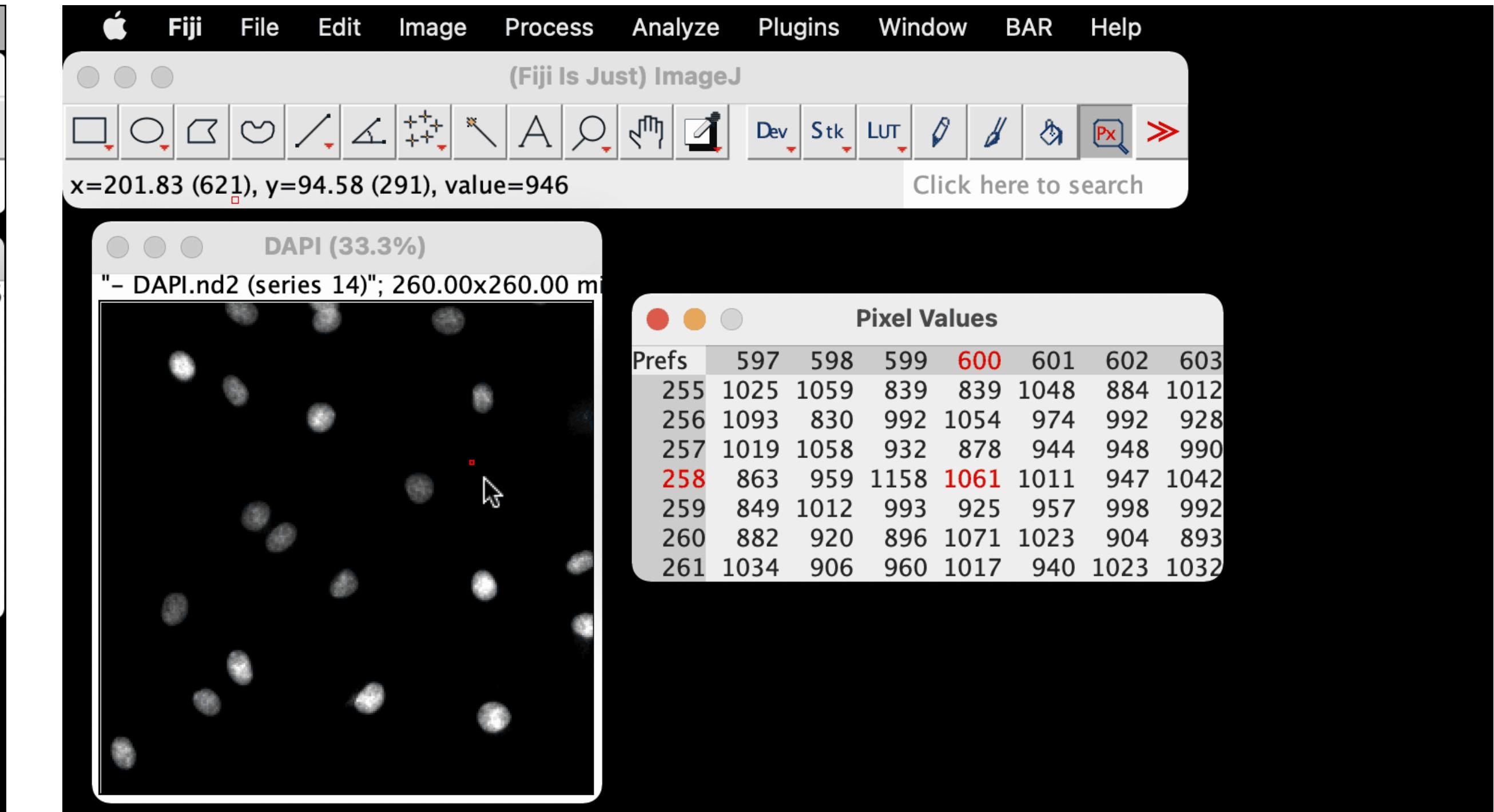
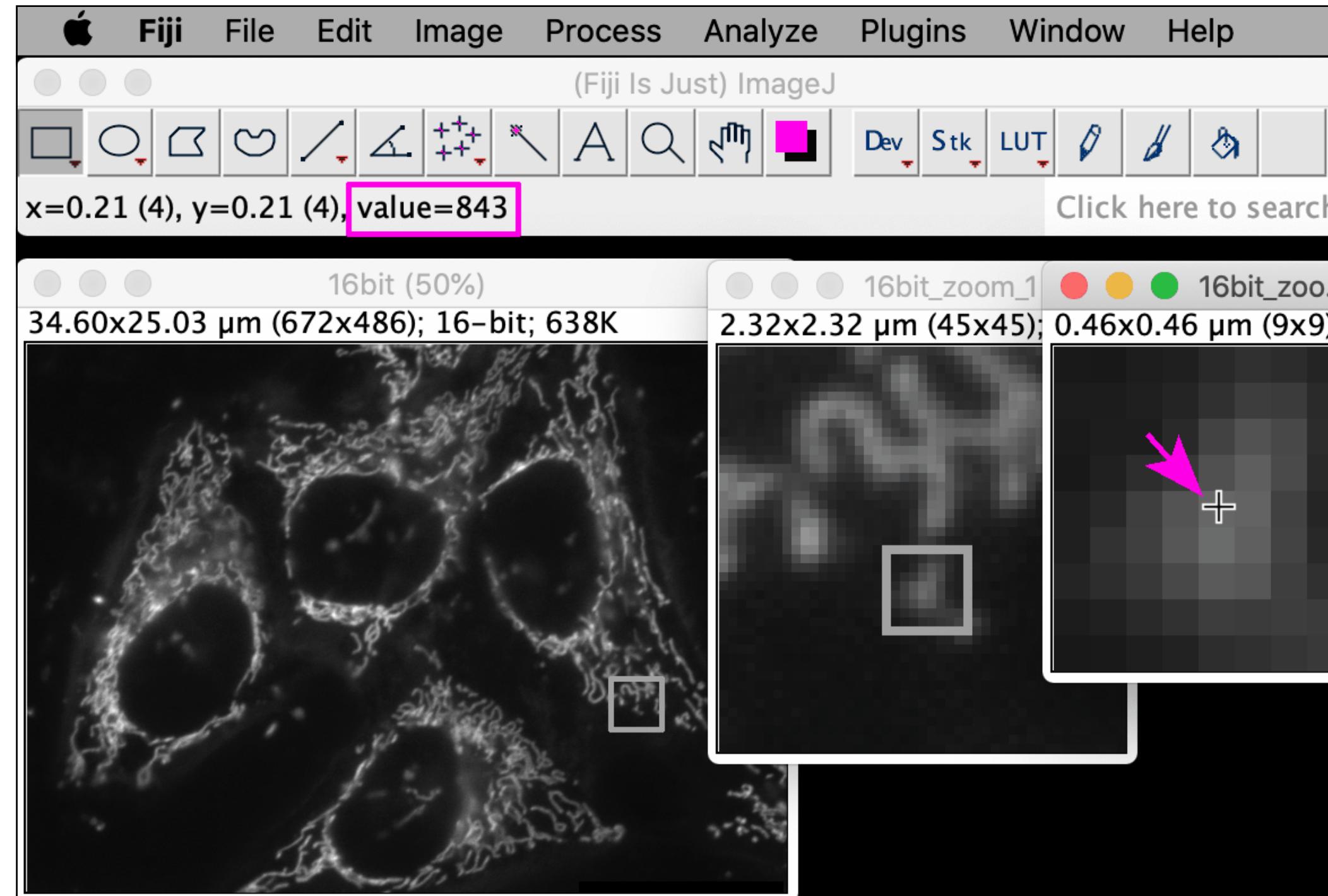
this is your **result**



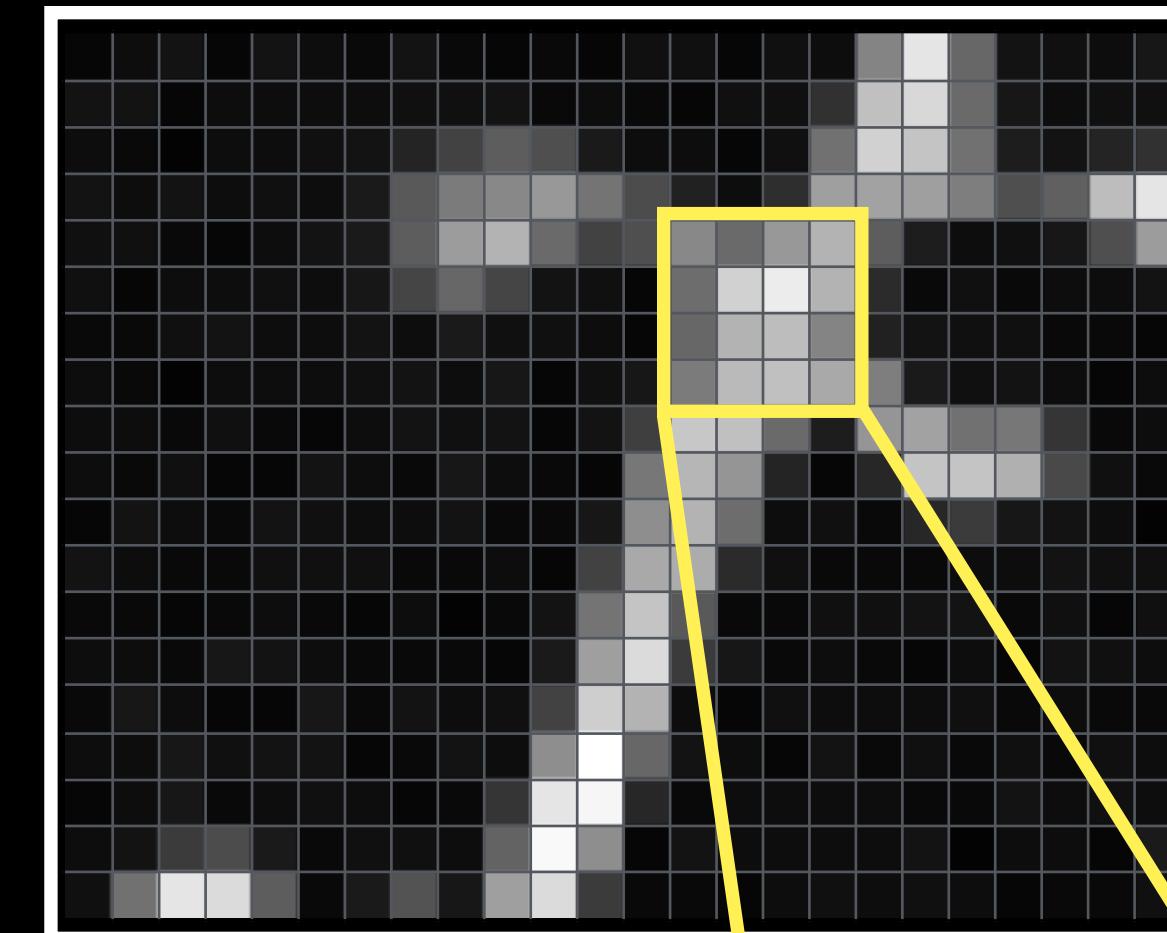
this just helps to  
**communicate** the result



# Individual Pixel Values in Fiji



# A digital image is a matrix of numbers!



=

6	13	19	6	19	13	9	19	9	6	9	6	16	16	6	16	13	132	229	103	19	16	13	23	9	9
19	19	6	13	13	13	13	16	16	19	9	13	9	6	16	16	49	192	216	106	23	13	16	16	23	13
13	9	4	13	13	16	19	36	66	93	79	26	13	13	6	16	113	209	196	113	29	19	36	49	36	33
19	13	19	13	16	13	26	89	123	136	152	116	76	33	13	46	159	162	159	126	79	96	189	229	226	212
16	16	9	6	13	19	26	93	156	179	106	66	79	136	106	152	179	93	29	13	16	23	79	156	123	49
16	6	13	13	16	13	23	69	103	69	19	16	6	109	209	236	179	43	9	16	9	13	13	19	13	13
9	9	16	19	13	13	19	13	26	16	16	13	6	103	179	189	132	33	19	16	16	9	9	6	6	6
13	9	4	13	13	13	16	19	13	23	6	16	23	123	186	192	169	126	26	16	19	13	6	13	16	13
13	13	9	16	9	6	13	19	16	19	6	19	63	199	192	106	29	149	162	113	119	53	9	13	6	13
13	9	16	6	6	19	13	9	23	13	9	6	119	182	149	36	6	39	196	196	176	73	16	9	9	9
6	19	13	9	19	16	13	13	19	9	9	23	142	179	109	13	16	9	39	59	23	19	13	4	9	9
19	13	9	9	16	16	16	9	9	13	6	66	169	172	43	16	9	9	9	13	13	19	16	16	16	9
9	9	6	9	13	9	6	13	4	9	19	116	196	89	9	9	16	16	19	19	9	16	6	16	9	9
13	13	9	23	19	13	9	9	9	6	26	159	219	59	23	9	13	9	6	13	6	19	16	13	16	13
9	23	13	6	6	23	9	19	13	16	66	206	179	13	6	16	13	13	13	16	9	13	9	9	16	13
13	13	23	16	19	19	6	9	19	13	142	255	103	19	13	6	19	9	16	9	16	9	16	13	23	9
6	13	23	9	13	16	13	6	9	53	229	246	39	9	13	13	13	13	9	9	19	13	16	13	13	13
13	19	59	76	26	9	16	16	13	99	249	142	6	19	13	13	13	13	19	4	13	13	6	26	9	13
16	113	229	219	93	9	26	83	23	159	219	59	9	9	6	13	16	13	16	13	6	9	9	16	23	9

=

136	106	152	179
109	209	236	179
103	179	189	132
123	186	192	169

## Where do these numbers come from?

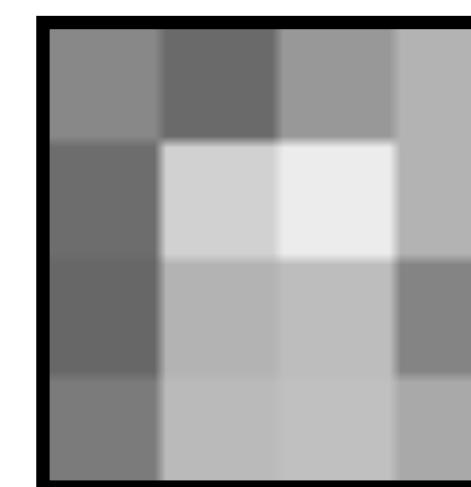
# Bit depth

# Detectors in Fluorescence Microscopy

The detectors used in fluorescence microscopy are **monochromatic**.

Cameras or PMTs are **not able to distinguish between different wavelengths** (they just collect photons), you need **fluorescence filters** to separate your fluorophores.

The detector converts photons in digital numbers (linear relation).



=

136	106	152	179
109	209	236	179
103	179	189	132
123	186	192	169

Each pixel in the digital image has **one digital value** that **depends on** the **intensity** of the signal emitted by the **sample**.

**Digital Values = Pixel Intensity Value**

The **range** of possible **digital values** is defined by the **bit depth**.

# Bit Depth

The **bit depth** defines the range of possible **digital values** that each pixel can have, usually **8, 12 or 16 bit**.

The **bit depth** is expressed in **grey values**.

**bit depth of the image = bit depth of the detector**

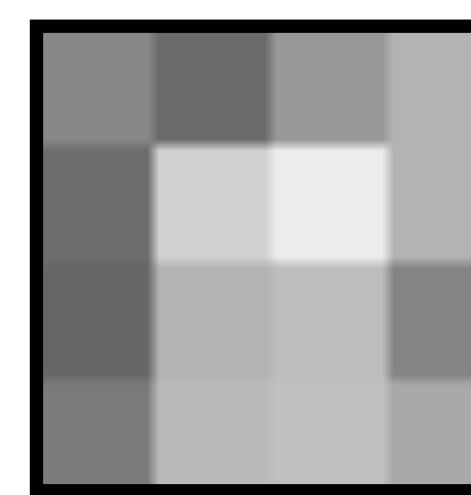
(Unless you change that during acquisition)

**$x$  bit = a range of  $2^x$  grey values**

**8 bit** image = **each pixel** can have  $2^8$  **grey values** = 256 grey values = **range 0-255**

**12 bit** image = **each pixel** can have  $2^{12}$  **grey values** = 4096 grey values = **range 0-4095**

**16 bit** image = **each pixel** can have  $2^{16}$  **grey values** = 65536 grey values = **range 0-65535**

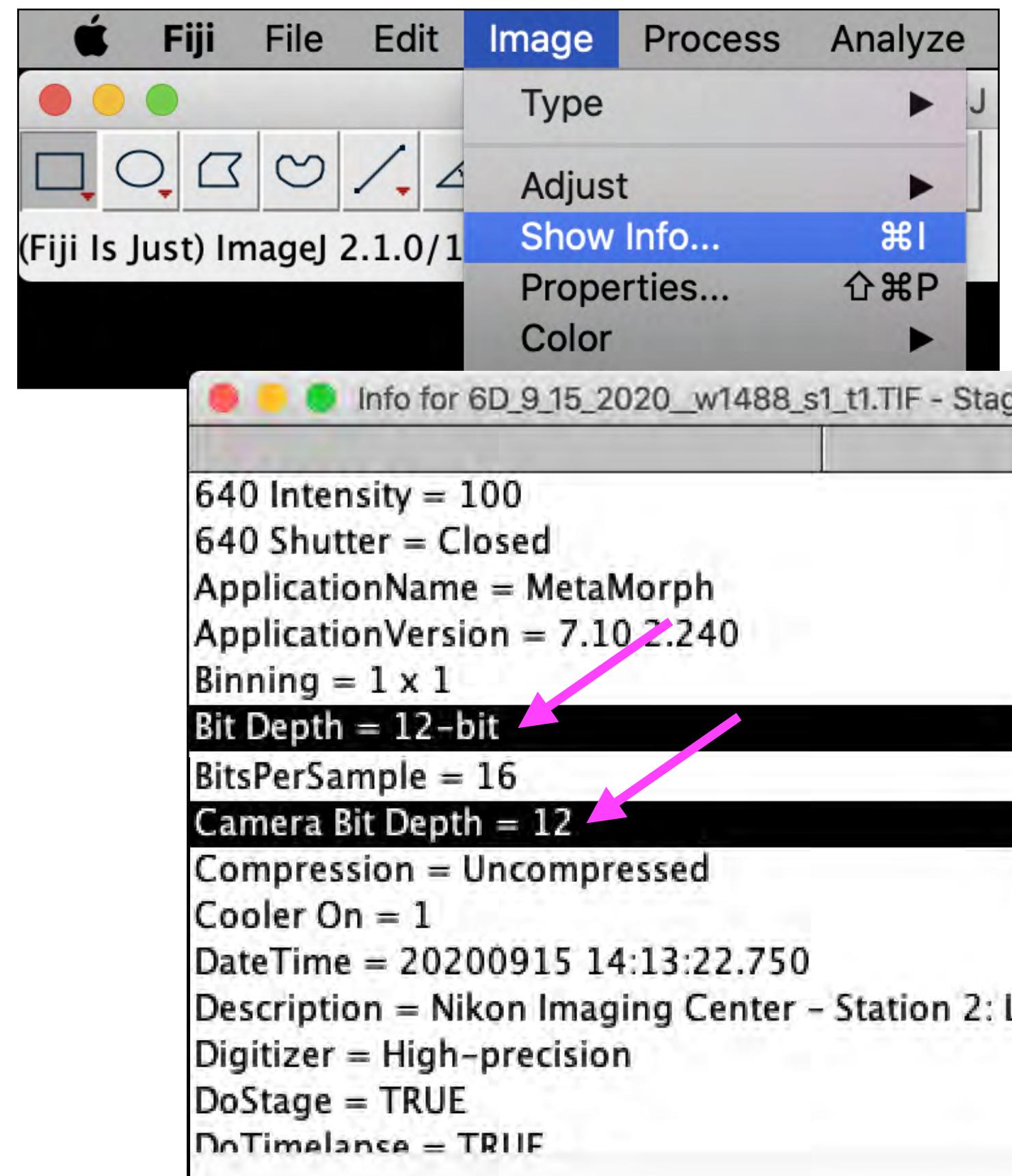


=

136	106	152	179
109	209	236	179
103	179	189	132
123	186	192	169

**Digital Value = Pixel Intensity Value = Grey Value**

# Image Metadata

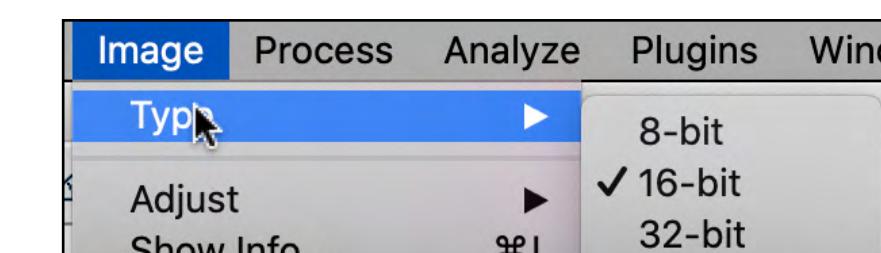
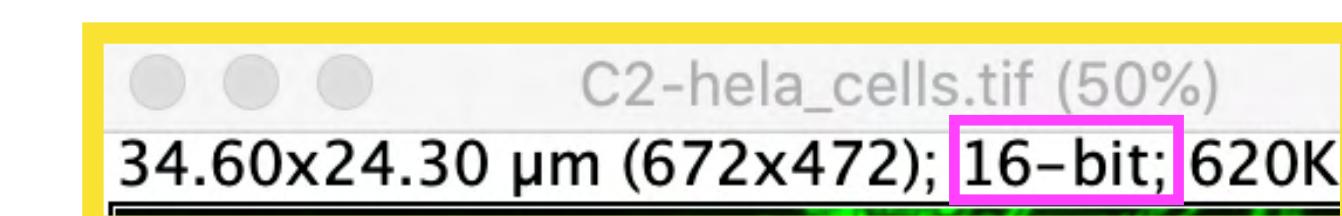


**Image > Show Info...**

**(cmd) + i**

Show the **Metadata stored with the file**

e.g. **bit depth, camera bit depth**

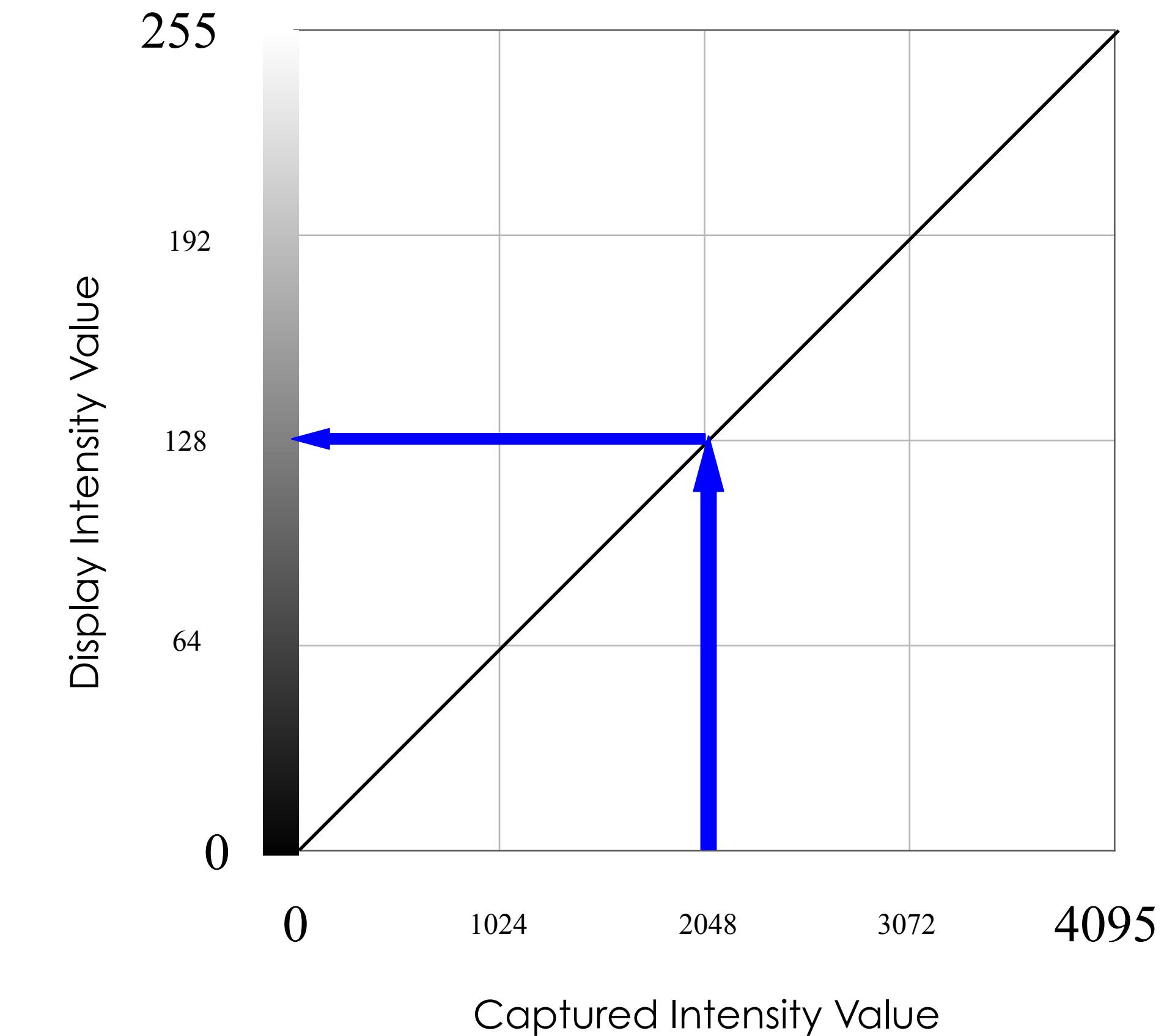
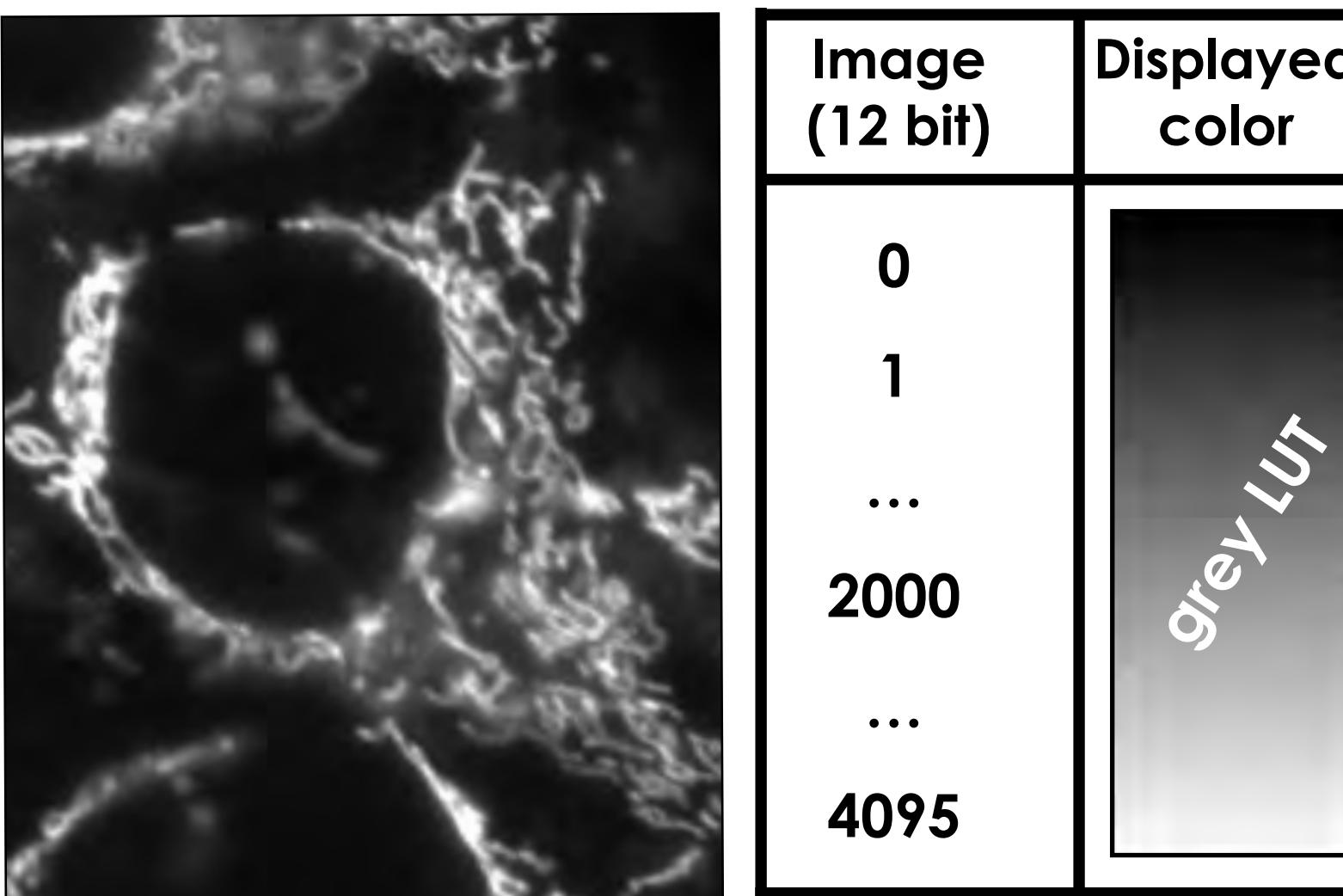
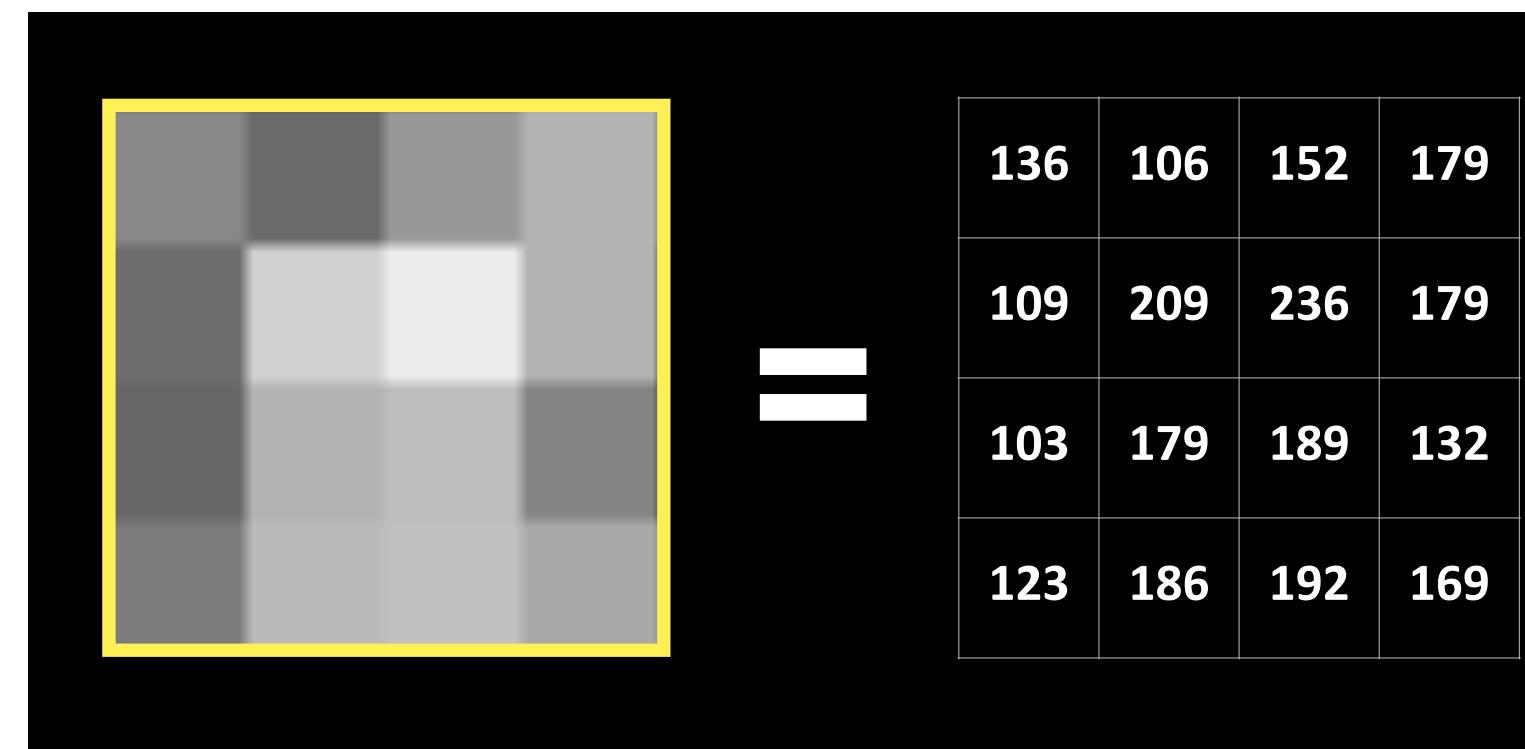


# display your images

# Mapping Image Intensity to Monitor Intensity (LookUp Tables)

**LUT = how the grey values are displayed**

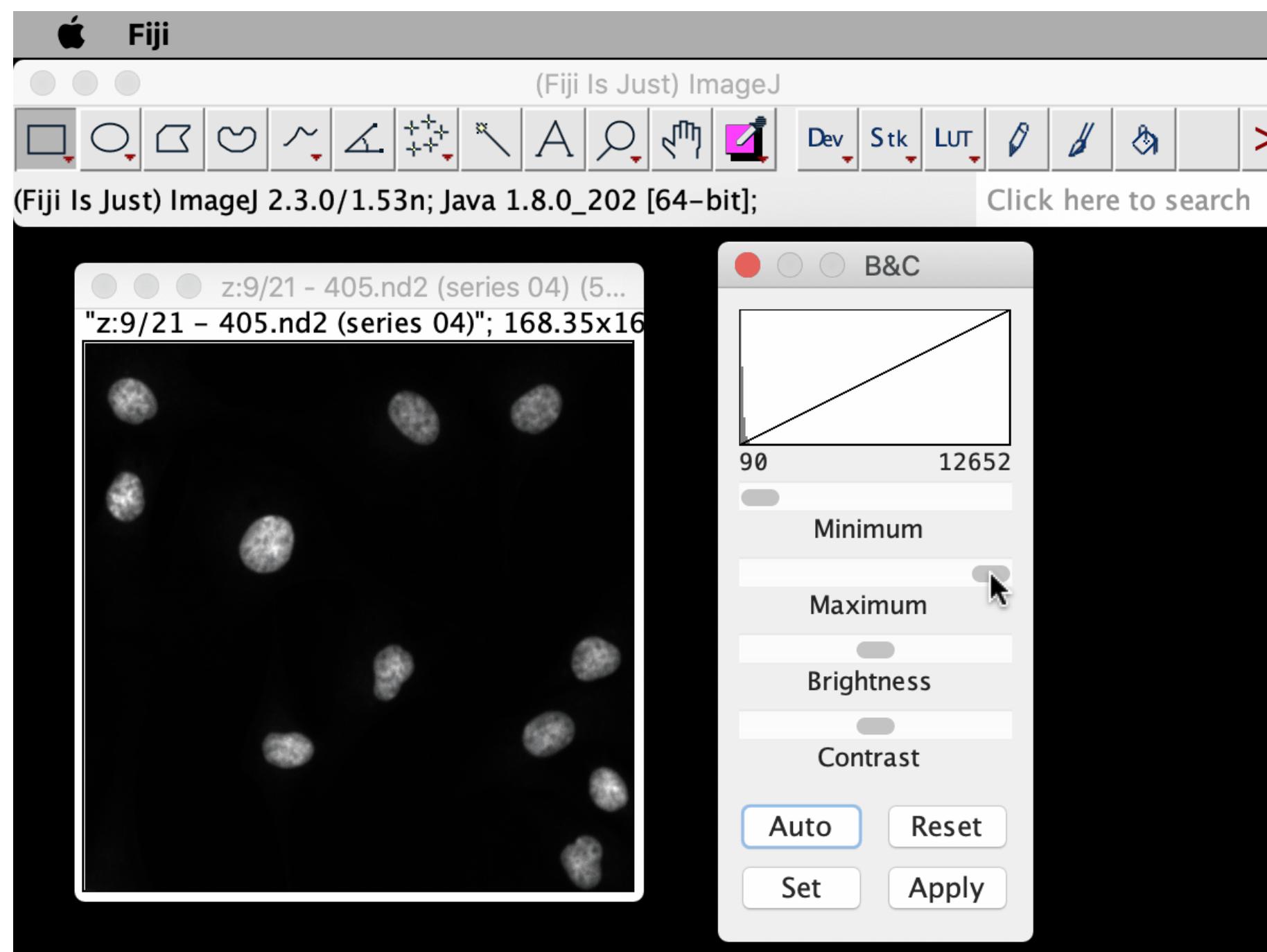
**LUTs do not change the pixel values**



# Display a file: Brightness & Contrast

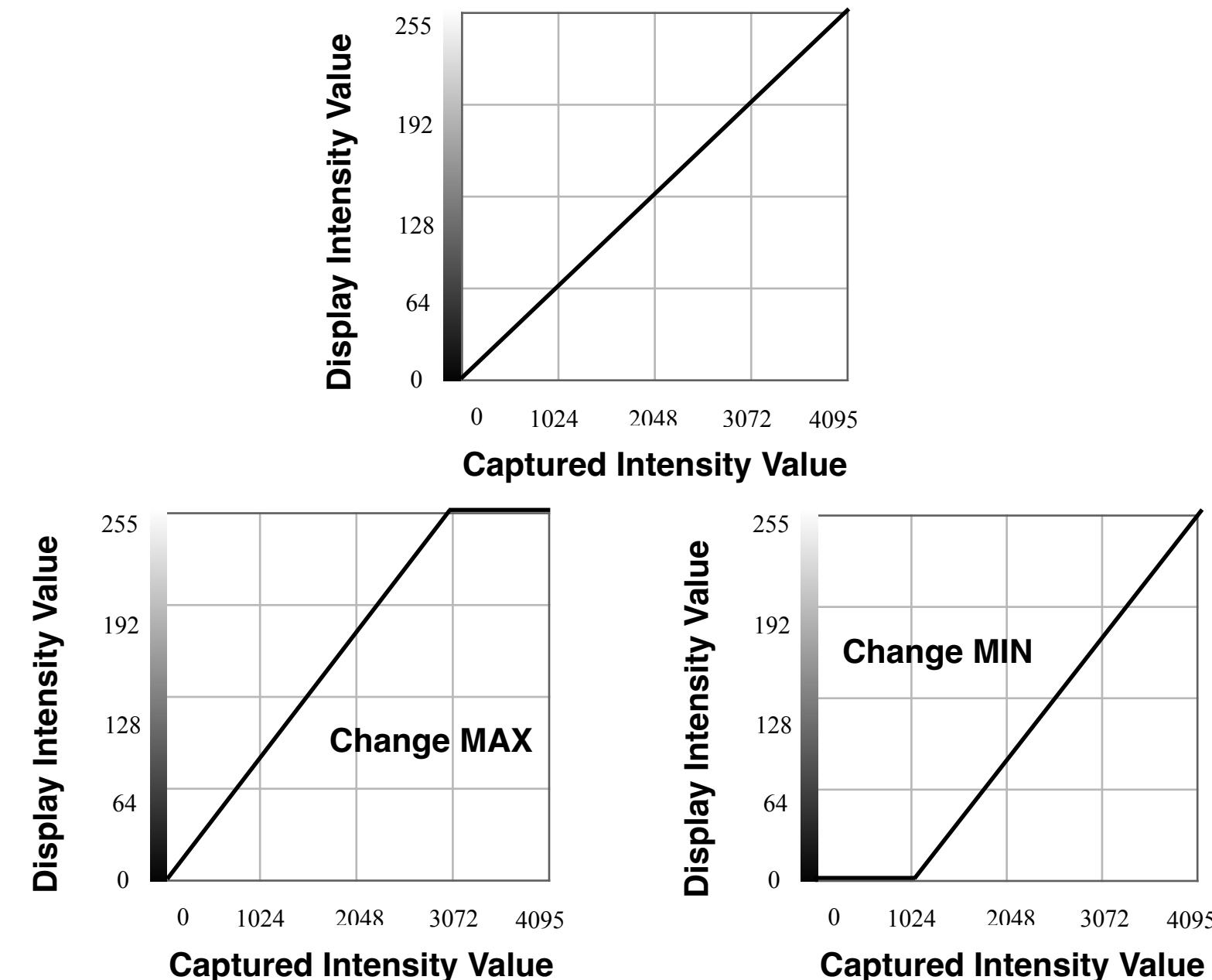
**Image > Adjust > Brightness/Contrast**

*Interactively change the displayed brightness and contrast of the active image.*



**(cmd) + shift + c**

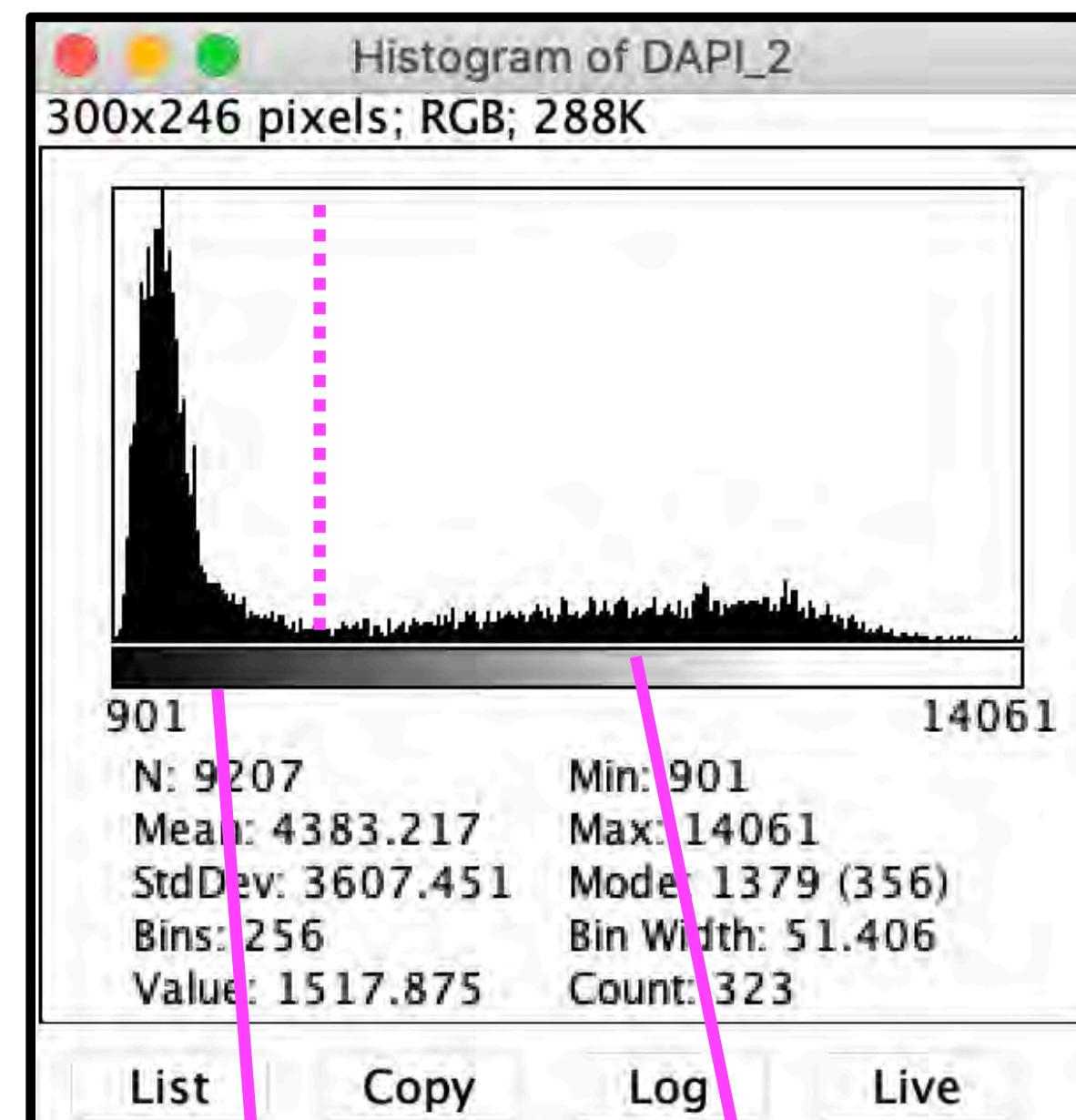
\*You are **NOT changing the pixels values, you are just changing how the image is displayed (unless you click on the “Apply” button).**



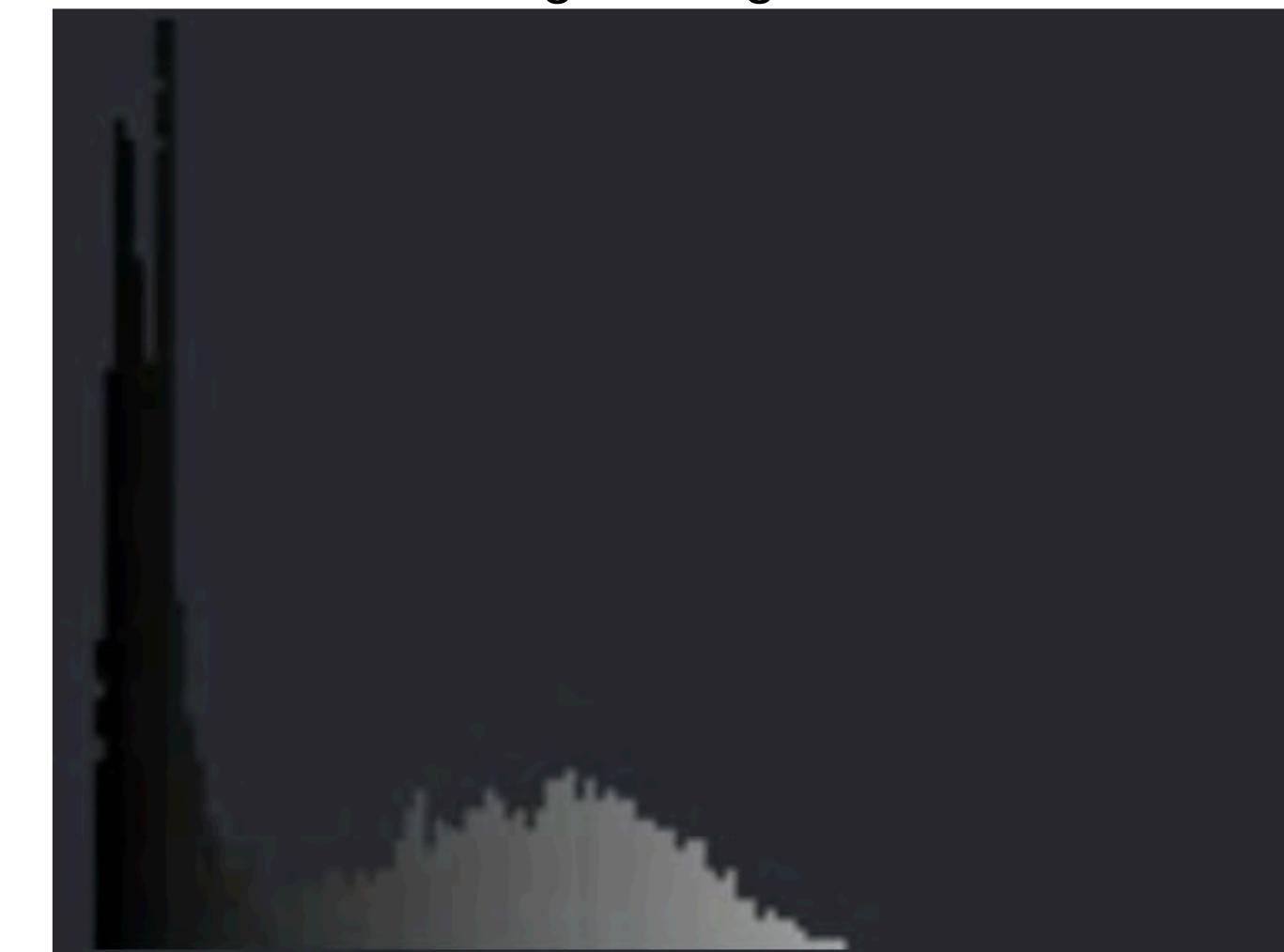
# Histogram

Analyze > Histogram

h



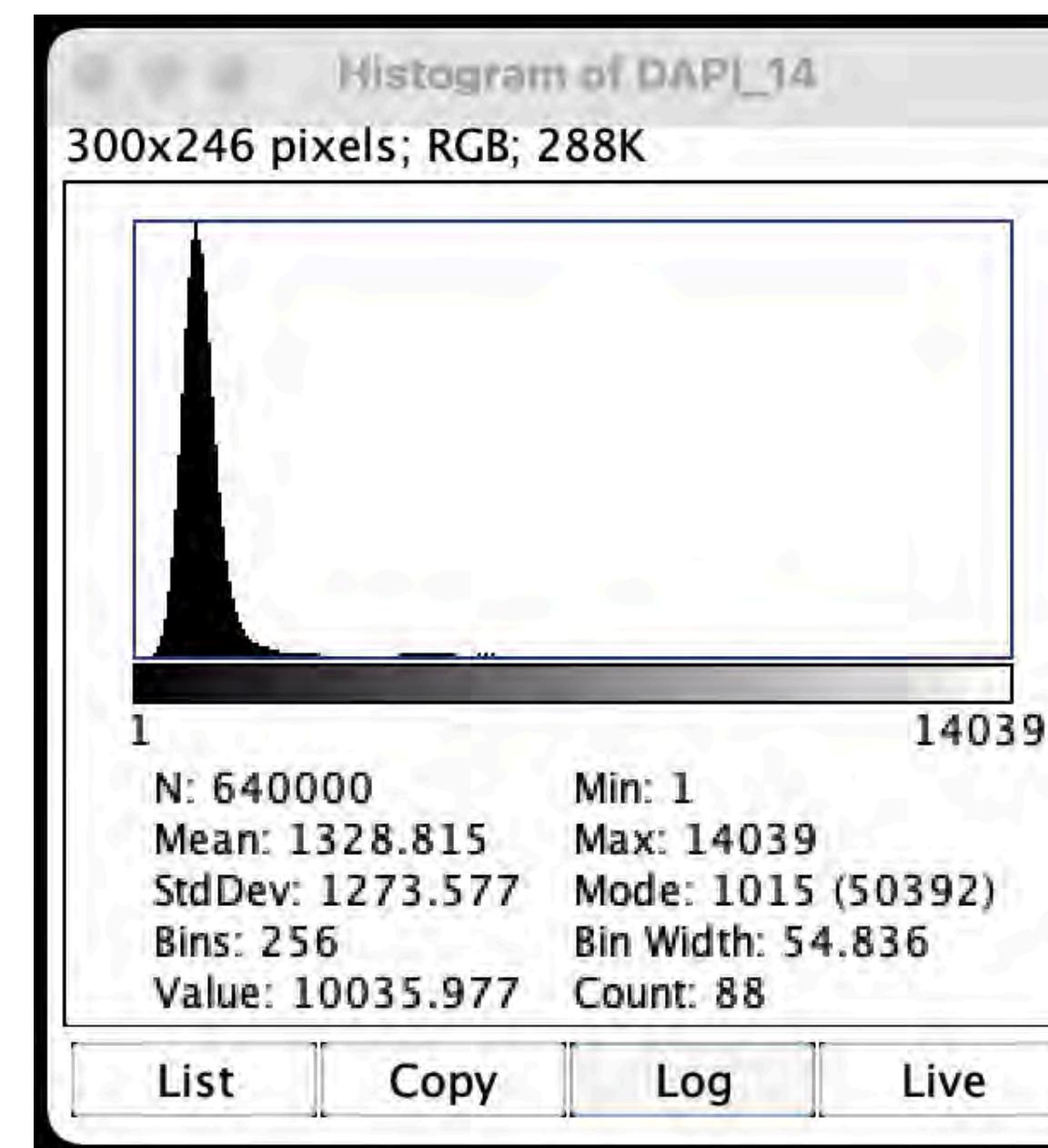
jaehyuk-lee: <https://jaehyuk-lee.com/animated-image-histogram/>



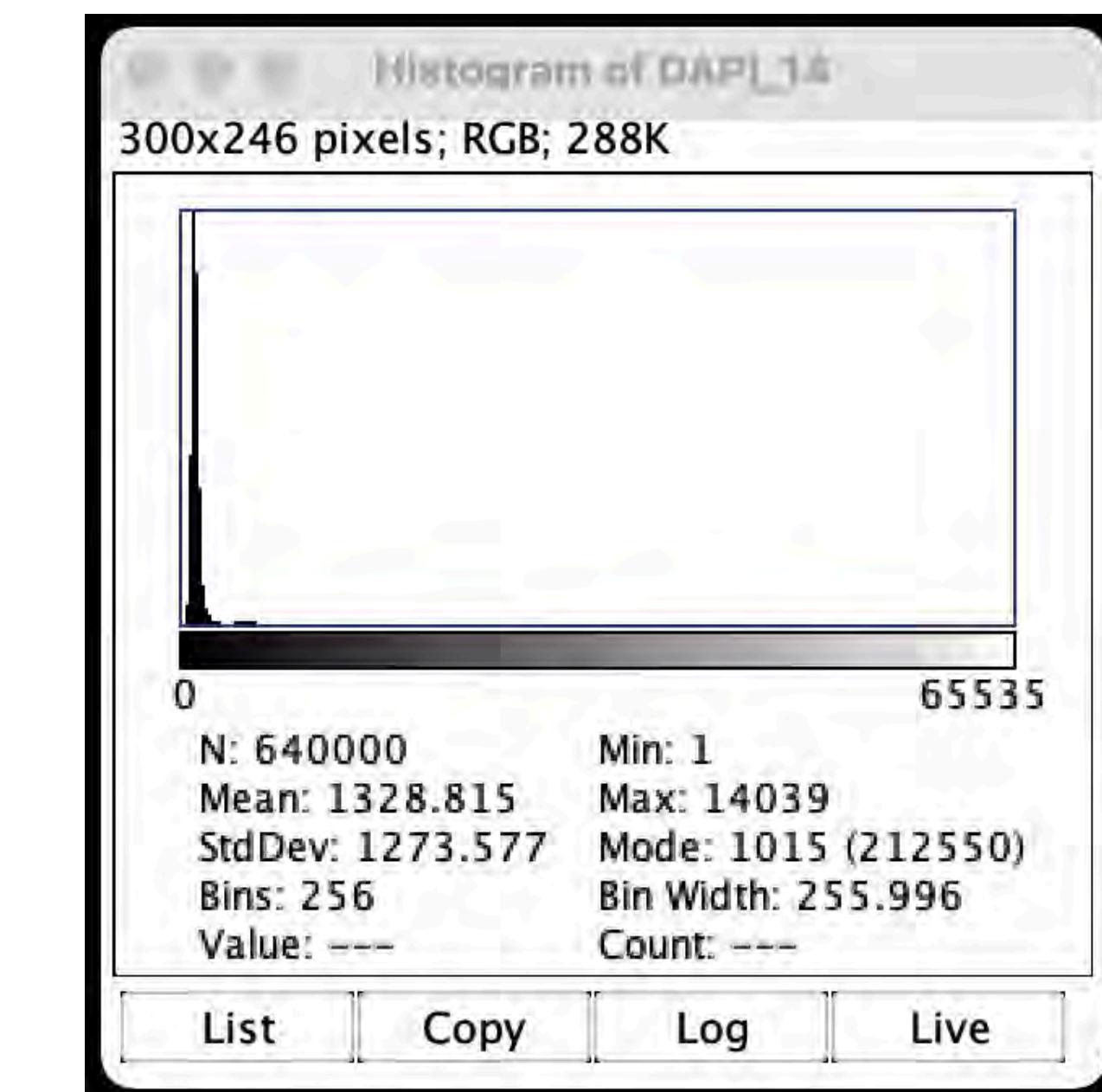
# Histogram

Analyze > Histogram

h



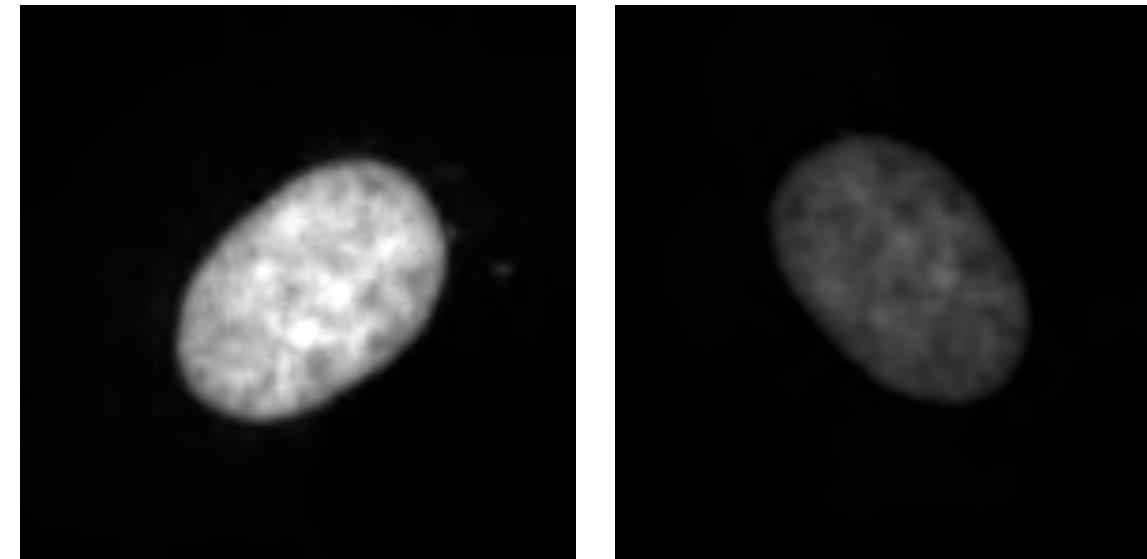
Fiji auto-adjust the range (default option)



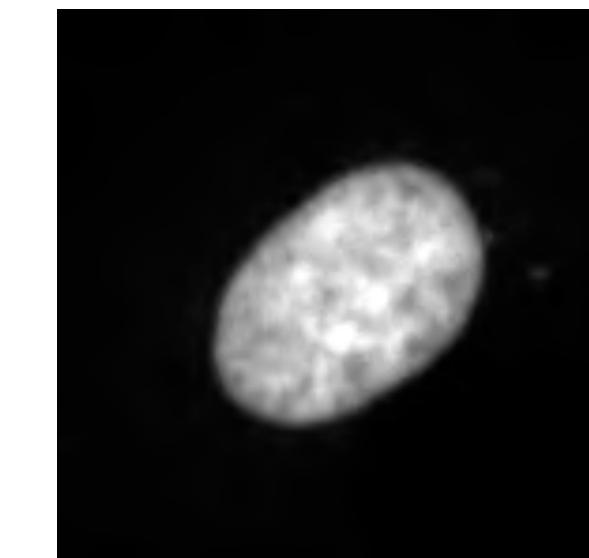
Bit depth

# Display a file: Brightness & Contrast

Which image has more fluorescence?



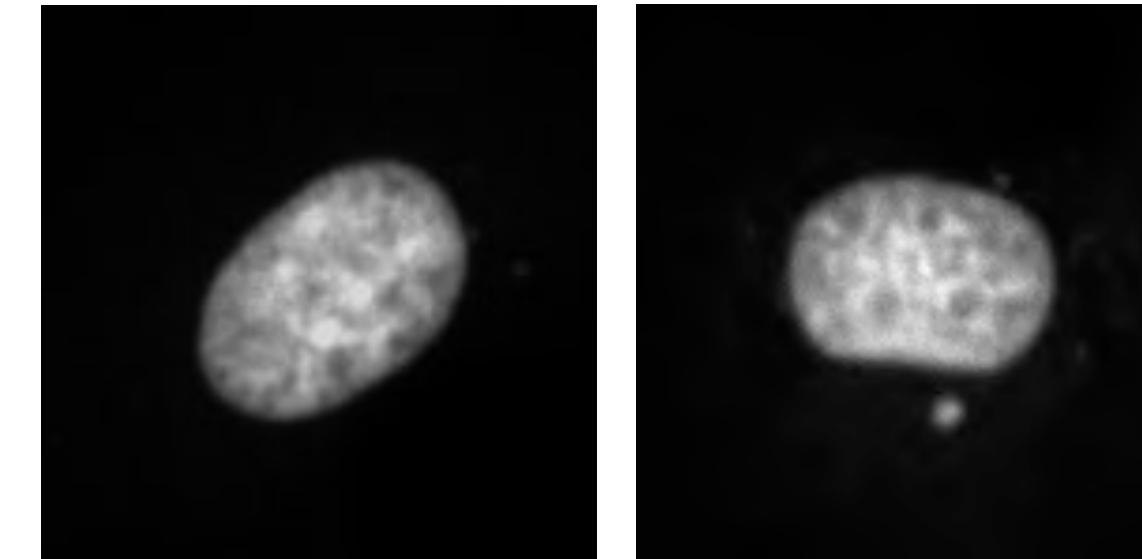
Mean:	<b>4803</b>	<b>4803</b>
Display range:	188- <b>16828</b>	188- <b>45514</b>



Mean:	<b>4803</b>	<b>4803</b>
Display range:	188- <b>16828</b>	188- <b>16828</b>

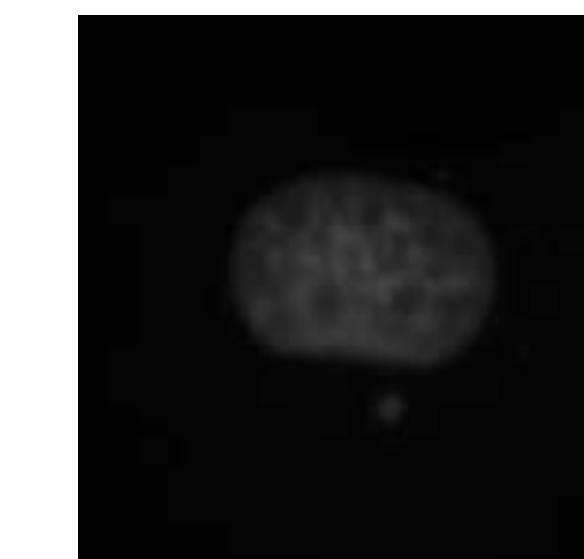
# Display a file: Brightness & Contrast

Which image has more fluorescence?



Mean:	<b>4803</b>	<b>2074</b>
Display range:	188- <b>19540</b>	112- <b>7768</b>

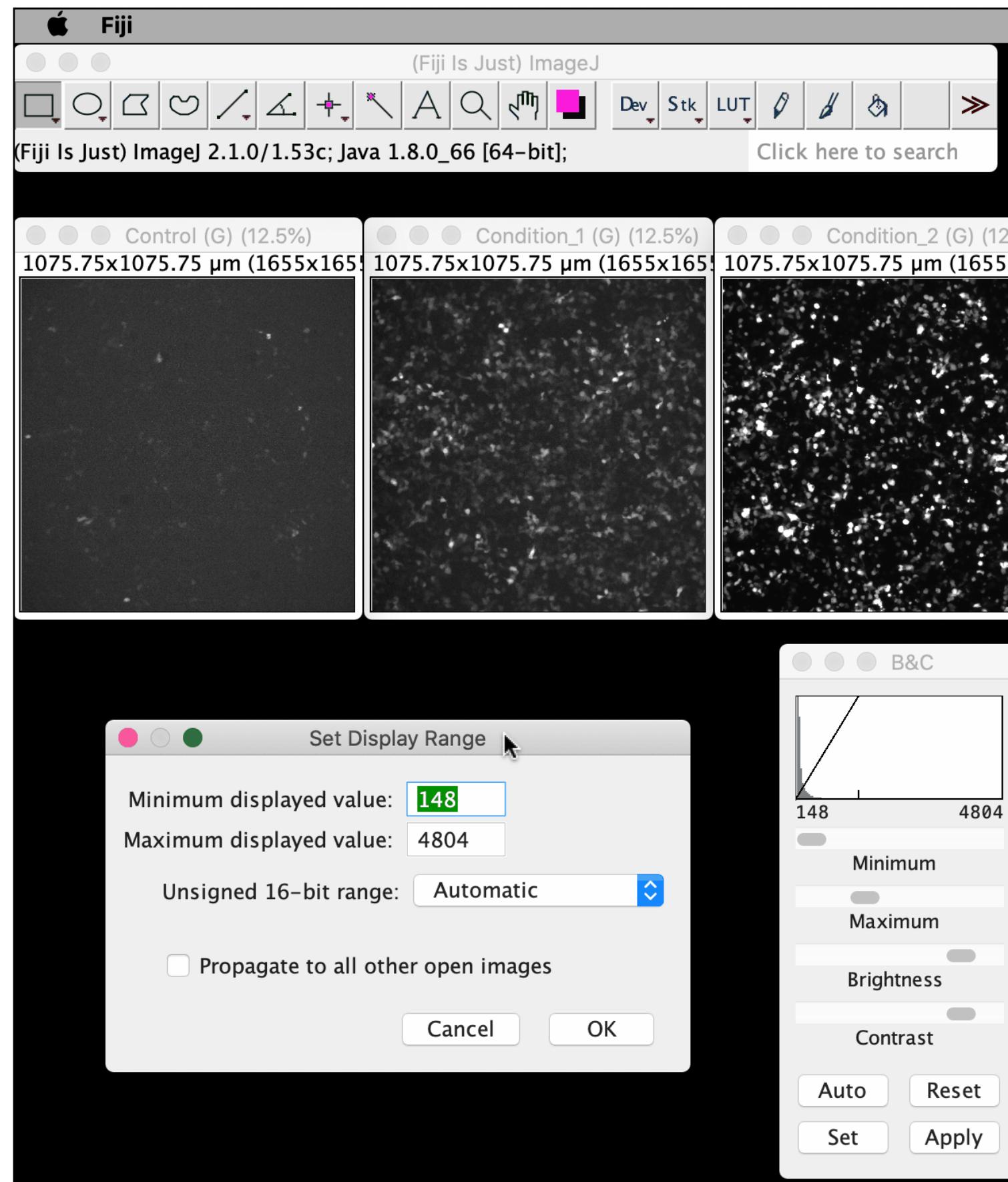
Do NOT trust your eyes,  
rely on numbers!



Mean:	<b>4803</b>	<b>2074</b>
Display range:	188- <b>19540</b>	188- <b>19540</b>

# Display a file: Brightness & Contrast

**“Set” button - visually compare images**



**Use the same acquisition parameters  
(per channel) if you want to compare images!!!  
(e.g. same exposure time, illumination intensity,...)**

Example	Exposure time Condition1	Exposure time Condition2
Channel 1	100 ms	100 ms
Channel 2	200 ms	200 ms

**For a meaningful comparison, you have to extract numbers from your images (analysis). Use the images to support your results.**

**If you save the images as jpeg/png/tiff, they will maintain the display range you set.**



**1.1 image inspection**

**1.2 adjust brightness/contrast of all open images**

# Images and Colors

## Lookup Tables (LUTs)

**LUT = how the grey values are displayed**

**LUTs do not change the pixel values**

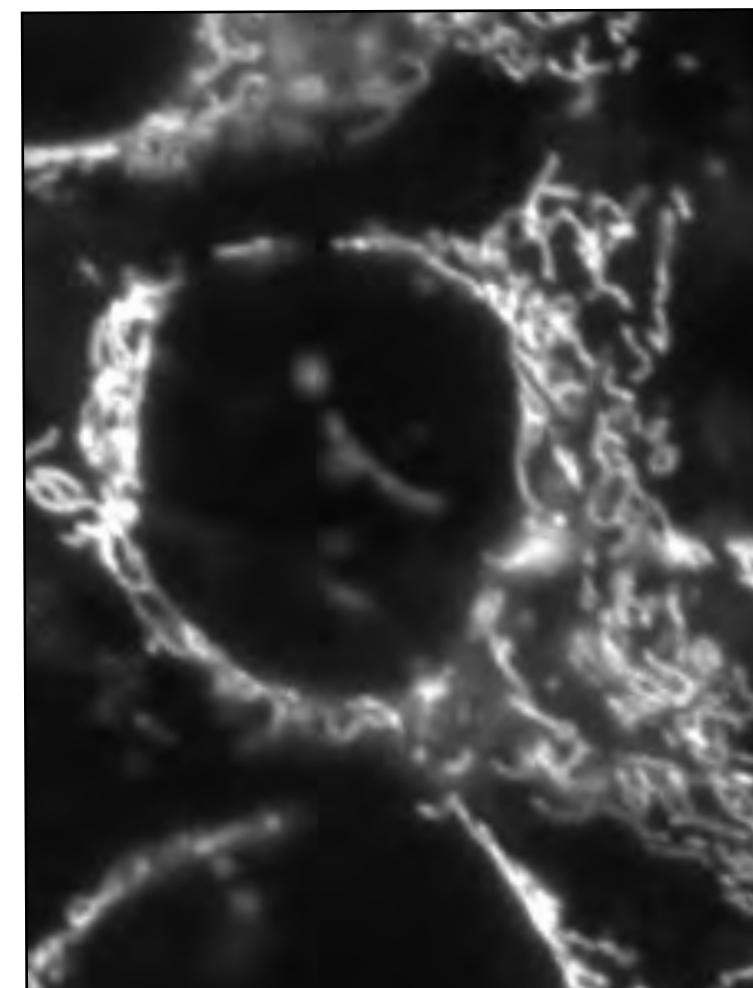


Image (8 bit)	Displayed color
0	
1	
...	
100	
...	
255	

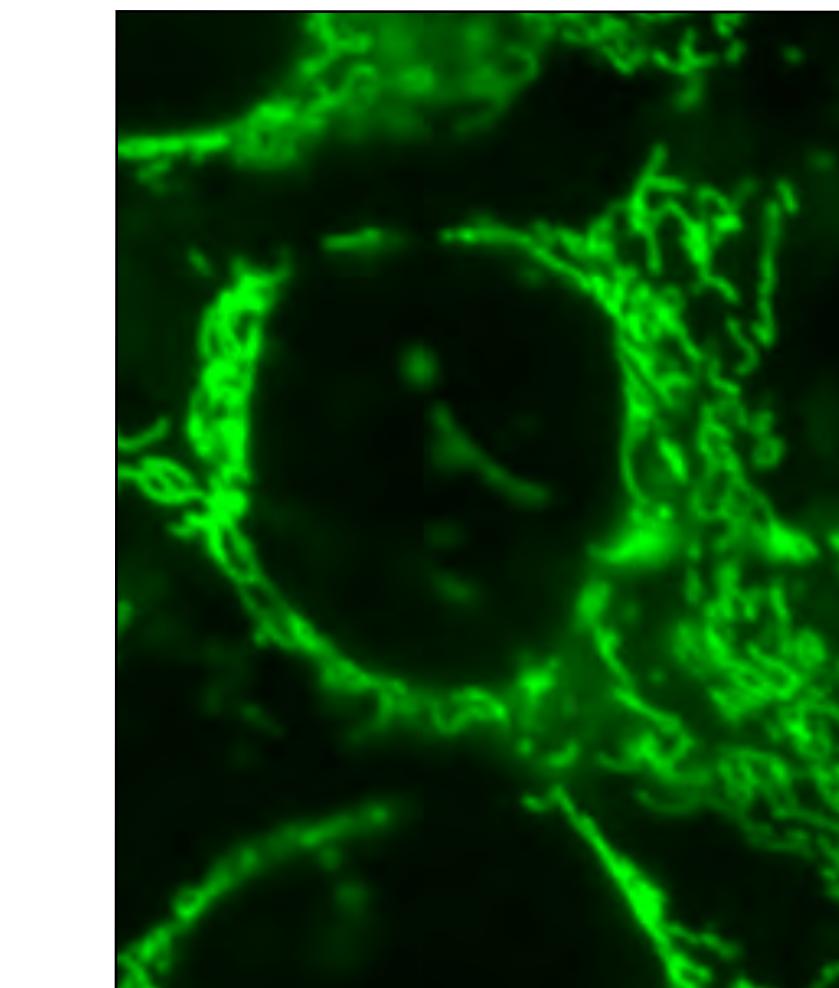
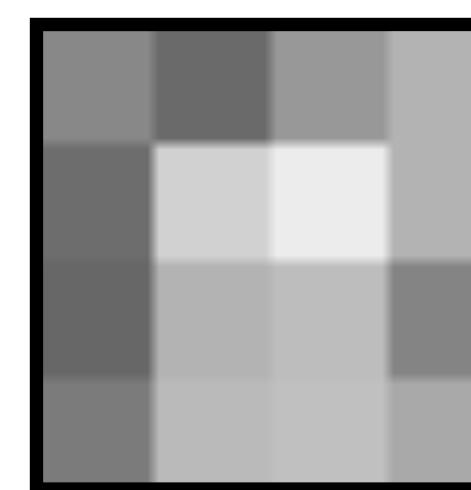
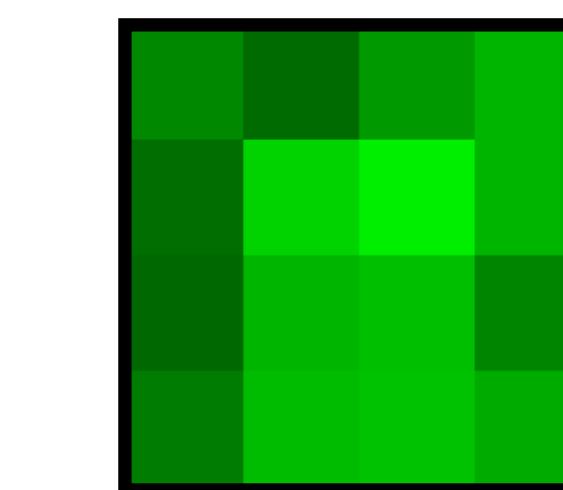


Image (8 bit)	Displayed color
0	
1	
...	
100	
...	
255	



=

136	106	152	179
109	209	236	179
103	179	189	132
123	186	192	169

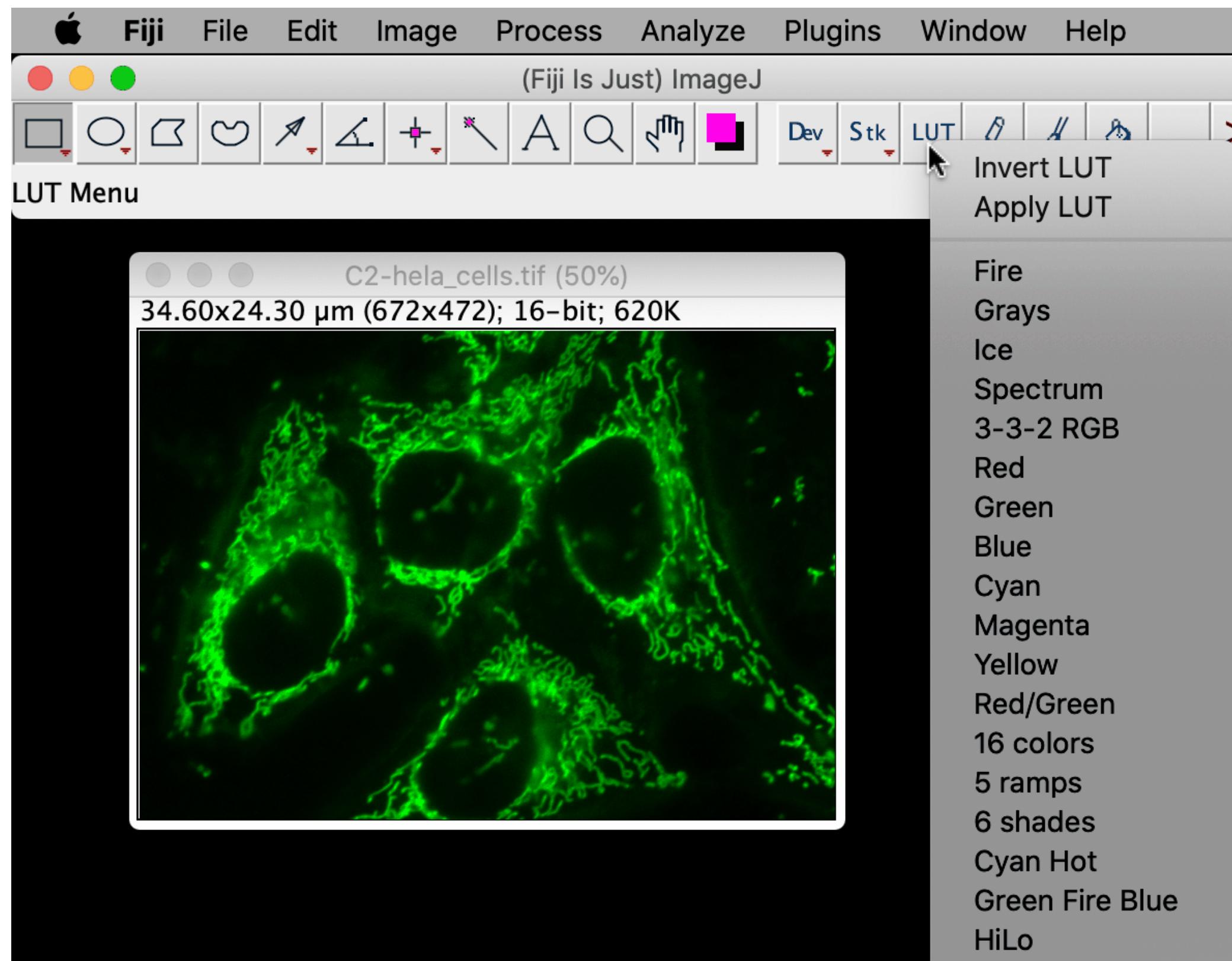


=

136	106	152	179
109	209	236	179
103	179	189	132
123	186	192	169

# Images and Colors

## Lookup Tables (LUTs)



**Image > Lookup Tables**

or

**"LUT" menu in Startup Tools Bar**



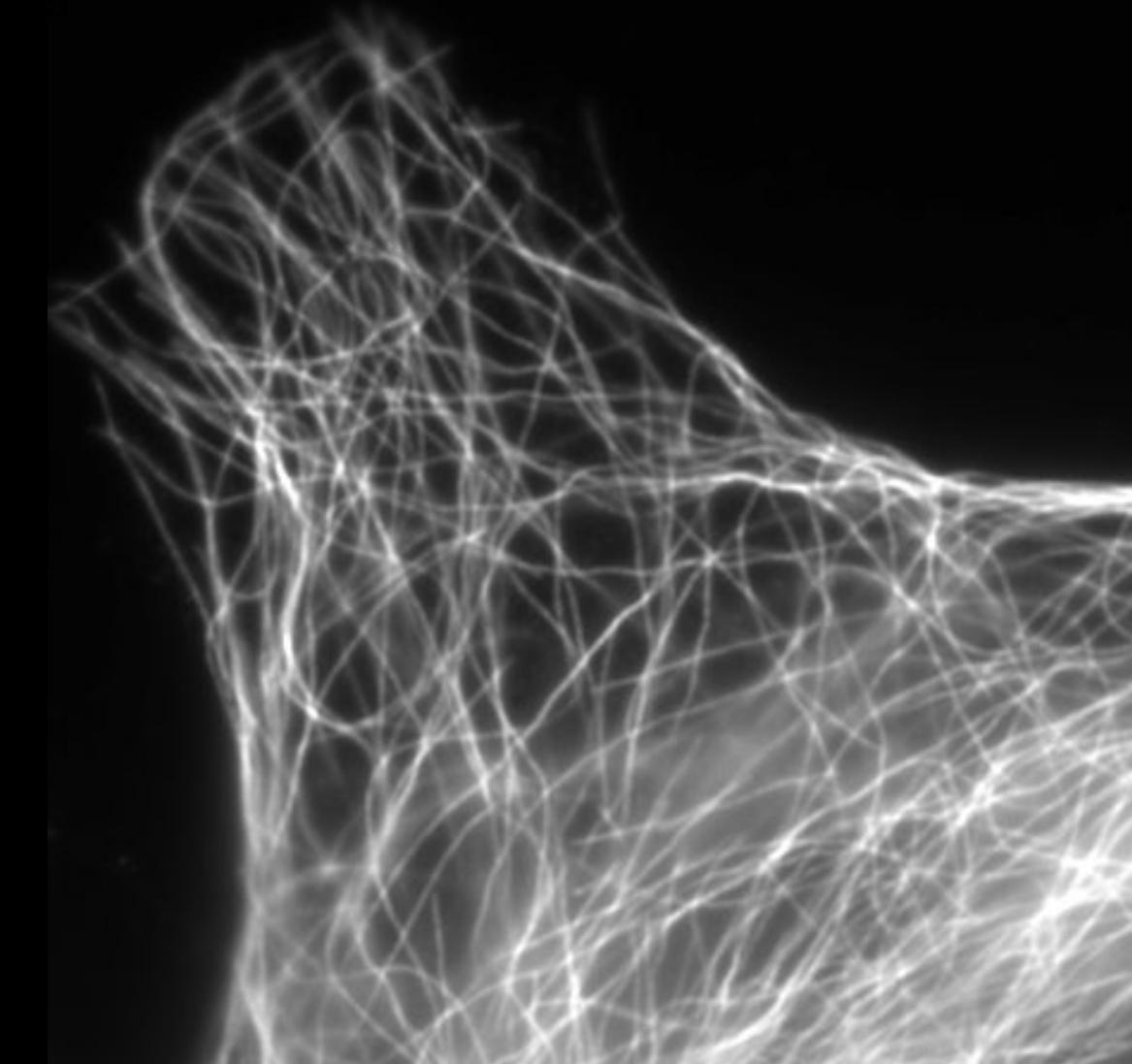
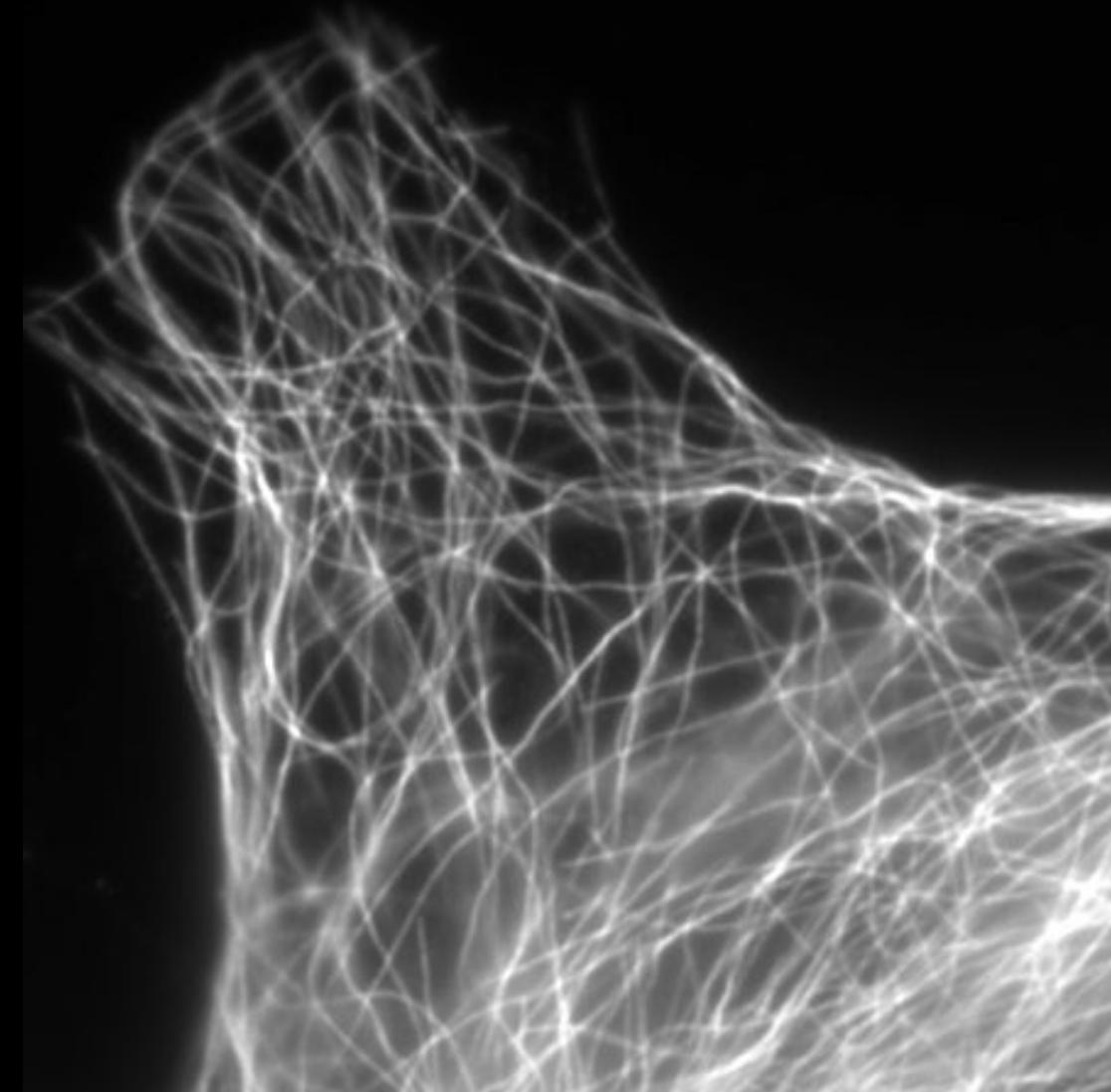
Examples of LUTs

\*Image > Color > Display LUTs - to display default LUTs available

# Images and Colors in Fiji

*Choose the right LUT*

Which is brighter?



The human eye evaluates intensity best in grayscale

If you are imaging for example a blue fluorophore, you  
are NOT FORCED to display it in blue!

# Images and Colors in Fiji

*Choose the right LUT*

Color blind people don't distinguish some colors

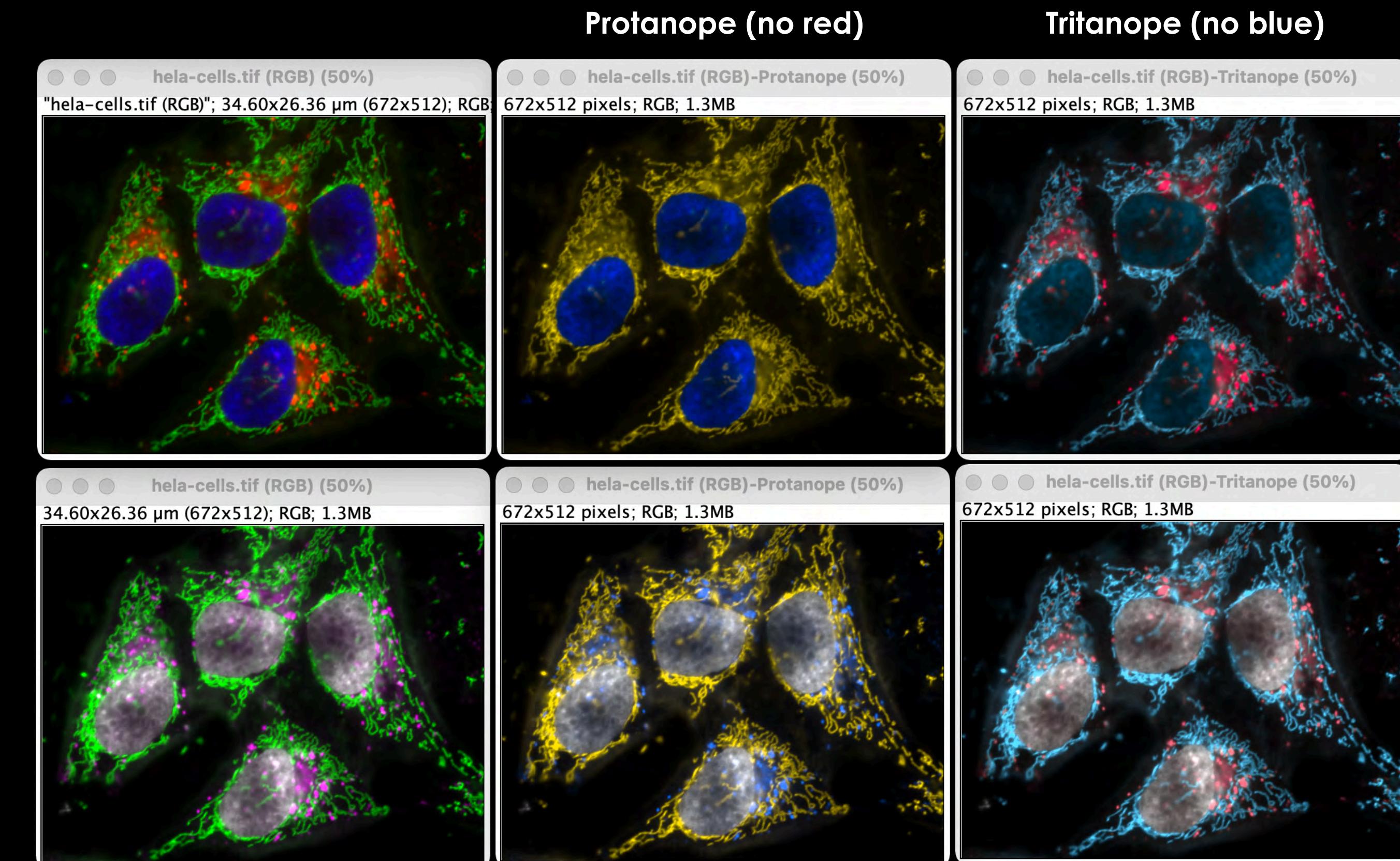


"If a submitted manuscript happens to go to three male reviewers of Northern European descent, the chance that at least one will be color blind is 22 percent"

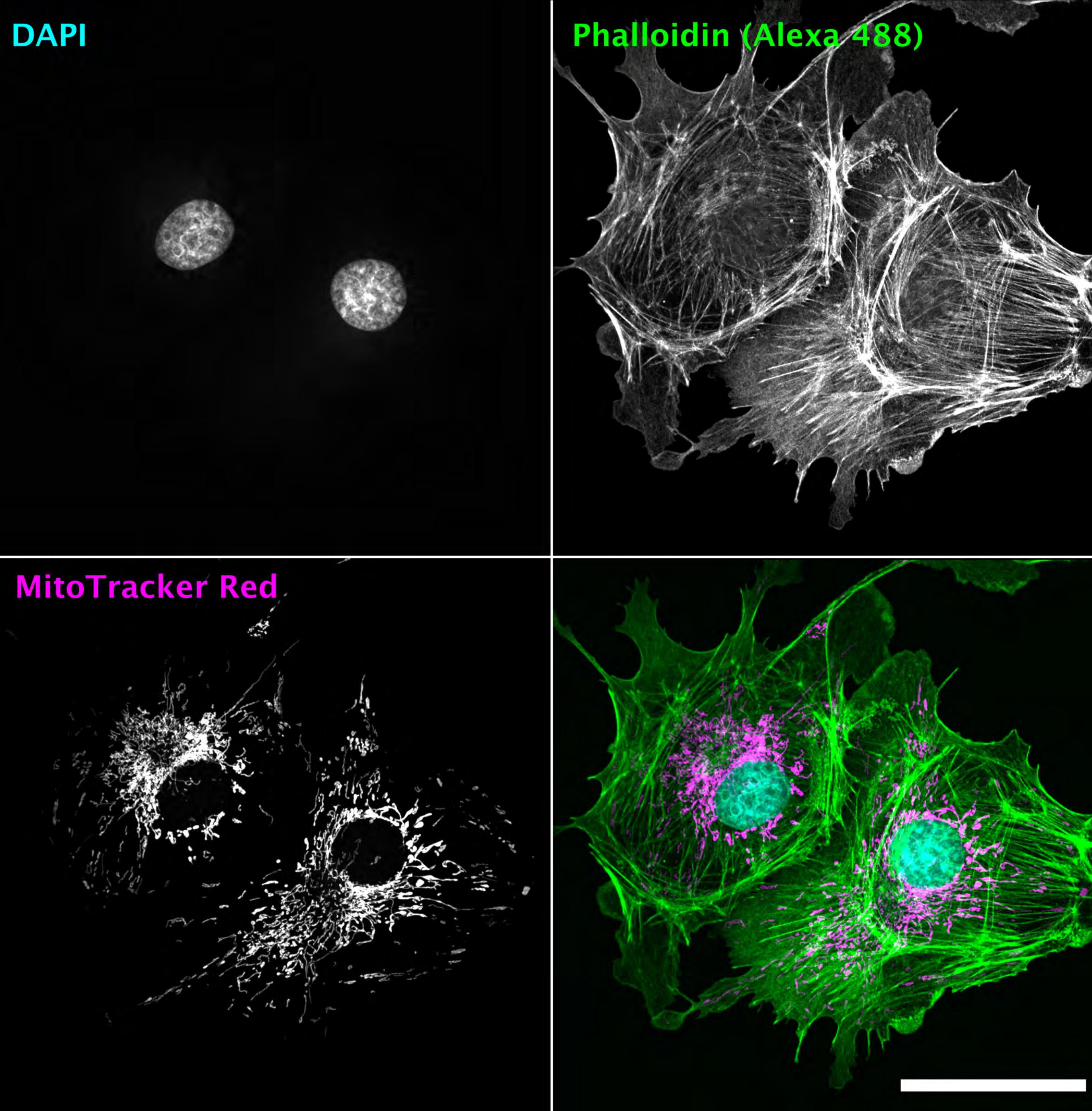
# Images and Colors in Fiji

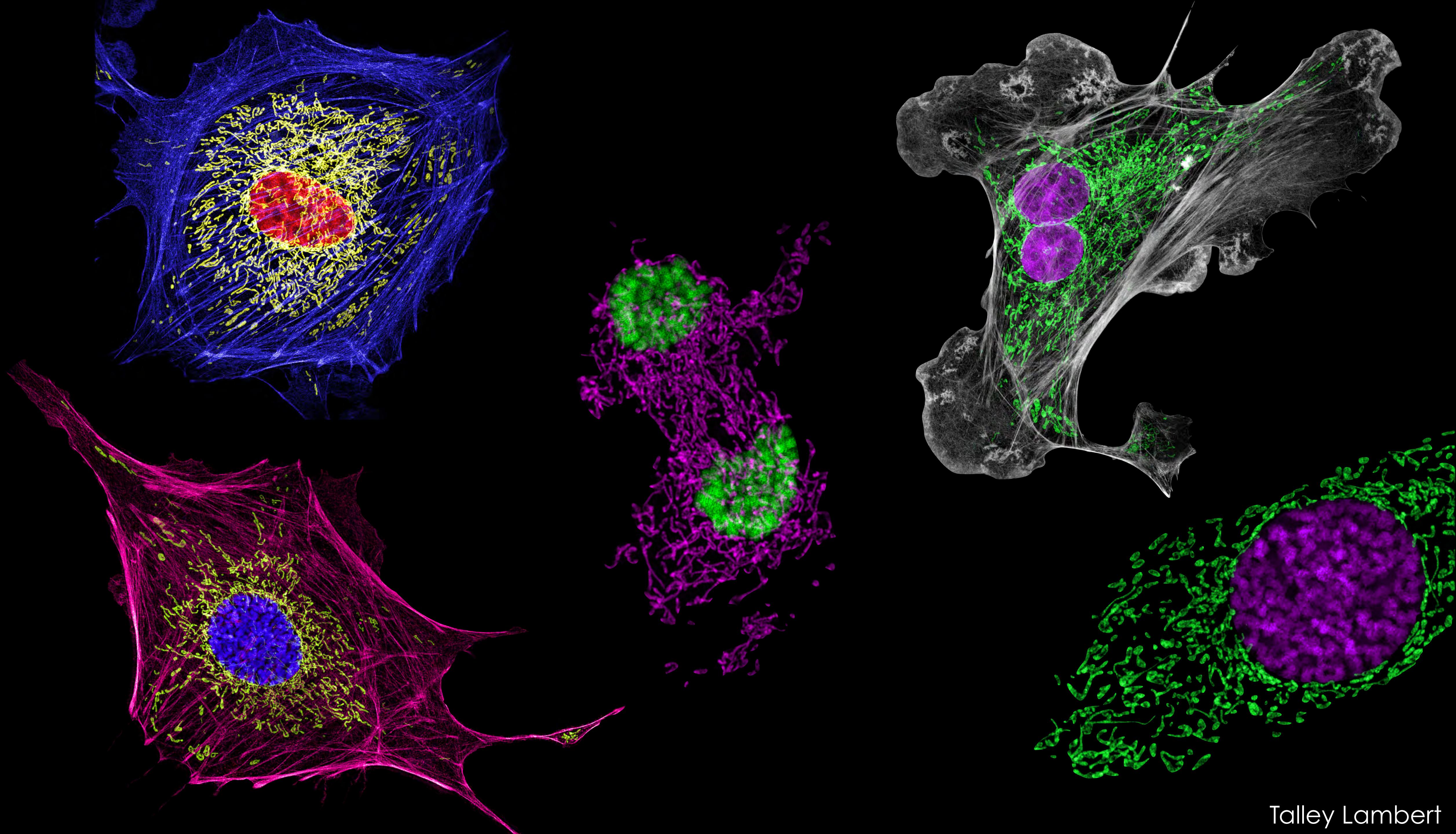
*Choose the right LUT*

Color blind people don't distinguish some colors



*Image > Color > Dichromacy or Image > Color > Simulate Color Blindness*

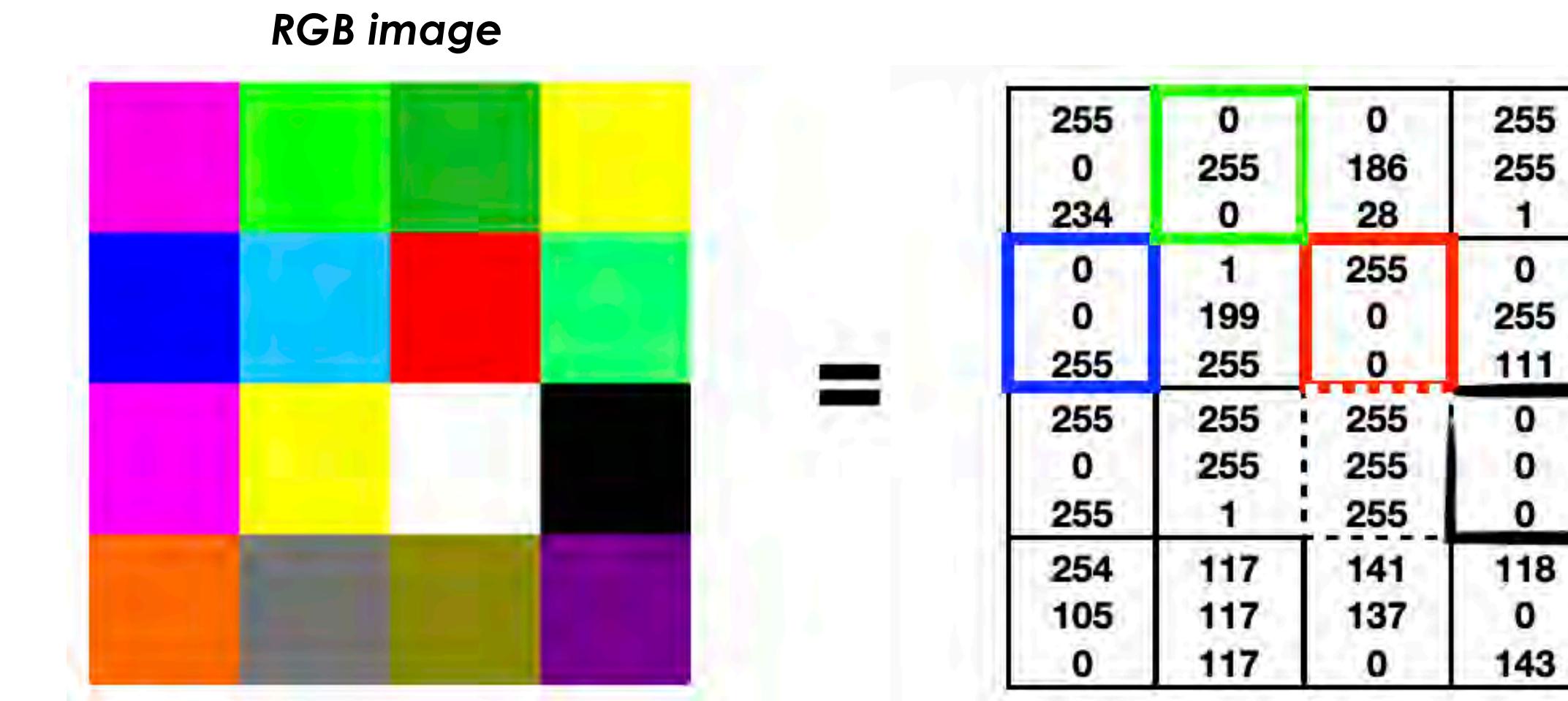




Talley Lambert

# RGB Images (still matrix of numbers)

LUTs **cannot** be applied to RGB Images



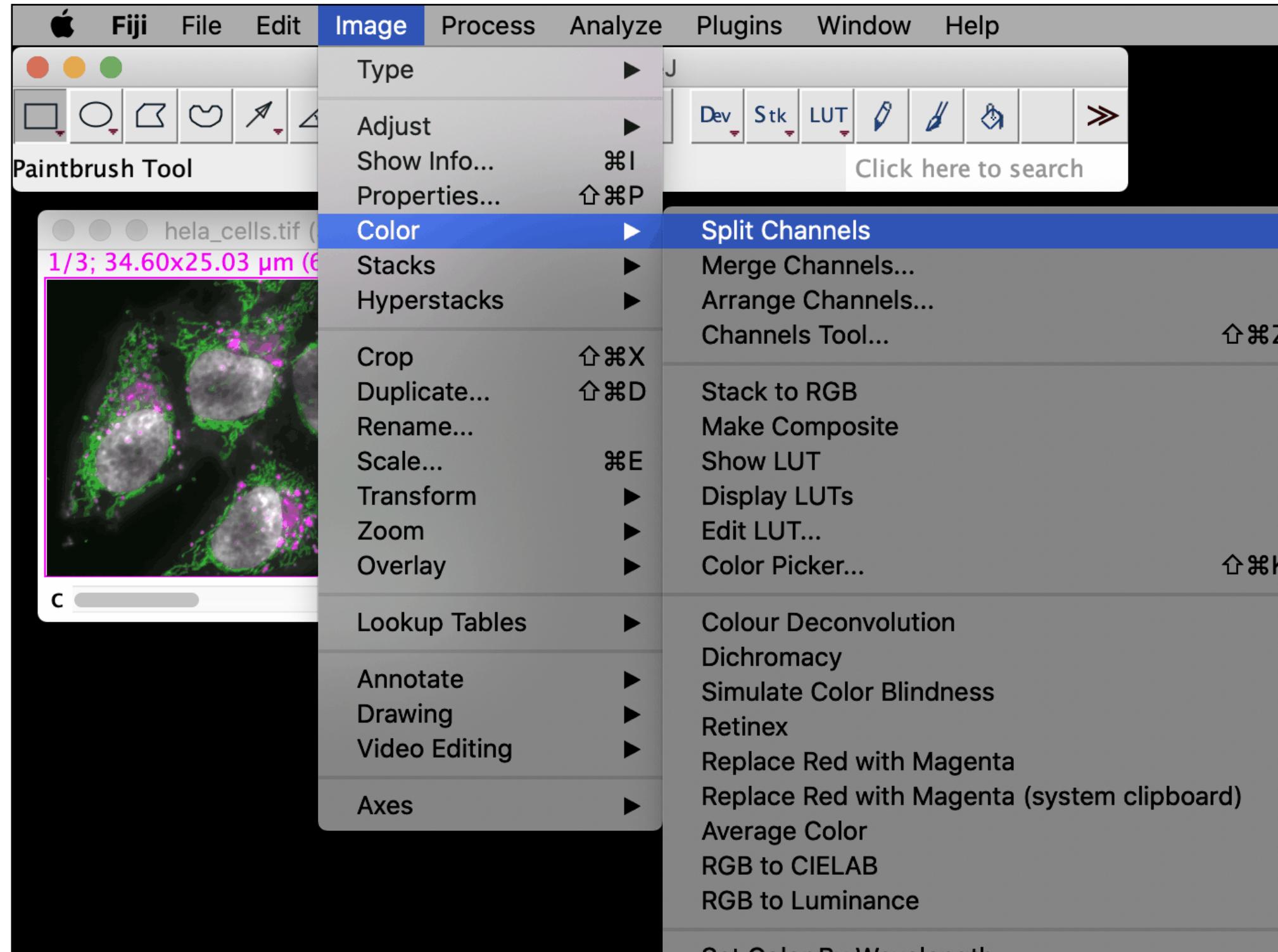
**RGB Color** image (e.g. jpeg, png) = **R**ed + **G**reen + **B**lue

**RGB Color** image = **8 bit Red, 8 bit Green, 8 bit Blue** = **R (0-255), G (0-255), B (0-255)**

**Image > Type > RGB Color      or      Save As...png, jpeg**

# Images and Colors in Fiji

## The Color menu



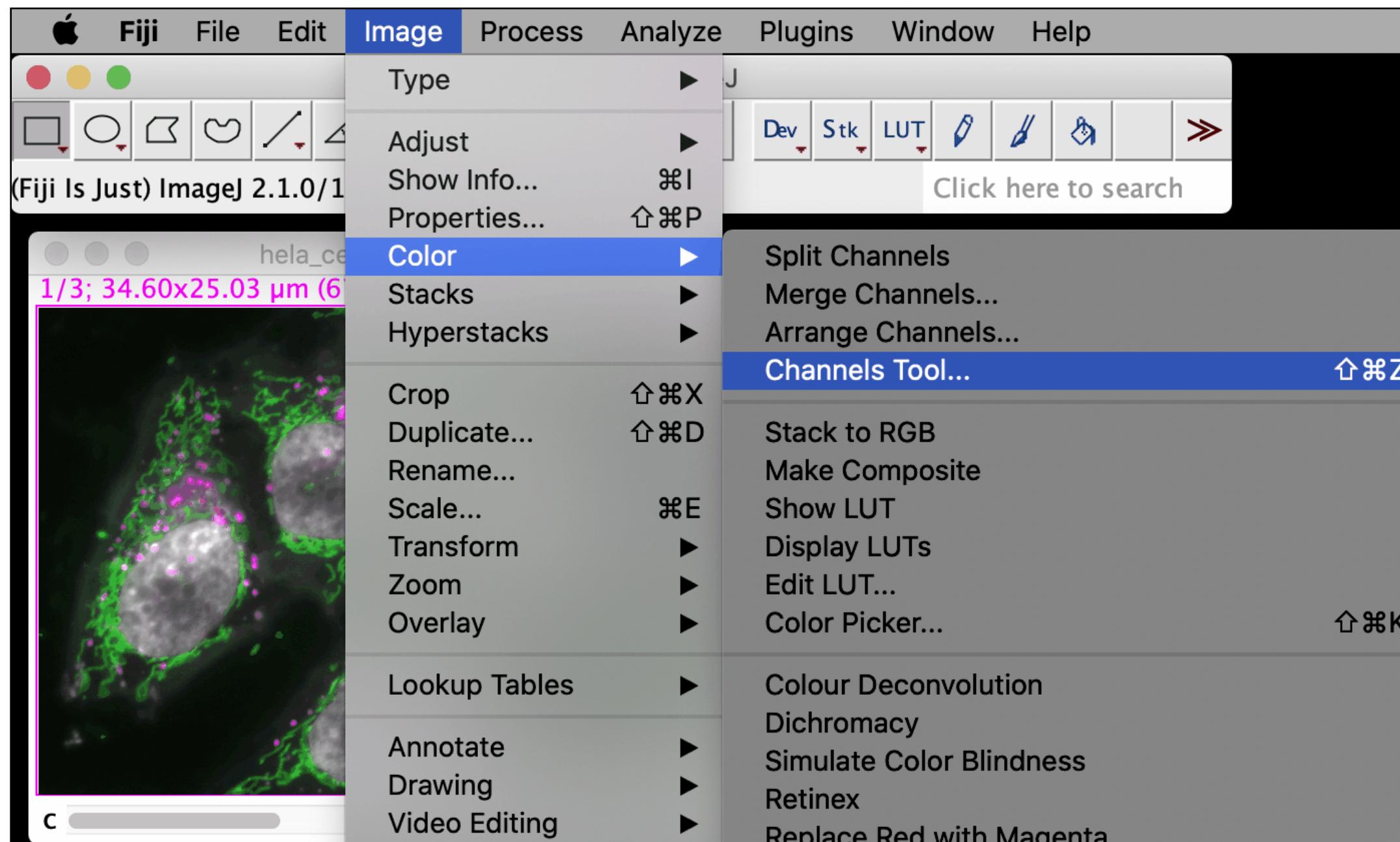
**Image > Color > Split Channels**

**Image > Color > Merge Channels**

**Split or Merge the channels of the active image**

# Images and Colors in Fiji

## The Color menu



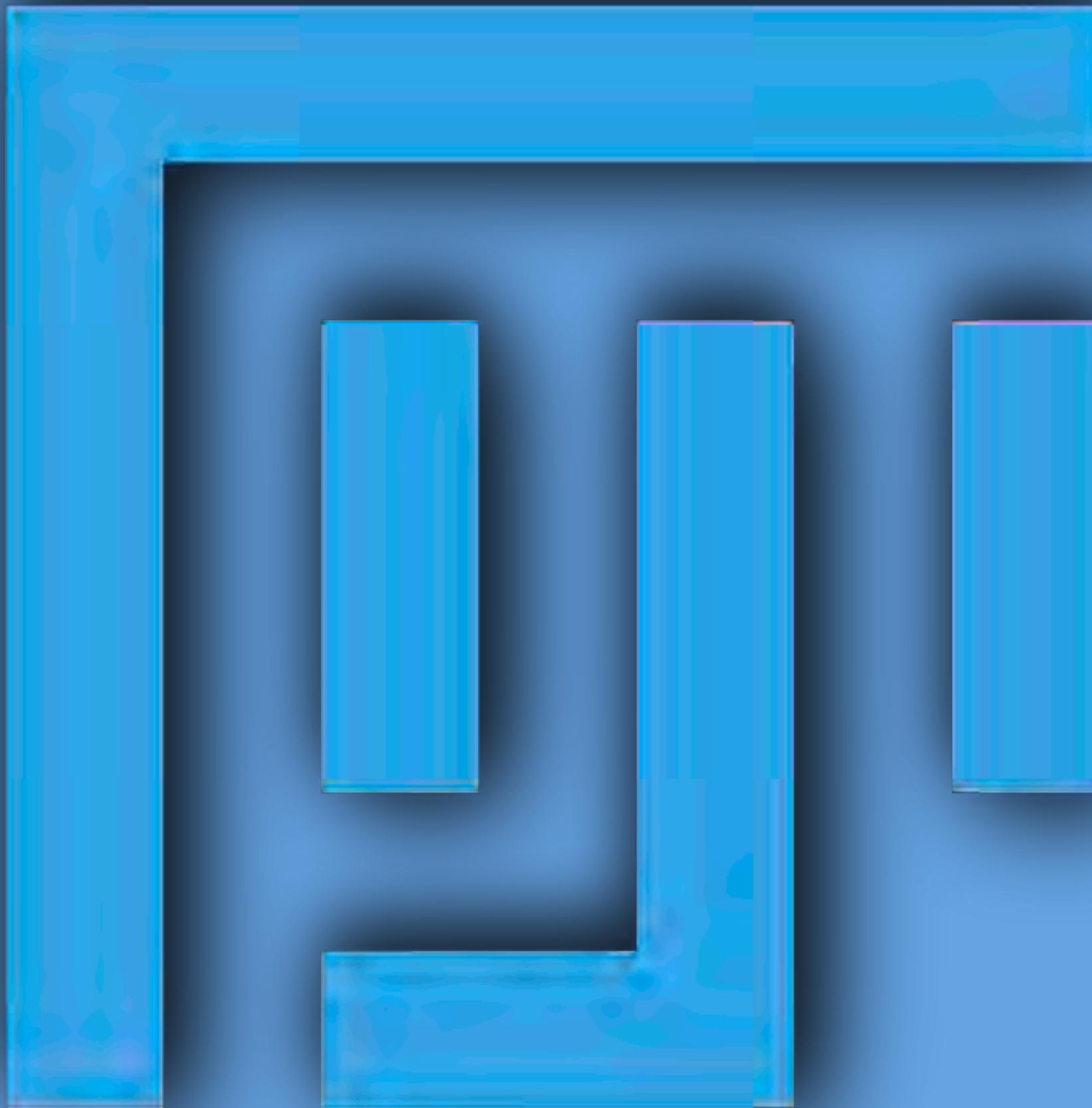
**Image > Color > Channels tools**

**(cmd) + shift + z**

*Interactive visualization/handling of each image channel.*

*Choose how to display your multicolour image: “**Color**” mode (single channel) vs “**Composite**” mode (overlay)*

# Segmentation with pixel based classifier—exercises



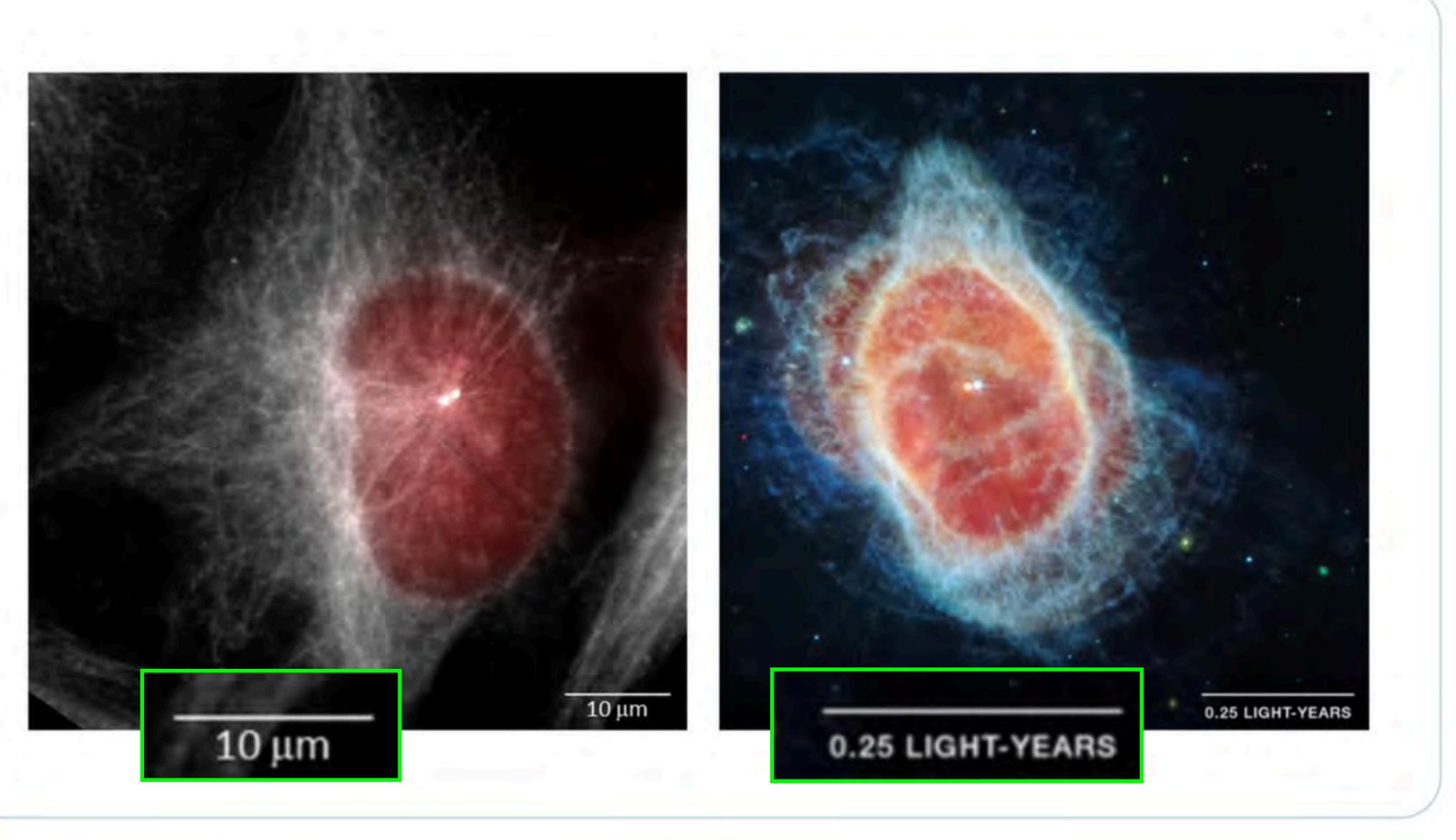
**2.1 composite images - splitting and merging**

# Pixel Size

# Scale Bar and Pixel Size

 Laurence Haren  
@HarenLaurence ...

never forget the scale bar! [@StearnsLab](#)  
when biology meets astronomy: cell vs nebula,  
centrosome vs dying star! [@EtienneKlein](#)



10  $\mu\text{m}$

0.25 LIGHT-YEARS

# Pixel Size and Scale Bar

If you want to make some physical measurements of your sample (length, size, ...),  
you need to know the **pixel size** of your **image**.

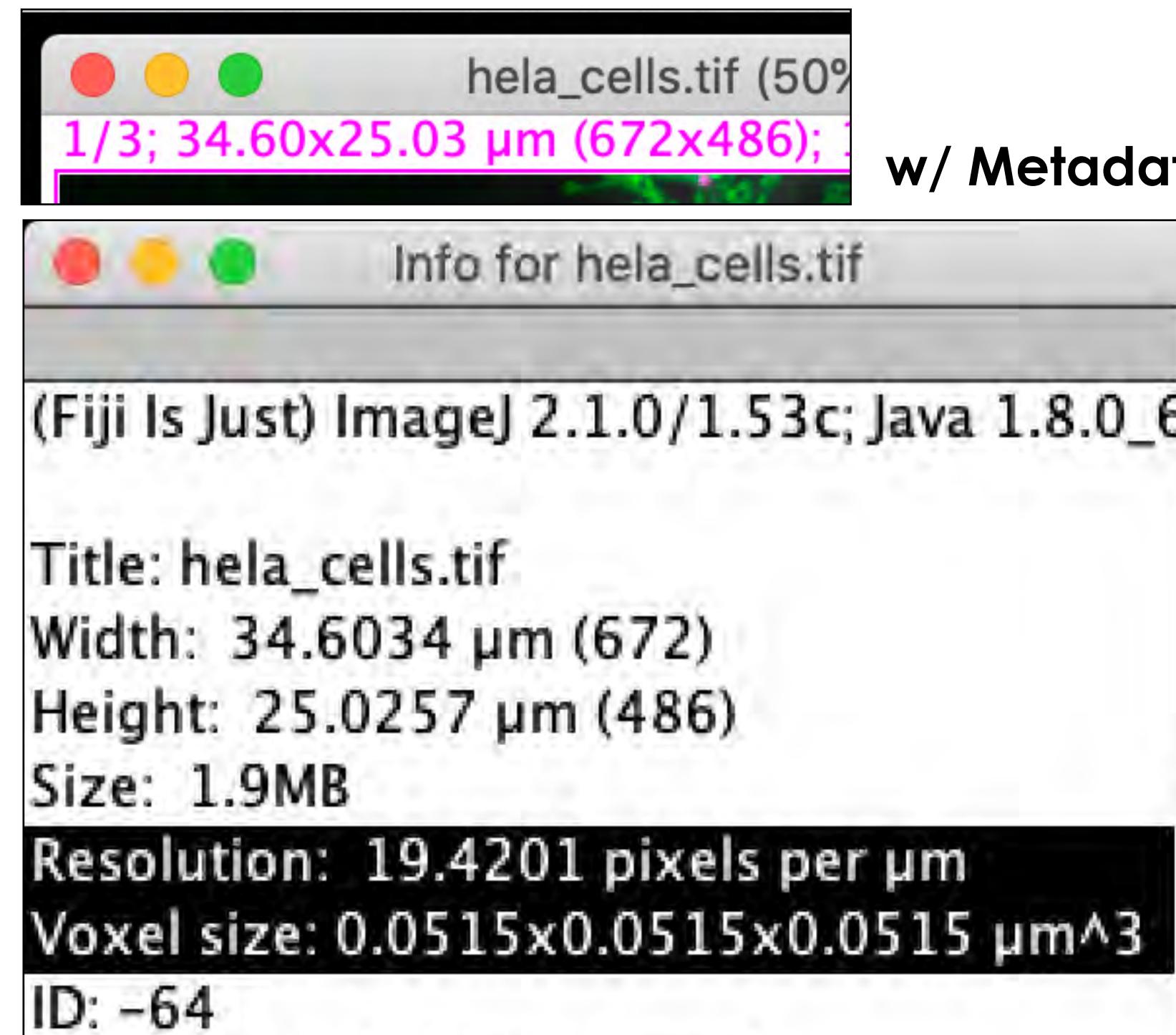


Image > Show Info...

(cmd) + i

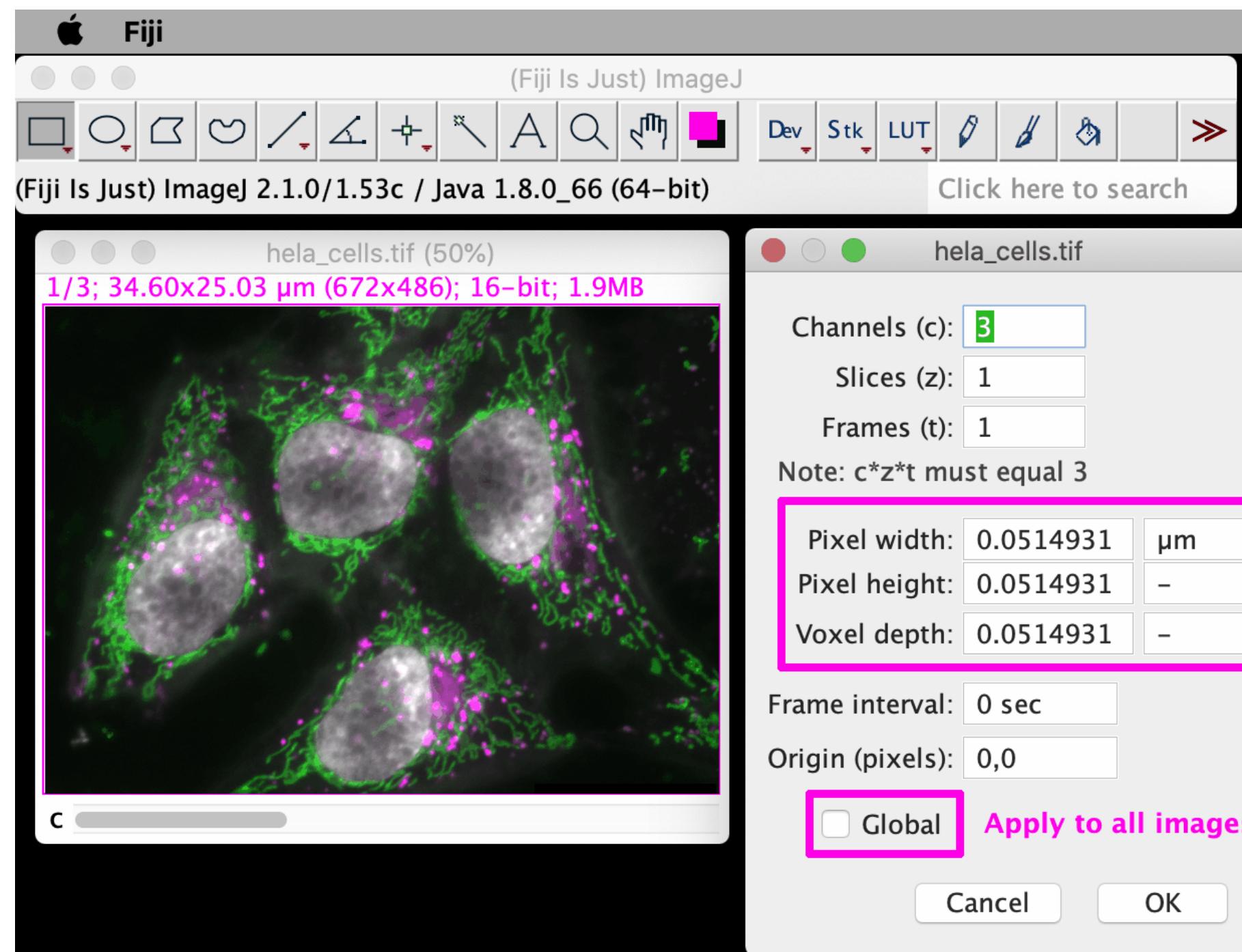
Show the **Metadata stored with the file**.

Pixel width (Pixel x) = 0.0515 µm  
Pixel height (Pixel y) = 0.0515 µm  
Voxel depth (Pixel z) = 0.0515 µm

1 µm = 19.4201 pixel (xy)  
(1 µm / 0.0515 µm)

# Pixel Size and Scale Bar

If you want to make some physical measurements of your sample (length, size, ...),  
you need to know the **pixel size** of your **image**.



**Image > Properties...**

**(cmd) + shift + p**

Open the **image properties** windows  
containing pixel size properties (and  
multi-dimensional properties)

You can **set/read** the image  
pixel size in xyz.

# Scale Bar and Pixel Size

What if the pixel size is not stored in the metadata?

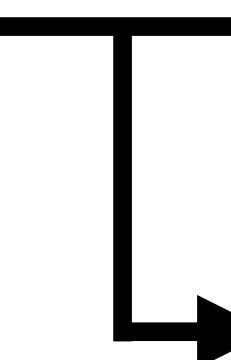
If you know the **magnification** and the **camera** you used for the acquisition, you can estimate the image pixel size.

image pixel size = camera pixel size/magnification

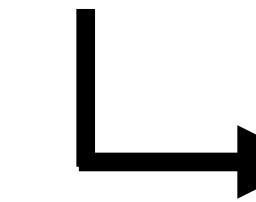
Example

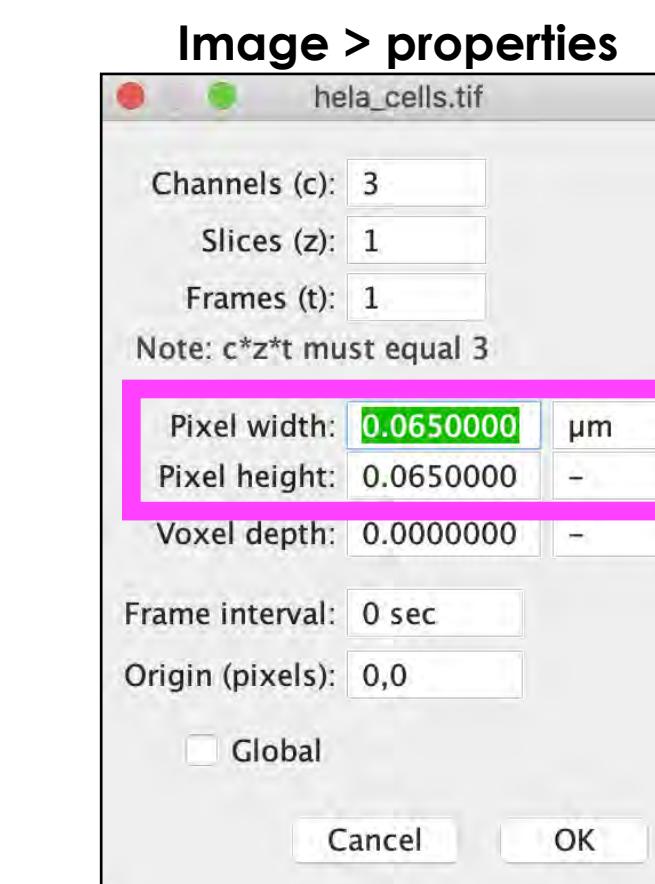
Magnification = 100x Objective

Camera = Hamamatsu Orca Flash 4



Product number	C13440-20CU
Imaging device	sCMOS
Cell (pixel) Size ( $\mu\text{m}^2$ )	6.5x6.5
Pixel Array (horizontal by vertical)	2048x2048
Effective Area (horizontal by vertical in mm)	13.312x13.312

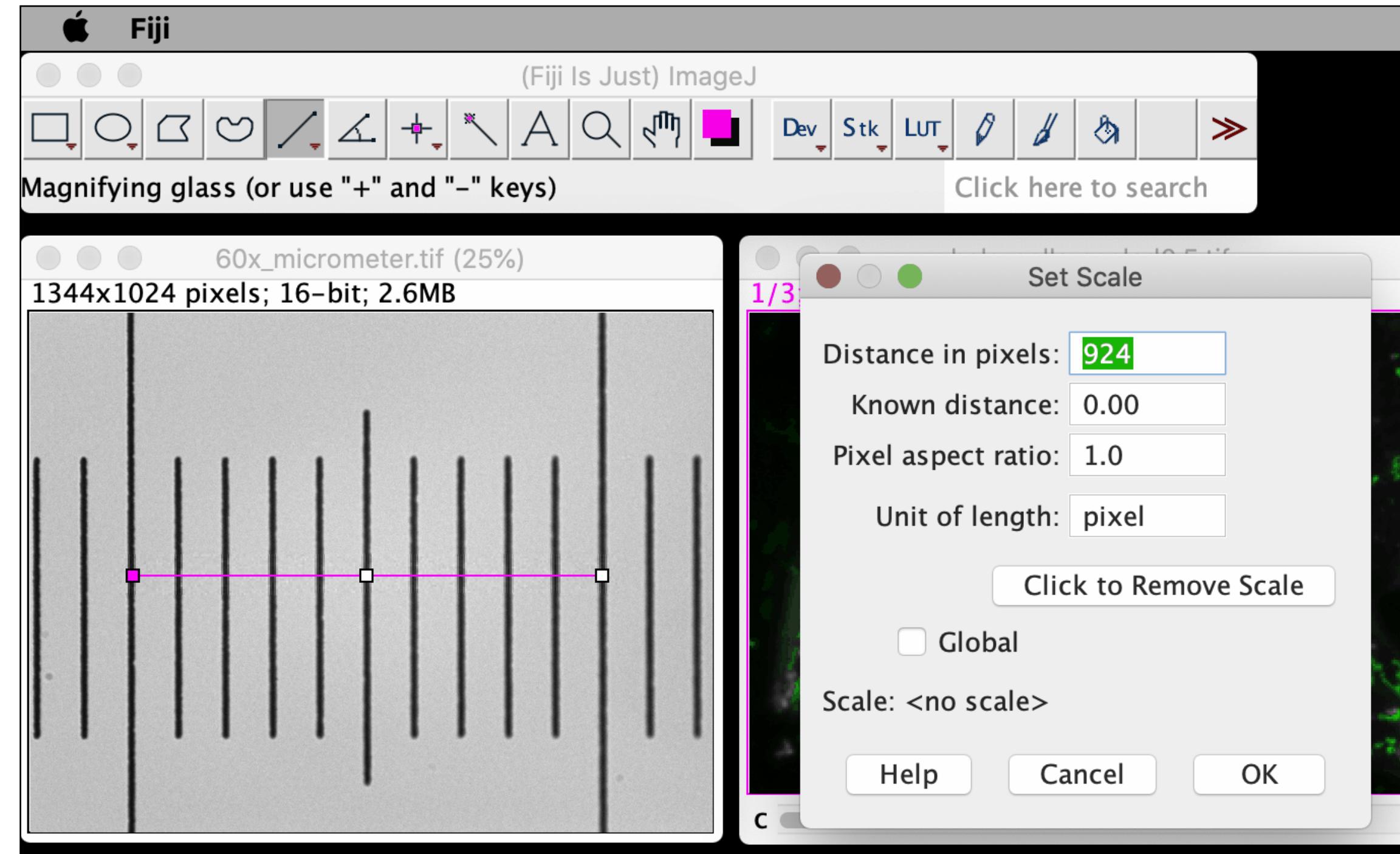

$$\text{pixel width and height: } 6.5 \mu\text{m} / 100X = 0.065 \mu\text{m}$$



If a z-series was acquired, "Voxel depth" is the z step size choose for the acquisition (cannot estimate).

# Scale Bar and Pixel Size

Add pixel information to your images from a reference (same magnification)



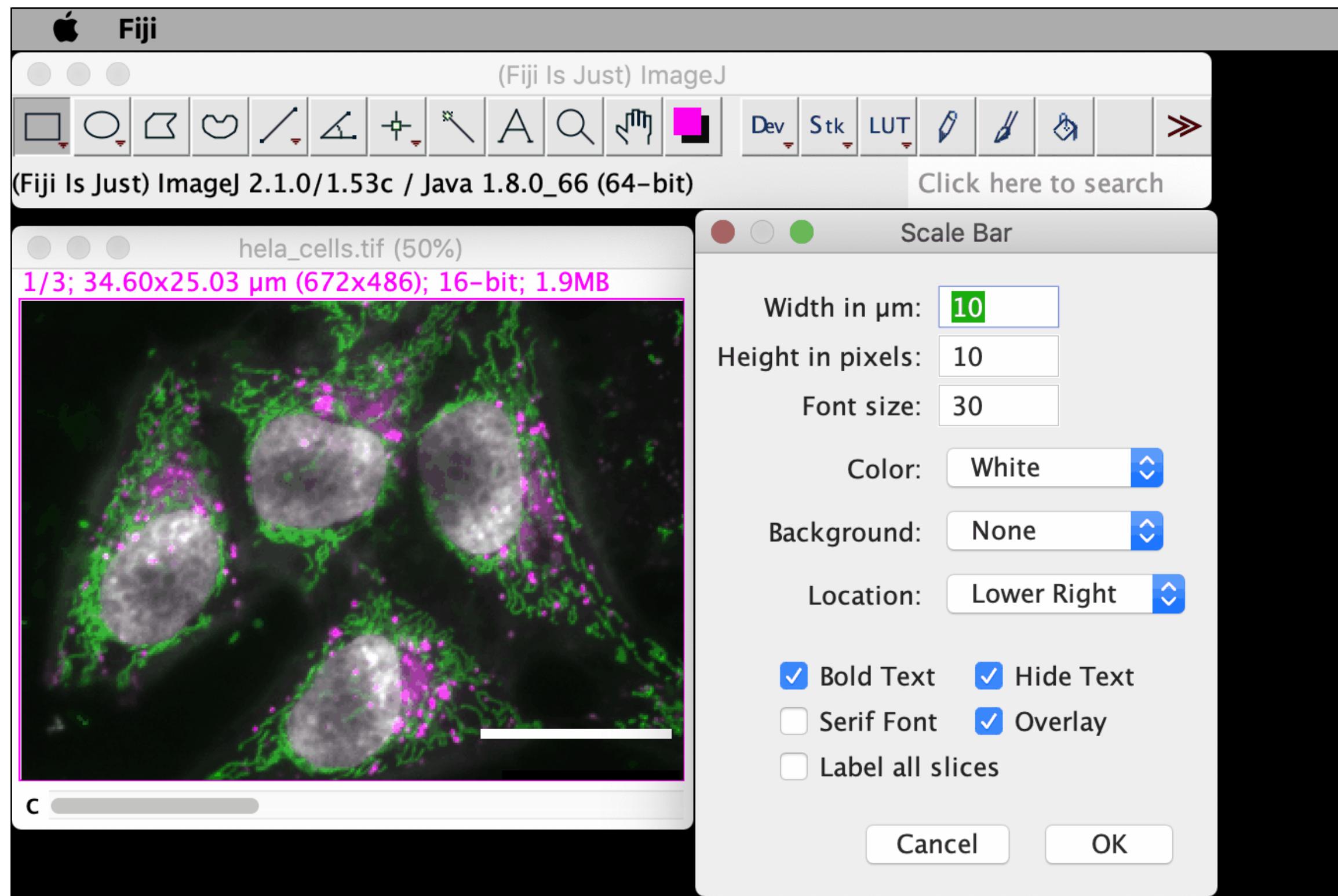
Analyze > Set Scale...

**Set the pixel size information starting from an object with a known dimension.**

**\*Perform multiple measures and average them to be more precise**

# Scale Bar and Pixel Size

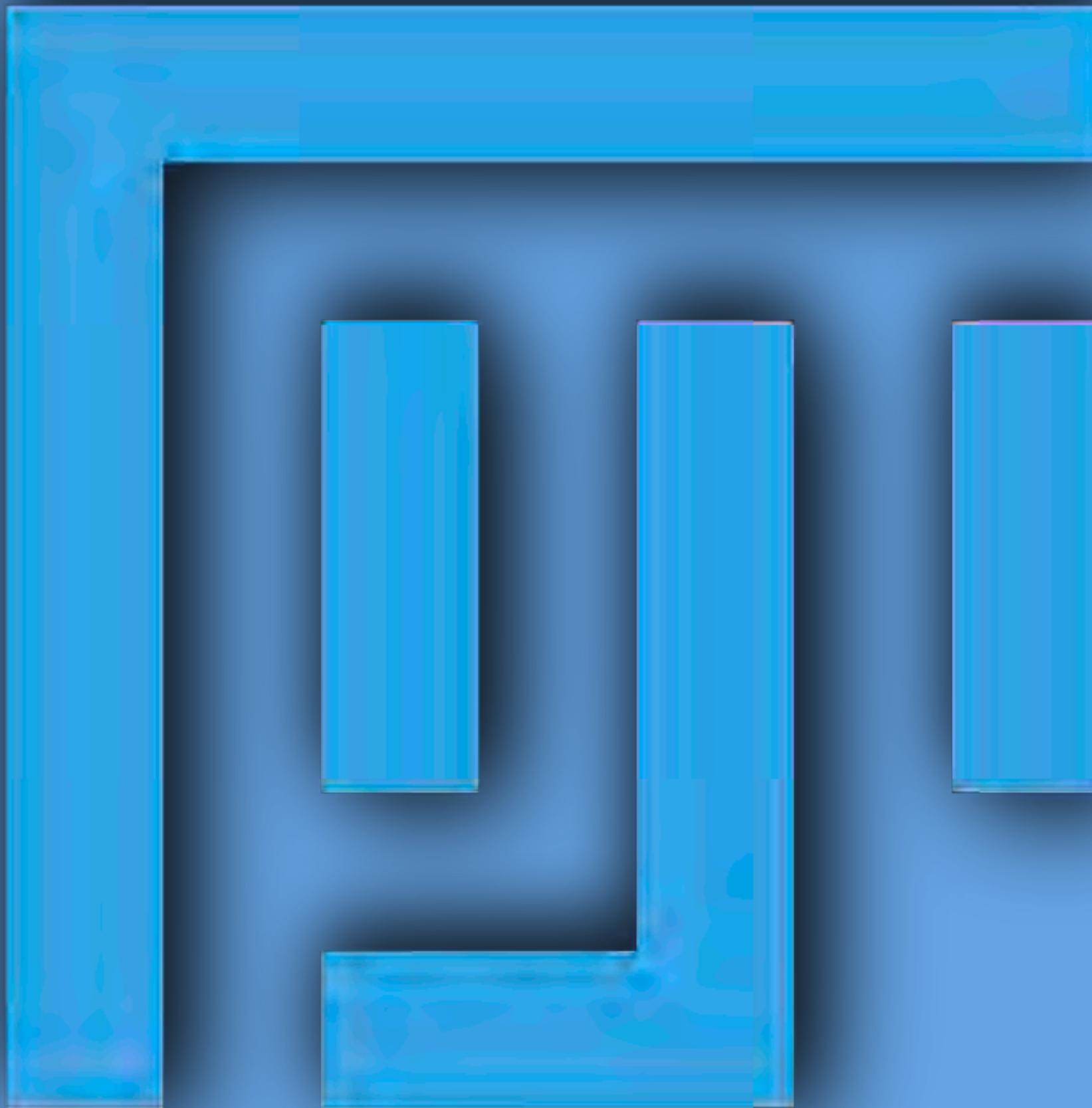
Add scale bar to your images



Analyze > Tools > Scale Bar...

- “Overlay” means that the scale bar is not embedded in your image. You can remove it: “Image > Overlay > Remove Overlay”.
- If you save the image as tiff, the scale bar will be saved as an overlay.
- If you save as jpeg/png, the image will have the scale bar embedded.

# Segmentation with pixel based classifier—exercises



**1.3 file handling and non-invasive editing**

**1.4 file handling and invasive editing**



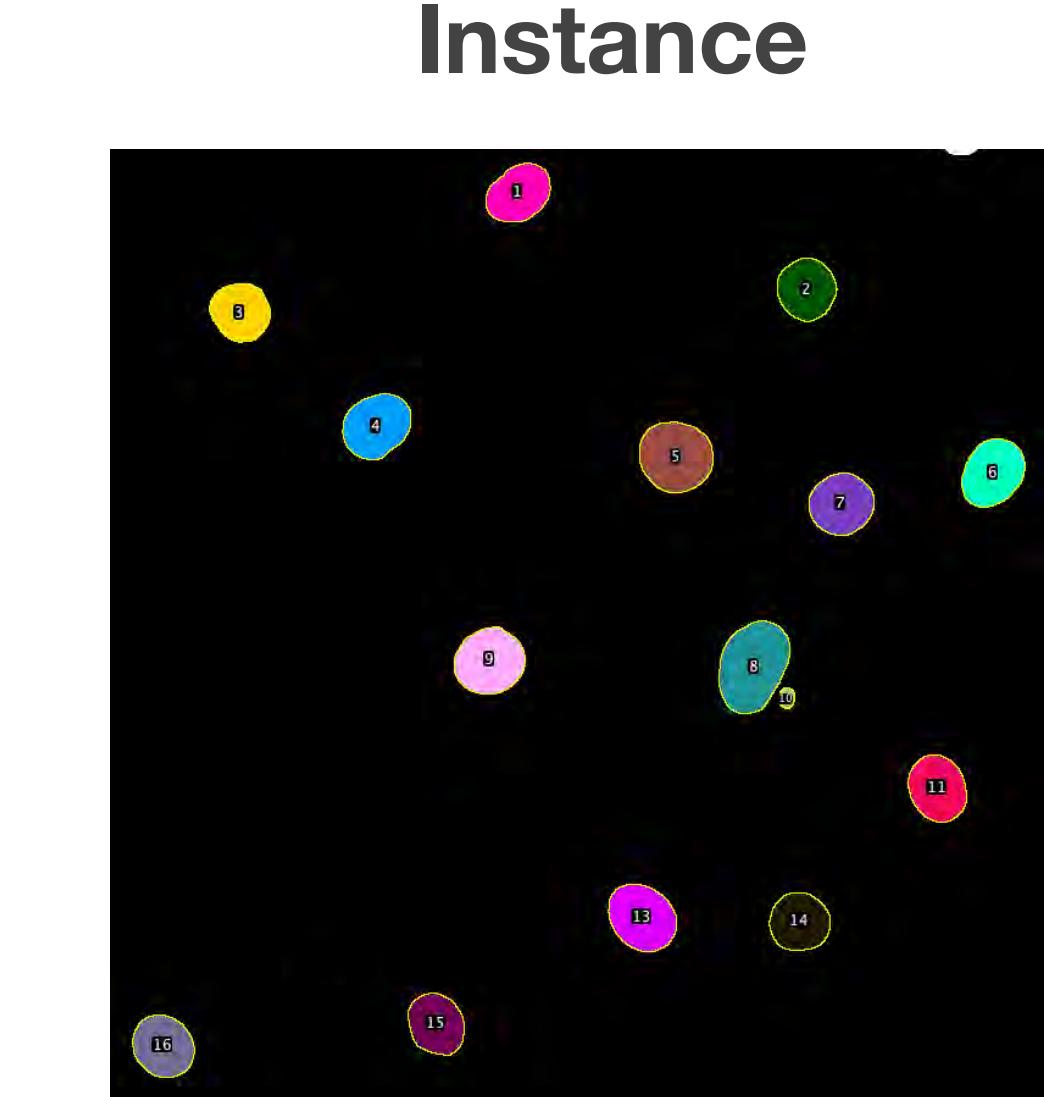
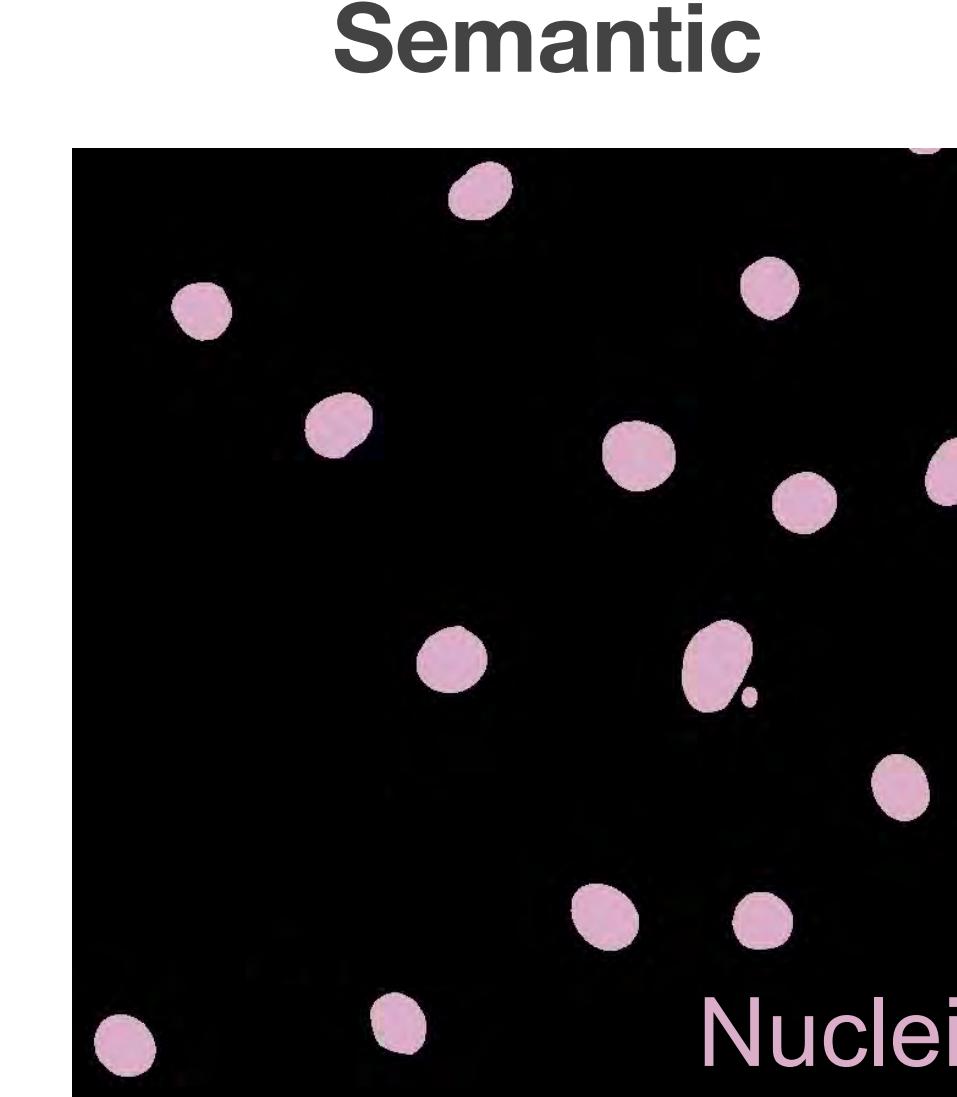
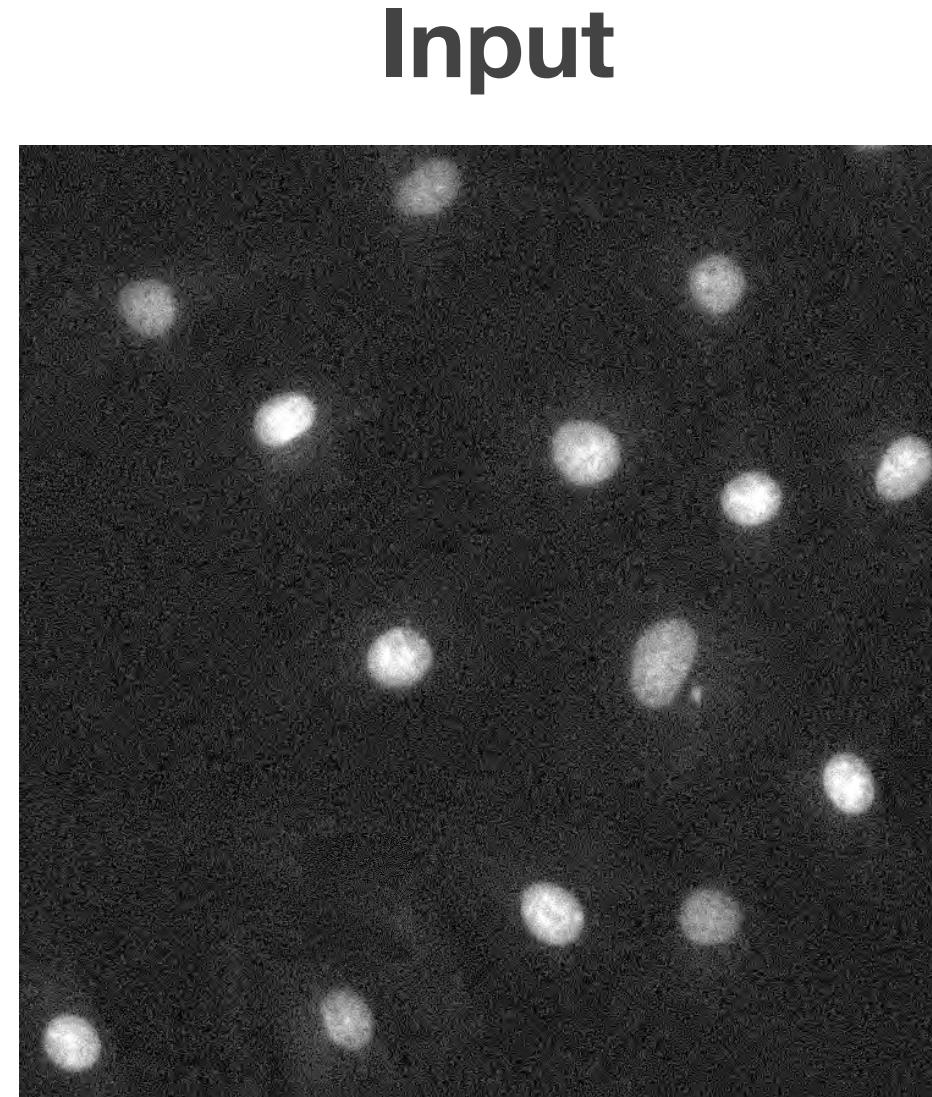
# Image Processing

---

## Segmentation

- Segmentation is the **division** of an image into **discrete regions**.

# Segmentation



Background  
Nucleus 1  
Nucleus 2  
Nucleus 3  
...

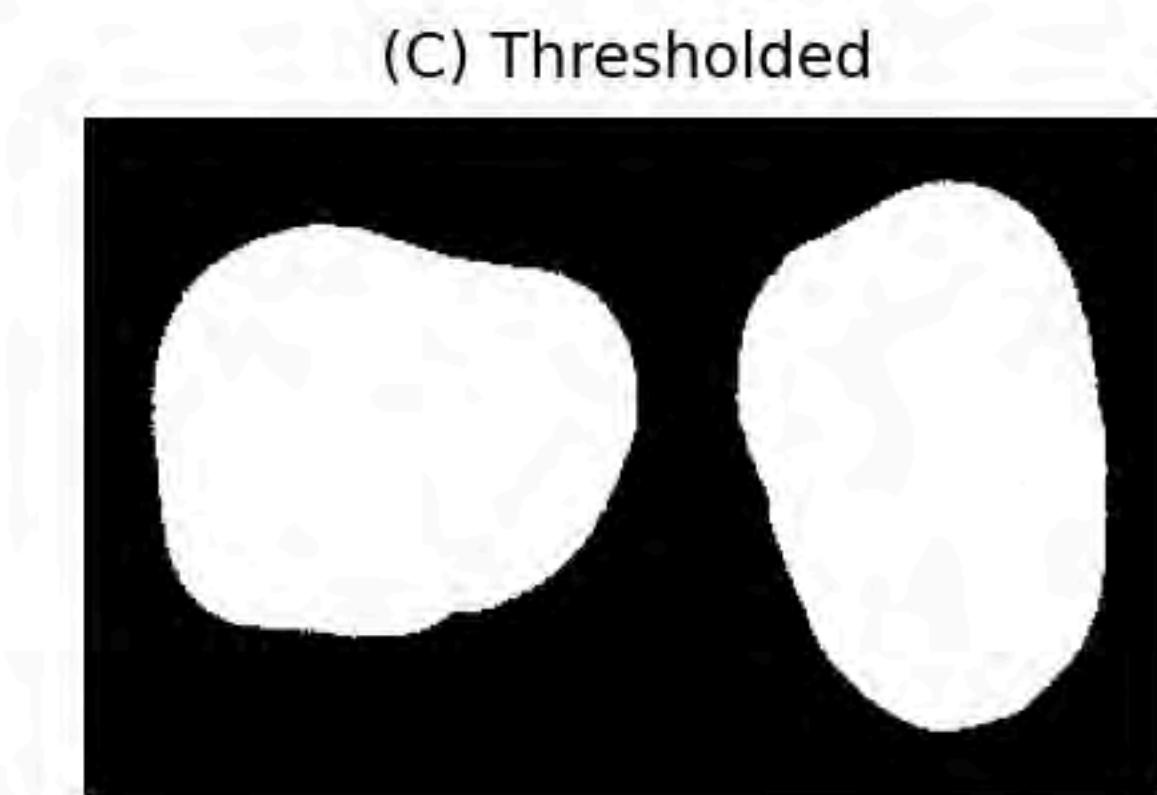
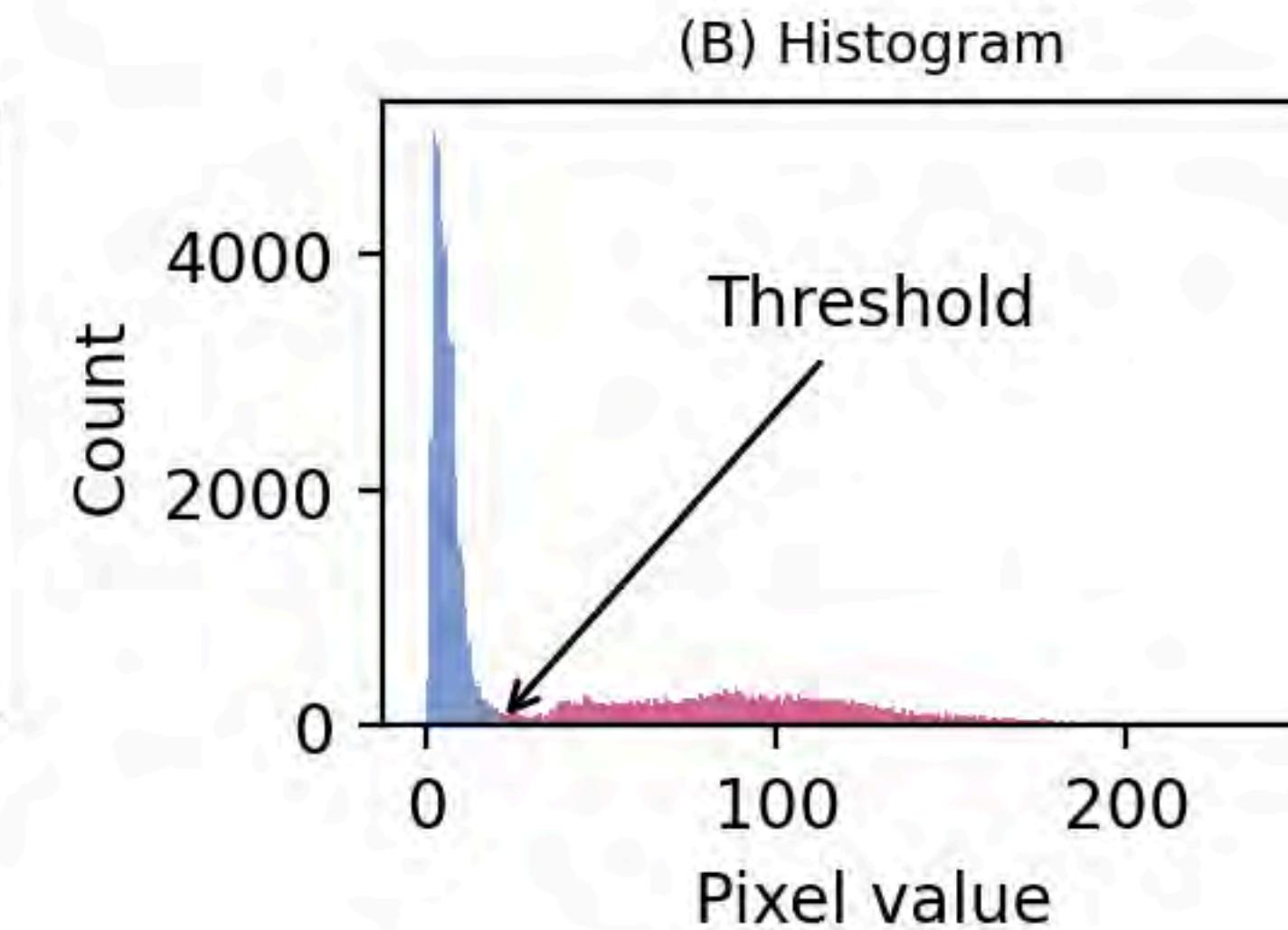
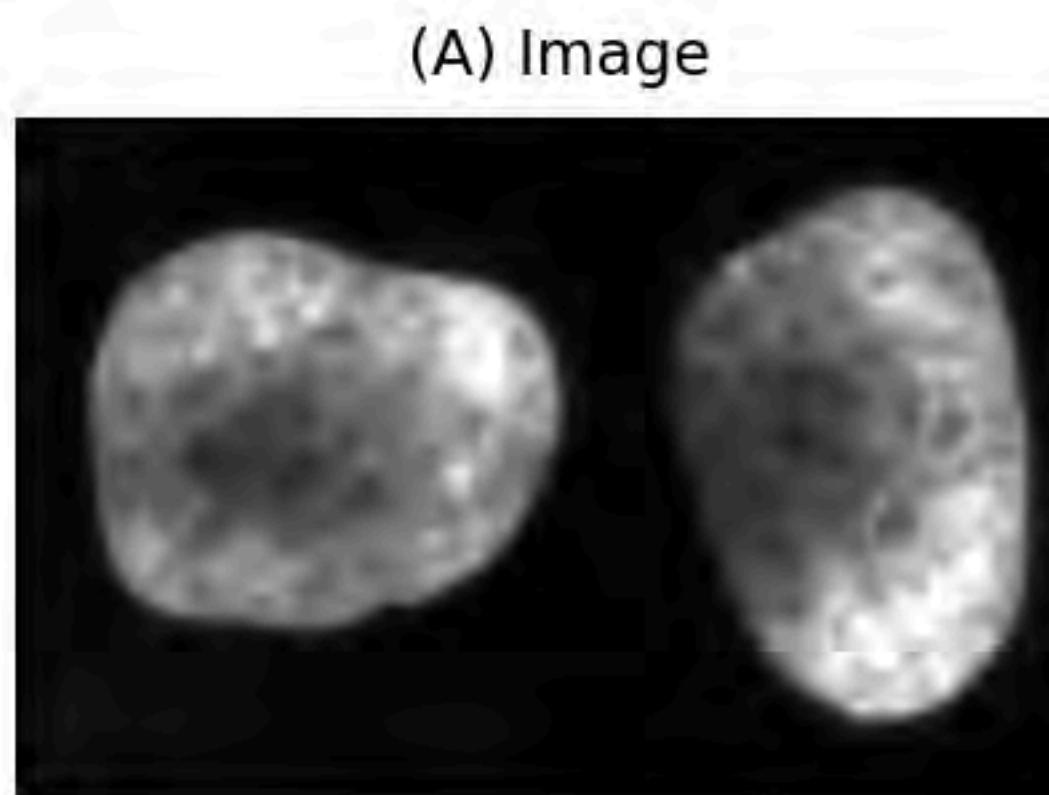
# How do we get segments?

- Thresholding-based
- Interactive tools based on classic machine learning
- Deep-learning based (Stardist, Cellpose)

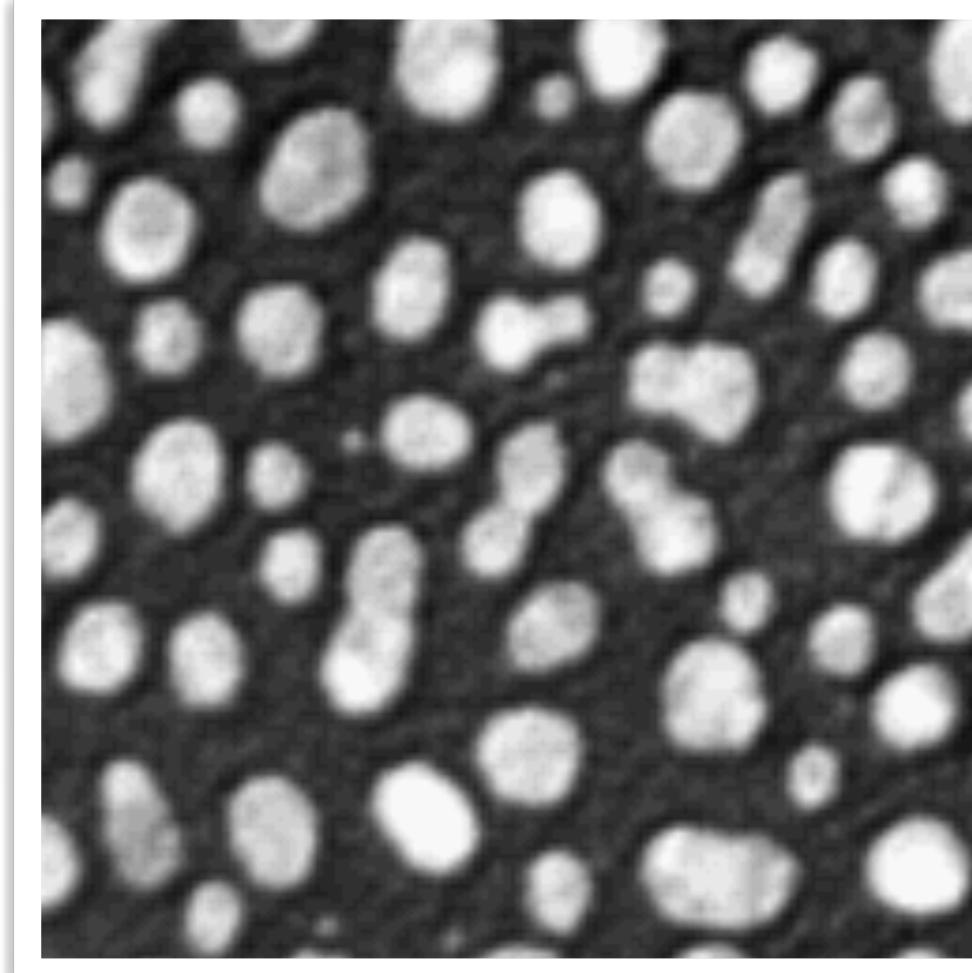
# Thresholding

The easiest way to segment an image is often by applying a *global threshold*.

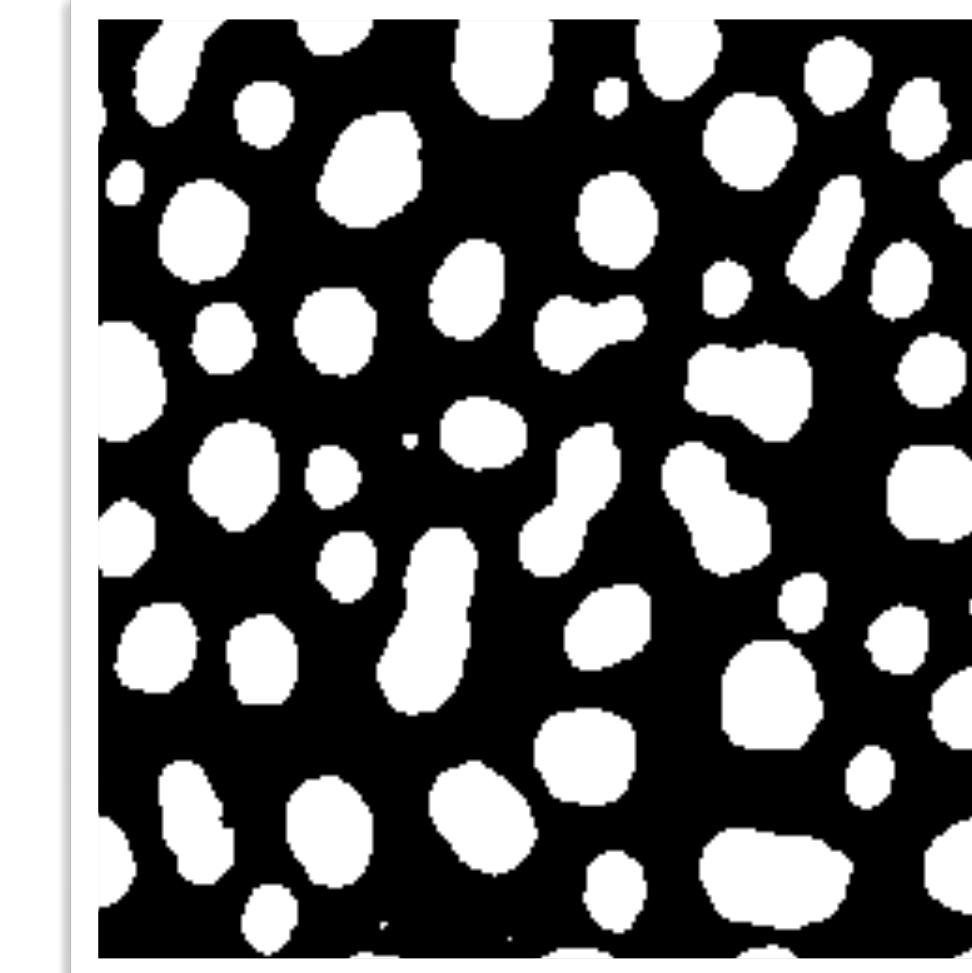
This identifies pixels that are above or below a fixed threshold value, giving a *binary image as the output*.



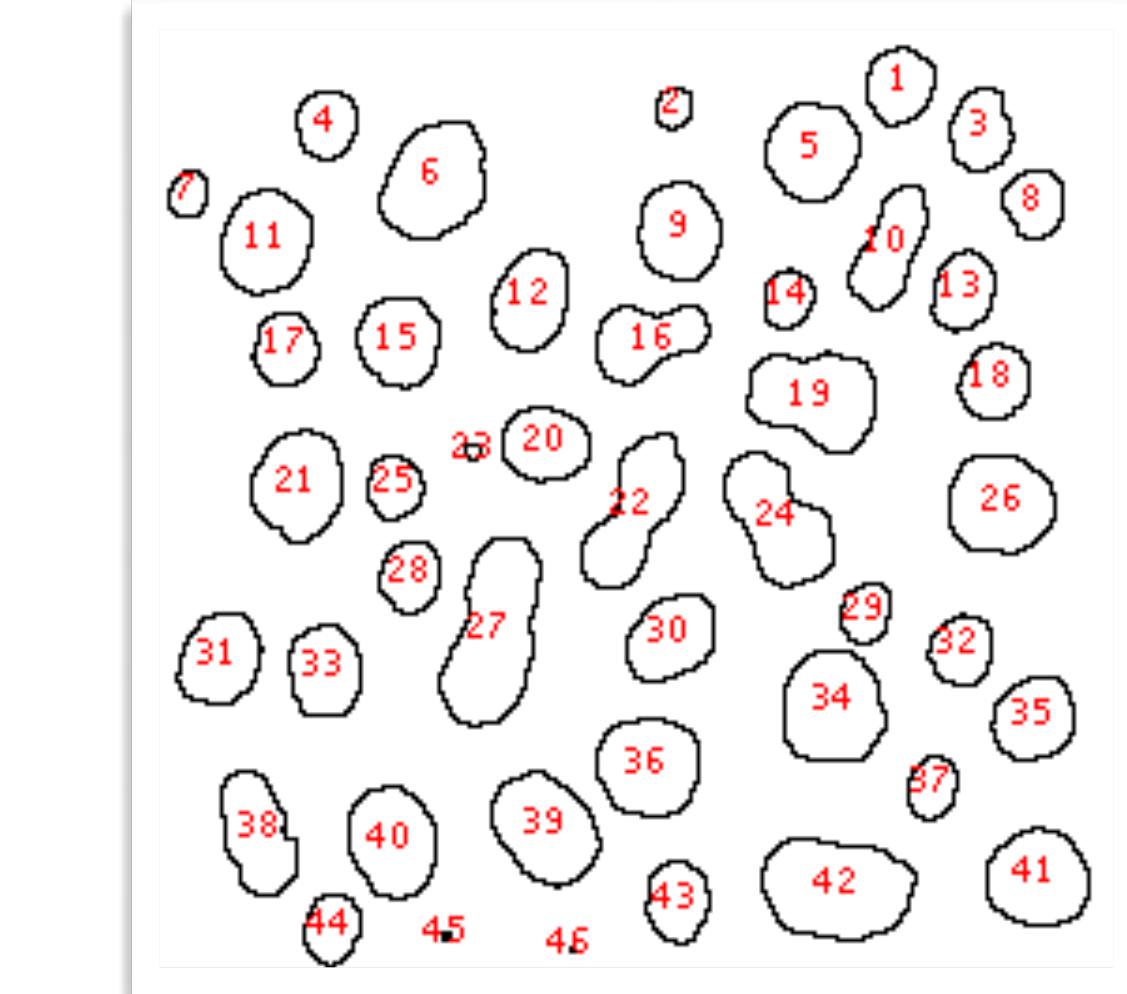
# Thresholding



**Original**, 8 bit grayscale  
Blobs: Fiji example



**Thresholded**

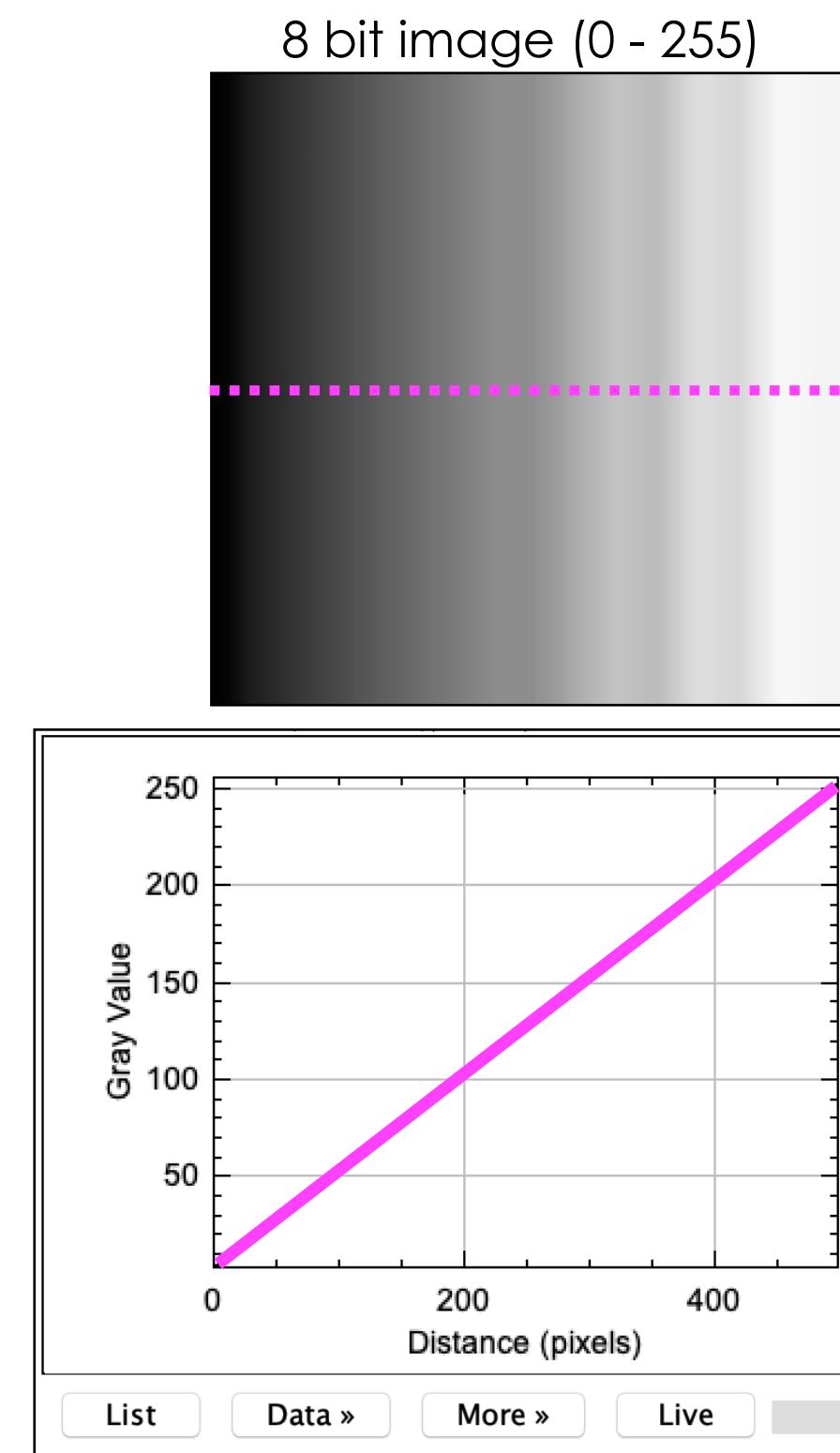


**Instance Segmented**

**Instance Segmentation** in FIJI: keeping white (connected-) objects.

# Thresholding method

Select only a range of digital values in the image.

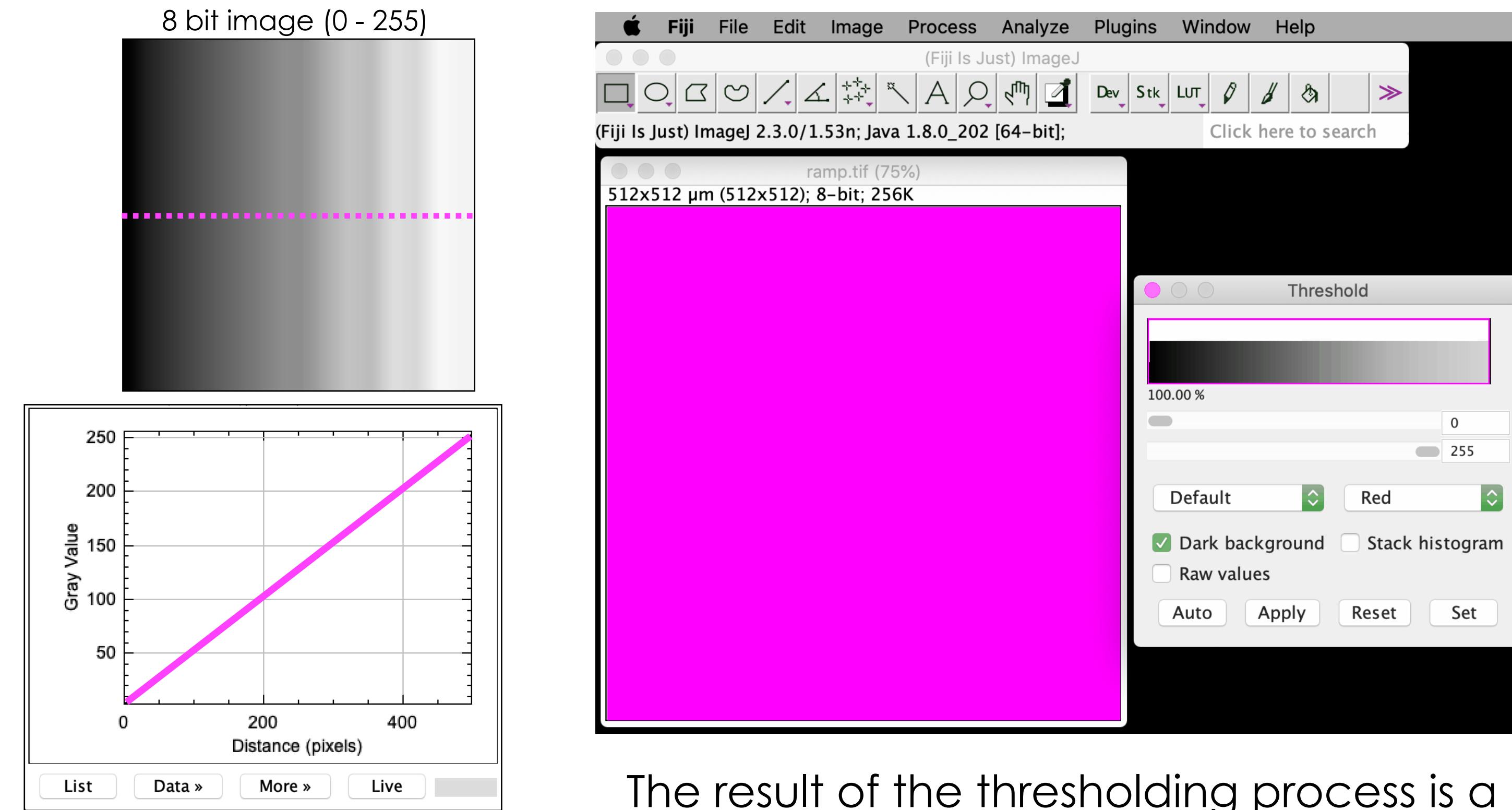


# Thresholding method

Select only a range of digital values in the image.

in Fiji: **Image > Adjust > Threshold...**

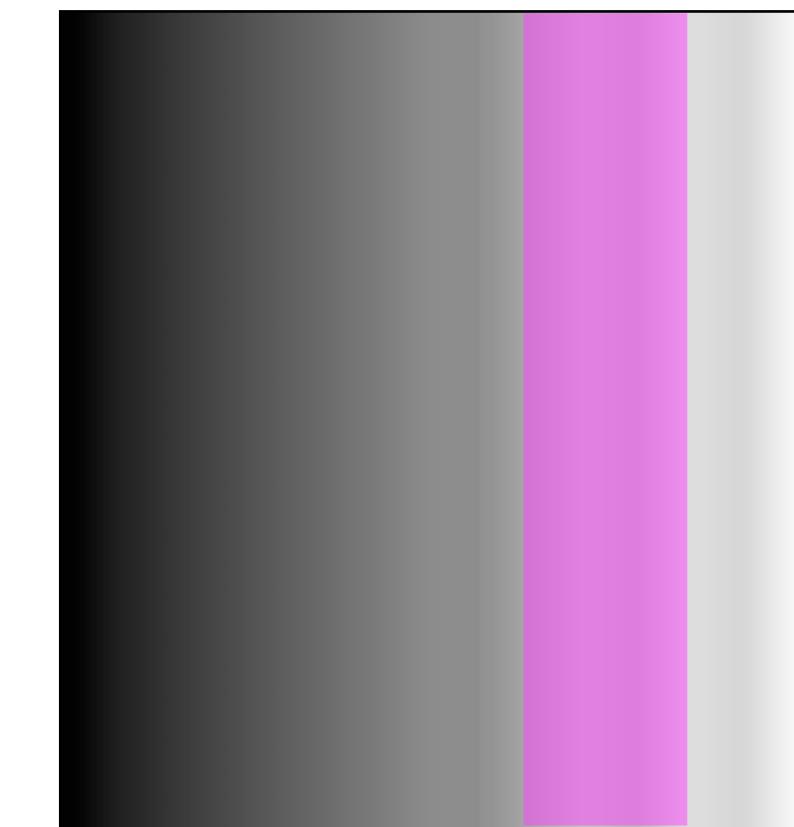
**(cmd) + shift + t**



# Thresholding method

# Generate a **binary mask**.

## 8 bit image (0 - 255)

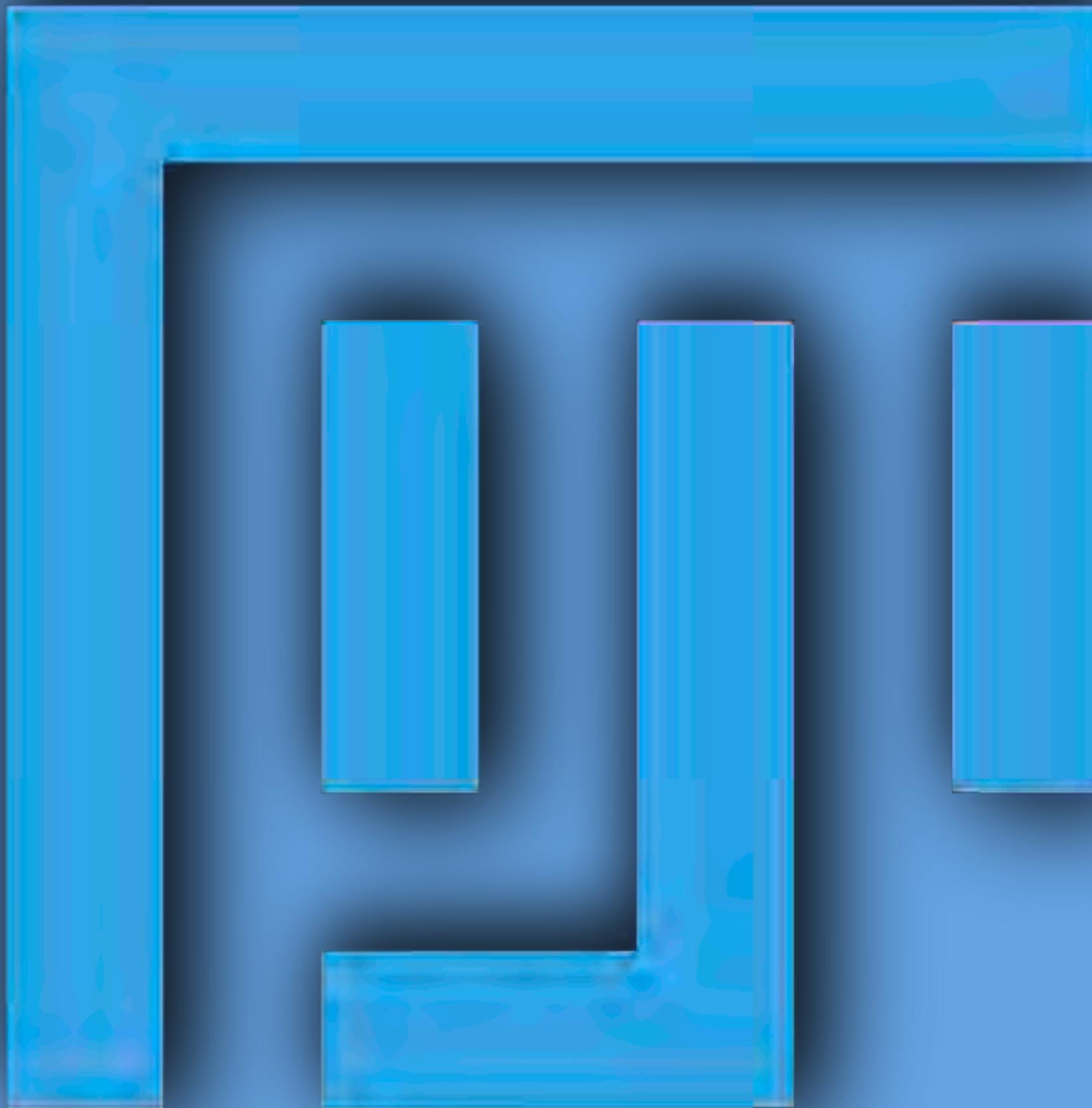


0 or 255

**Binary** because the image has only **two** pixel values,  
**one for the selected pixels** and **one for the “discarded” pixels.**

In Fiji the two pixel values are **0** and **255**.

# Segmentation with thresholding—exercises

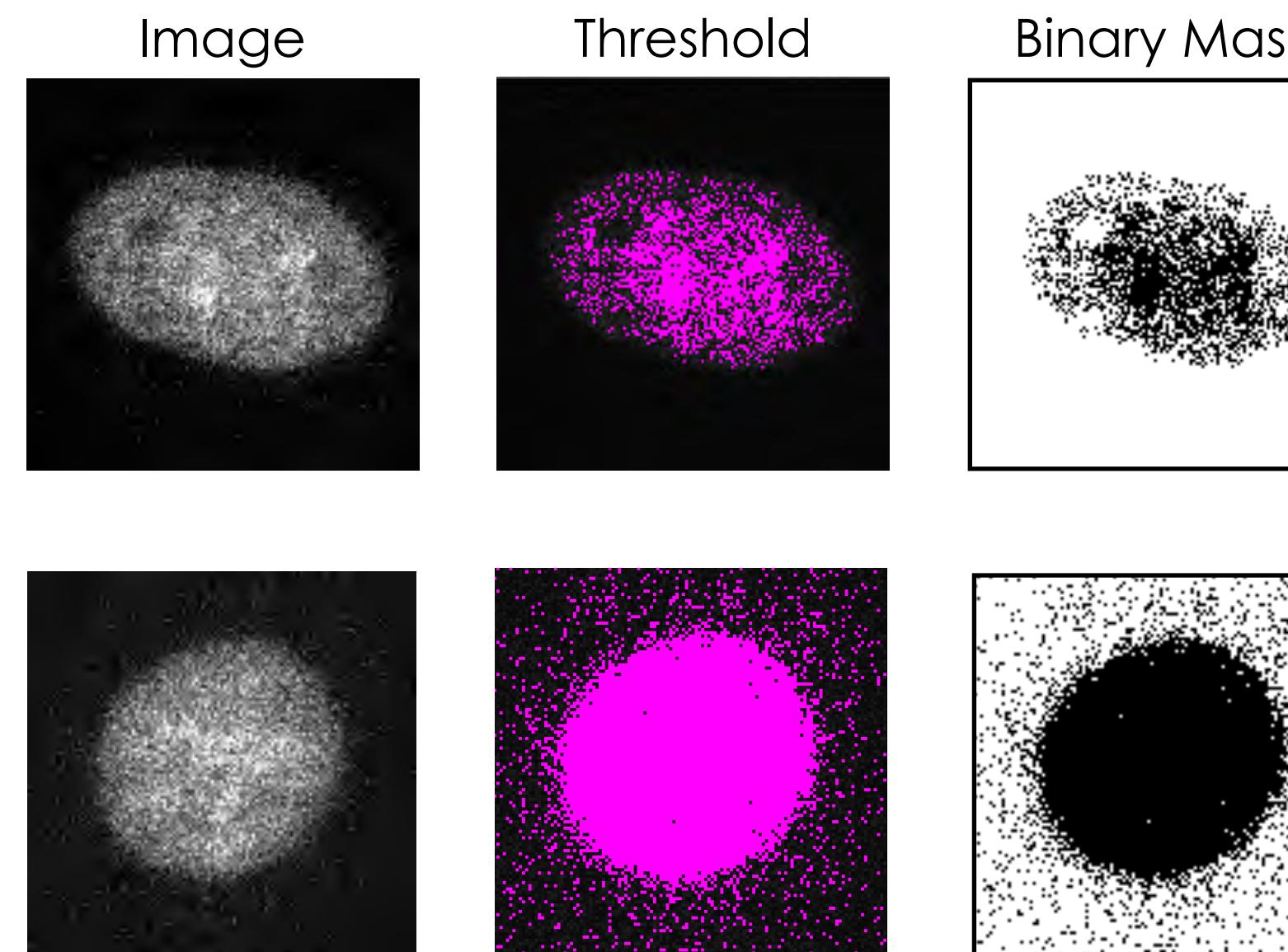


***4.1 DAPI segmentation with thresholding***  
***skip “Analyze Particles”***

# Thresholding method

What can go wrong?

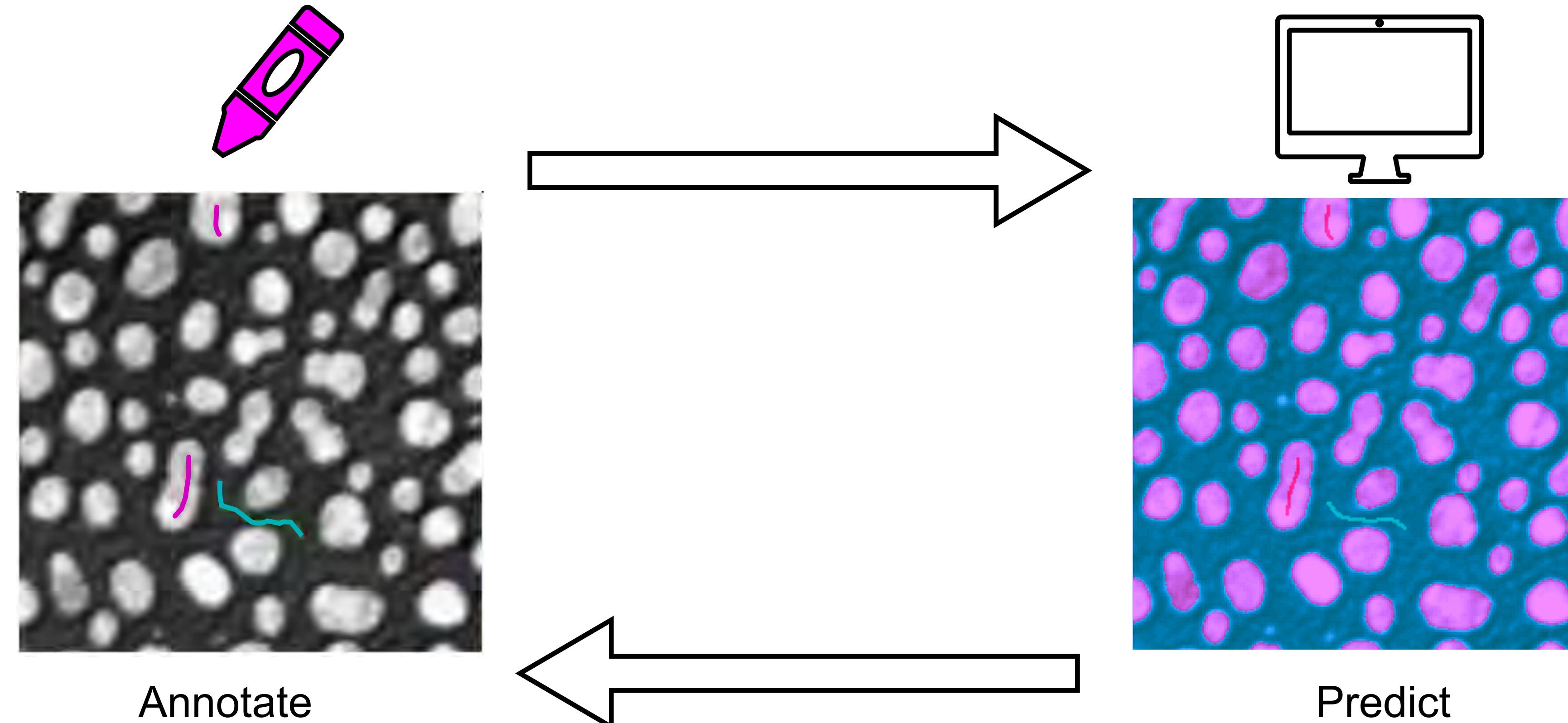
Usually, if you apply **thresholding** to the “**ORIGINAL**” **image** (the one you get out of the microscope), you won’t be able to precisely **select all/only the pixels** you are interested in.



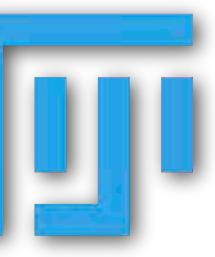
- **Fluorescence label** (e.g. DAPI)
- **Background** (uneven illumination, out-of-focus light, aberration, ...)
- **Noise** (detector read noise, Poisson noise, ...)



# Interactive tools based on classic machine learning



# Examples



## Standalone: Ilastik



<https://www.ilastik.org/>

## For histopathology



<https://qupath.github.io/>

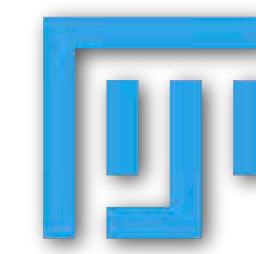
## As a Fiji plugin: Labkit



Labkit

<https://imagej.net/plugins/labkit/>

# Labkit Segmentation

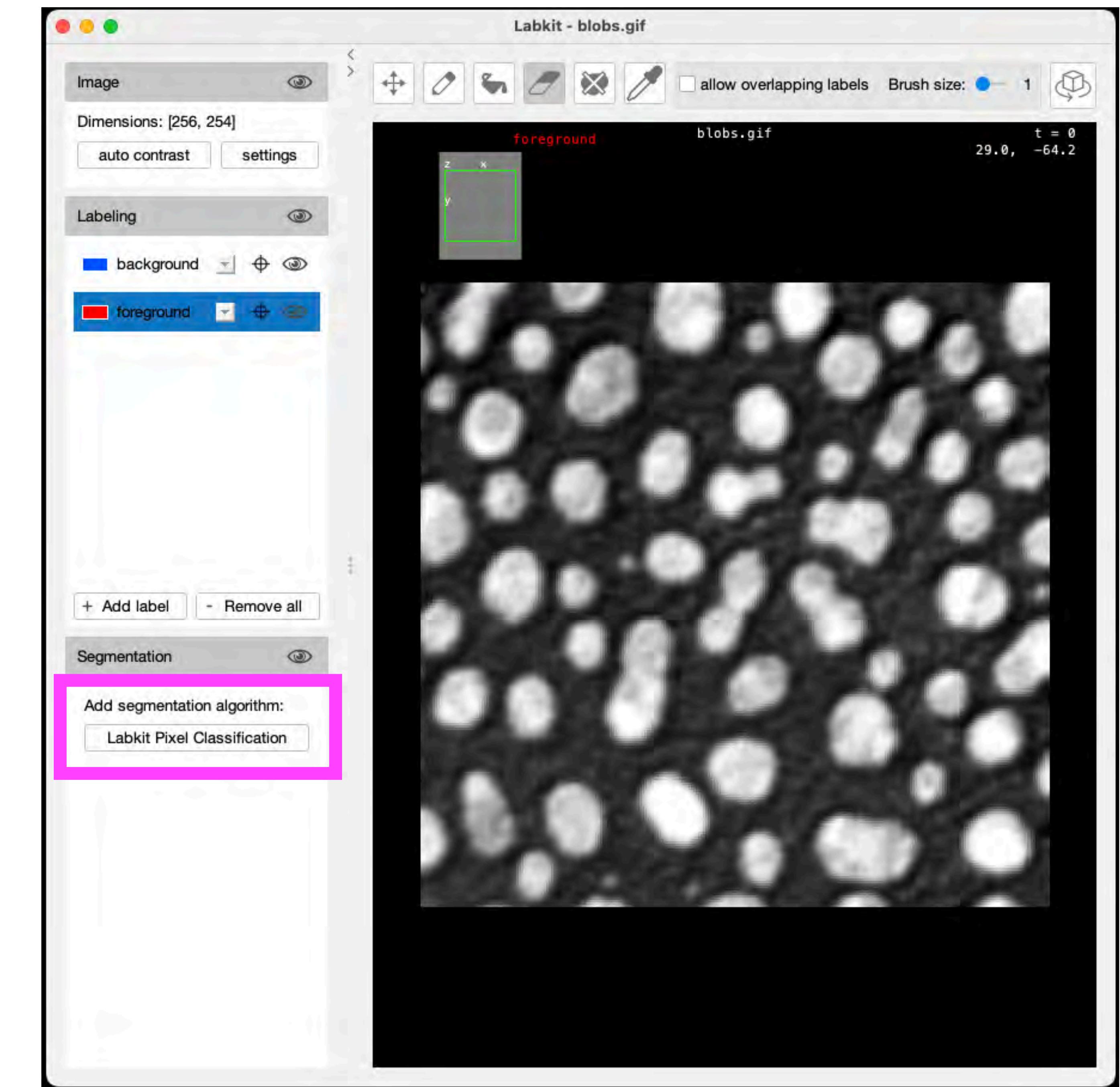
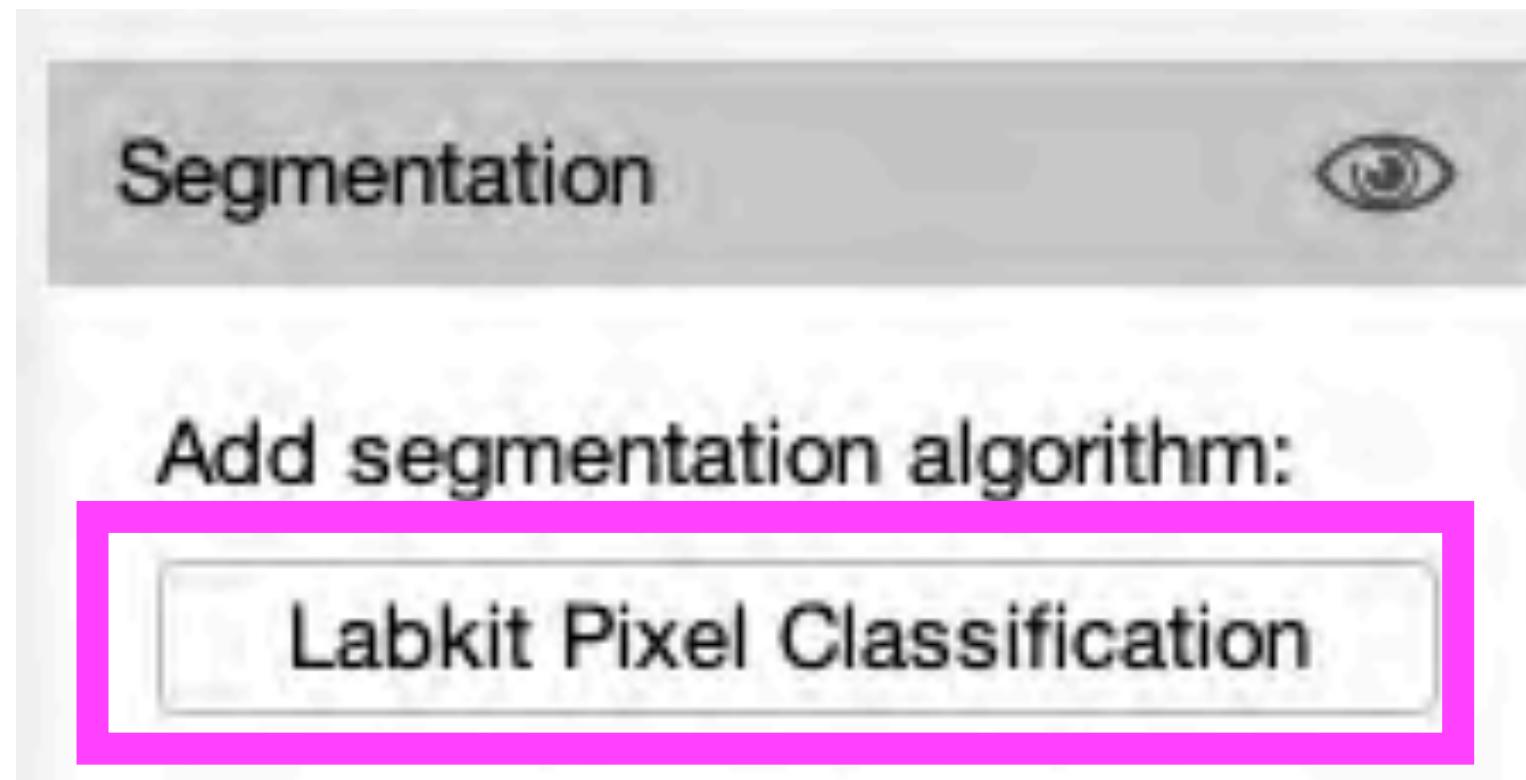


in **Fiji**:

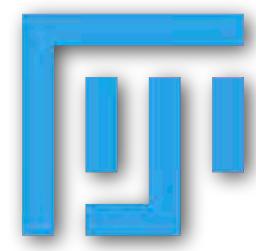
**Plugins**

> **Labkit**

> Open Current Image With Labkit



# Labkit Segmentation

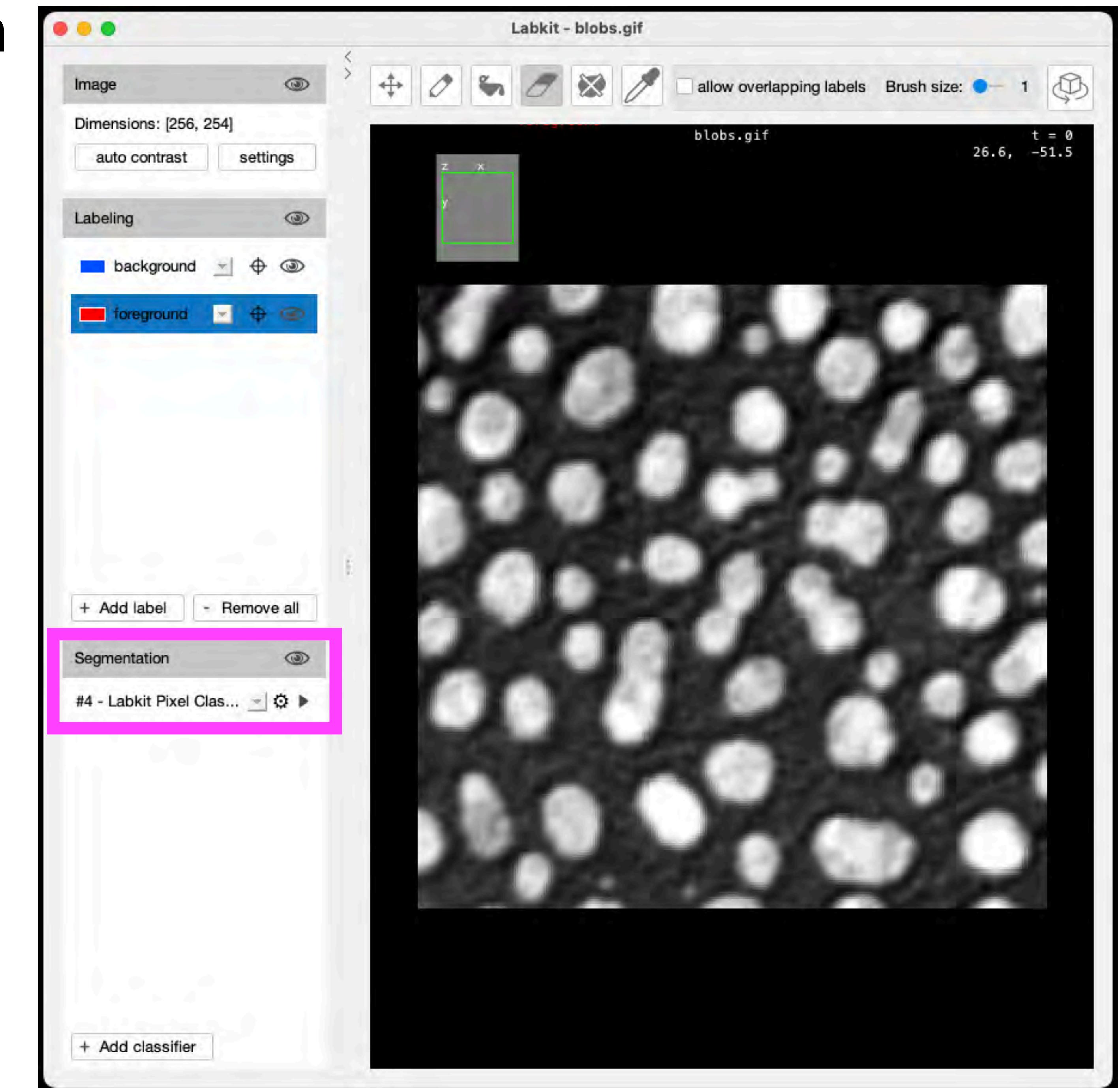
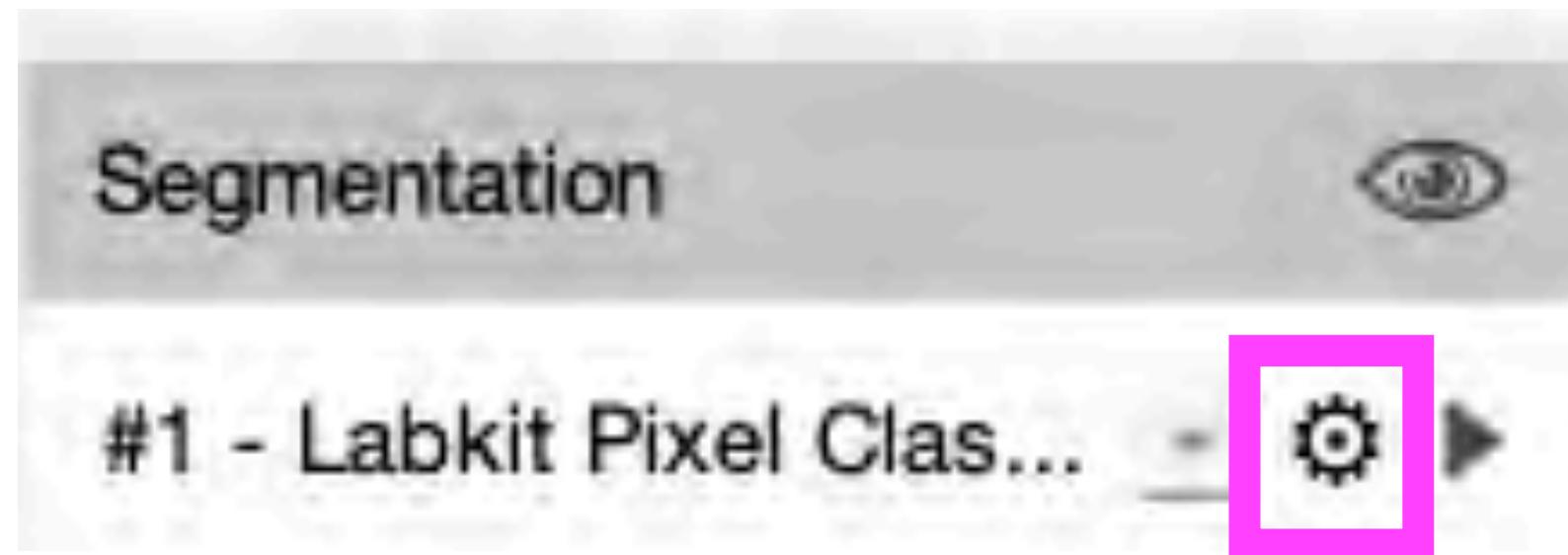


in **Fiji**:

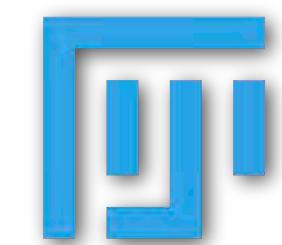
**Plugins**

> **Labkit**

> Open Current Image With Labkit



# Labkit Segmentation

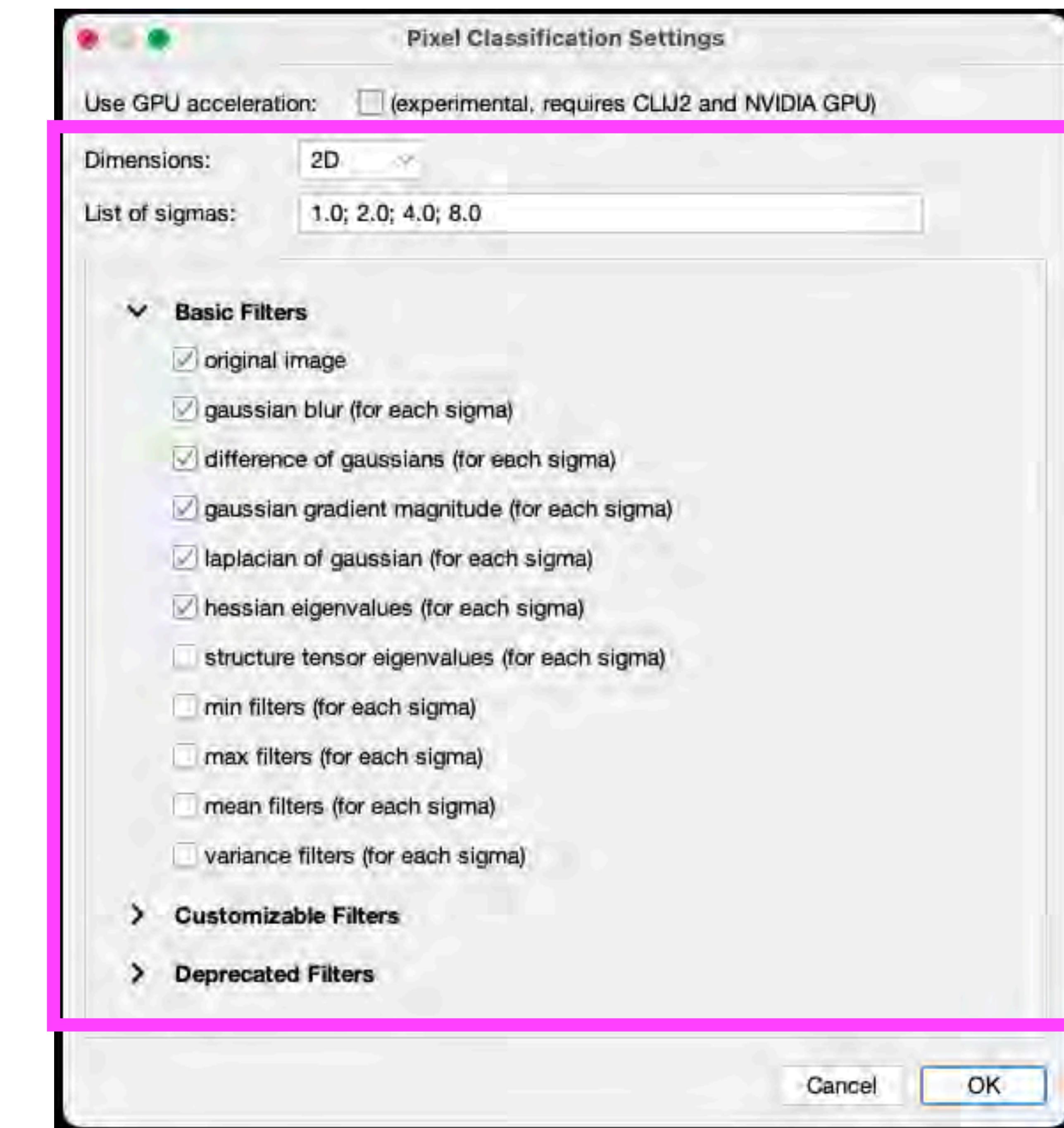
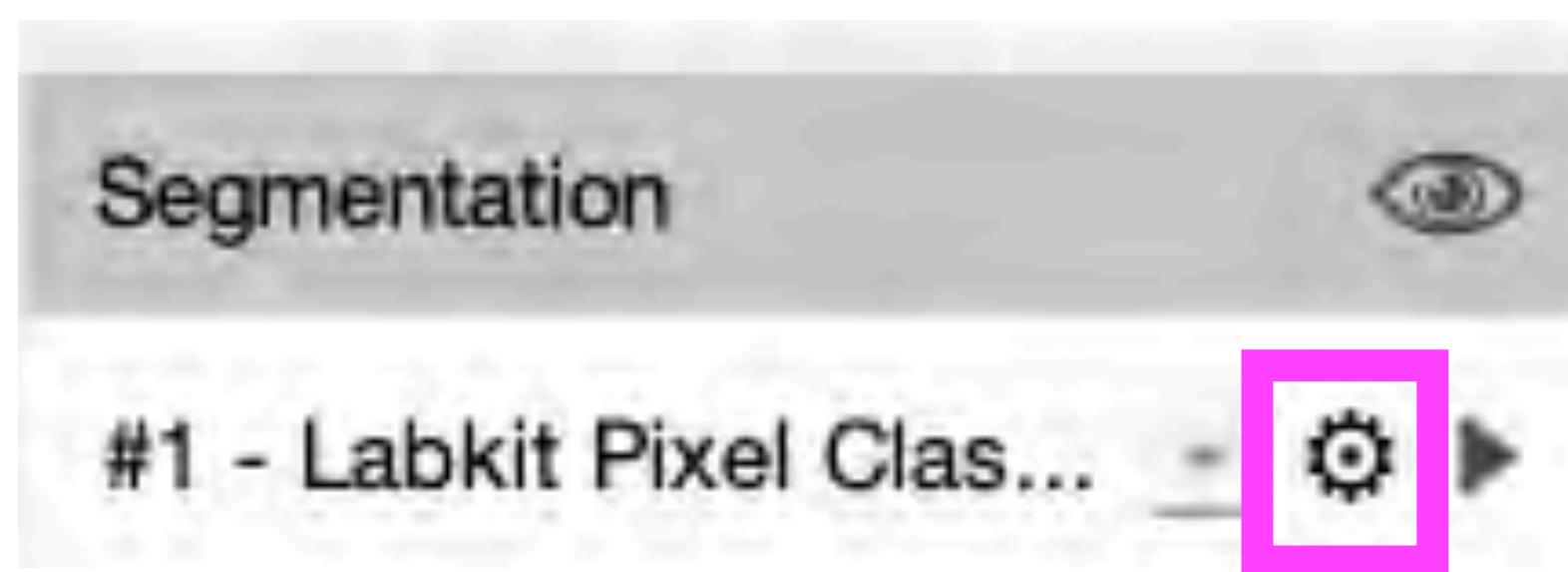


in Fiji:

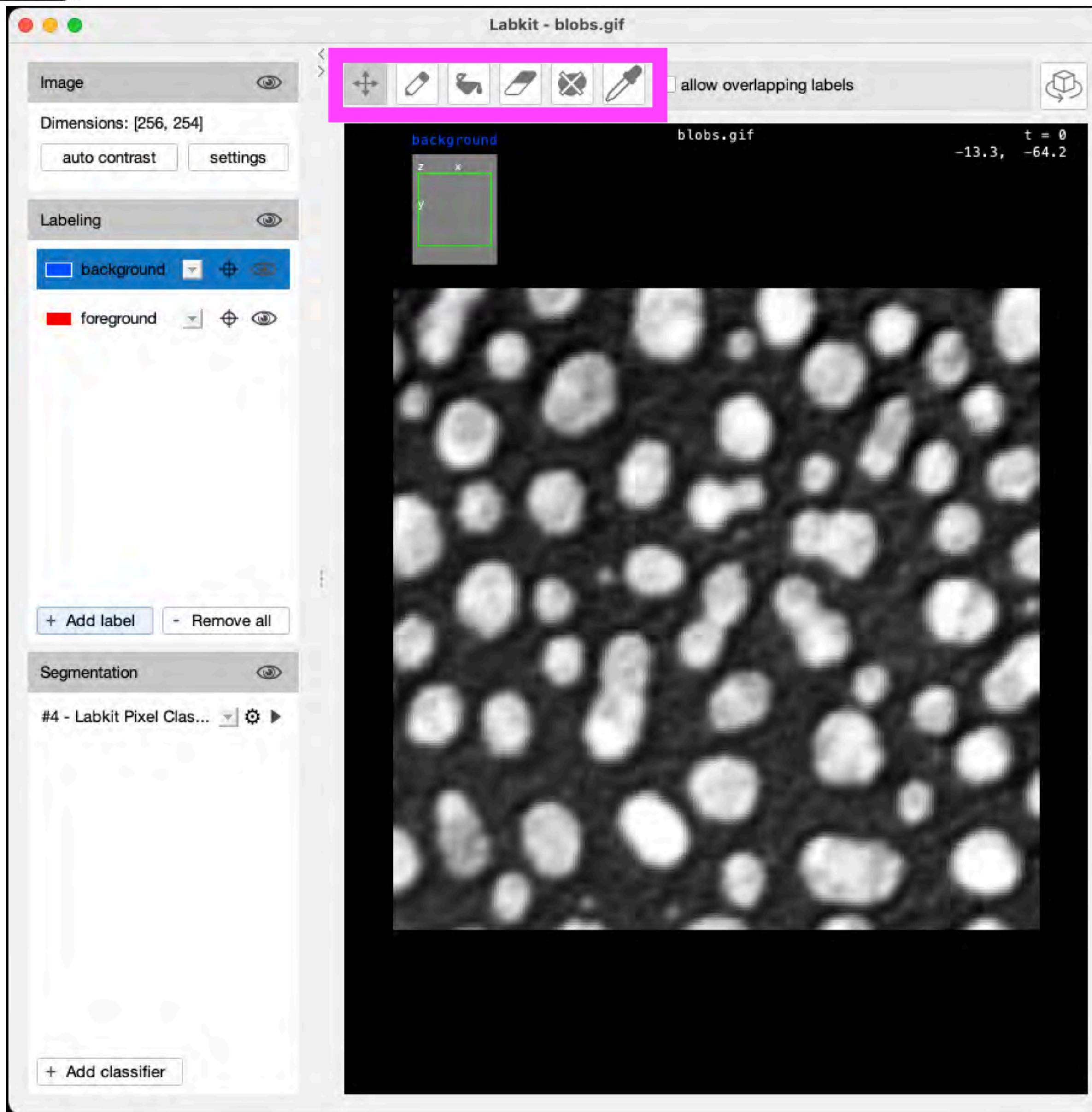
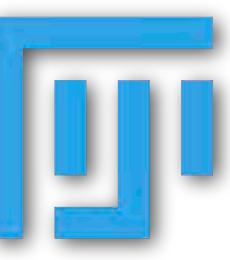
Plugins

> Labkit

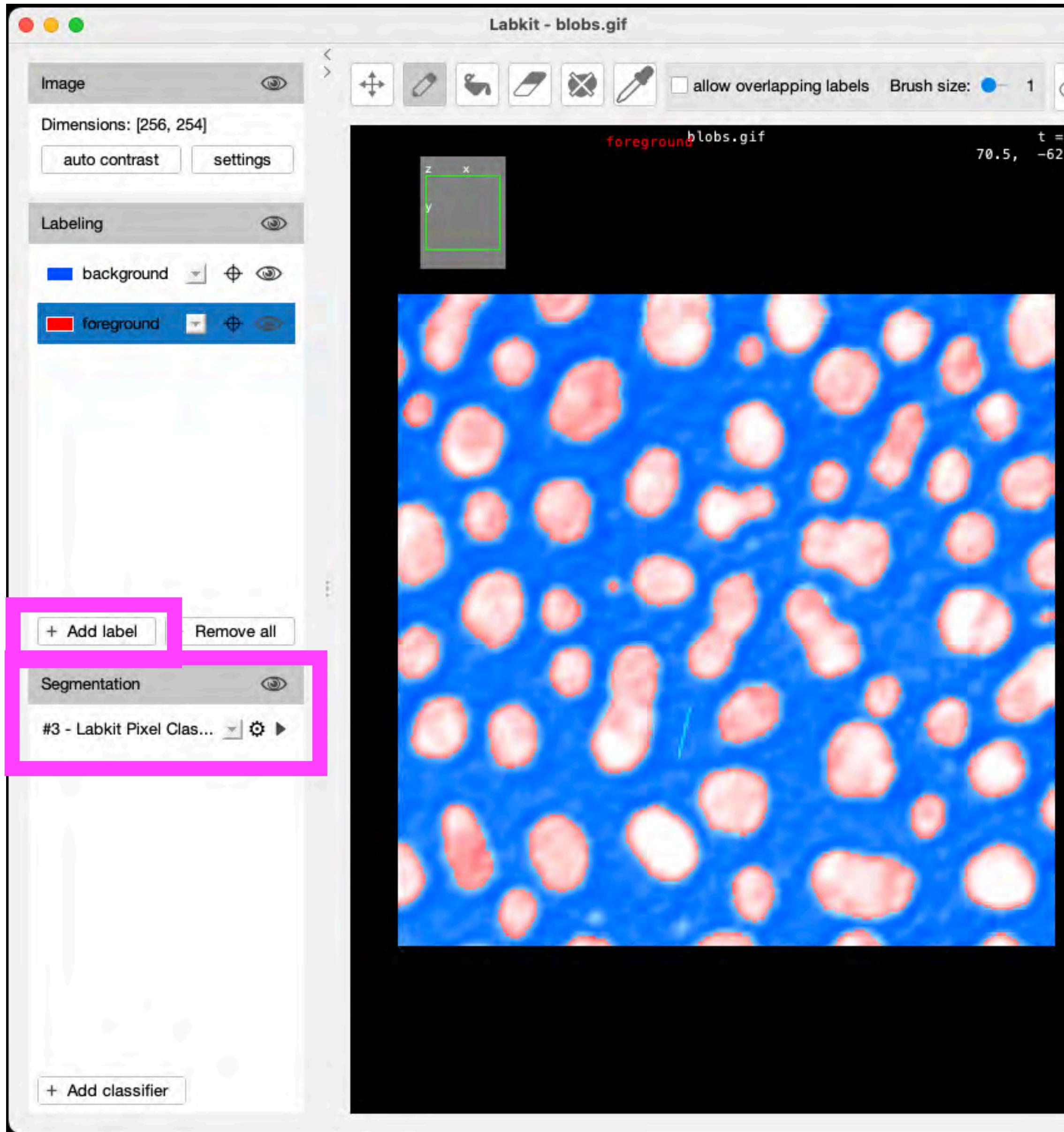
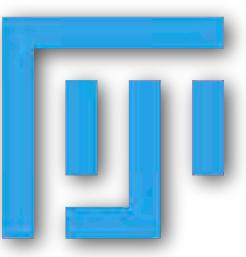
> Open Current Image With Labkit



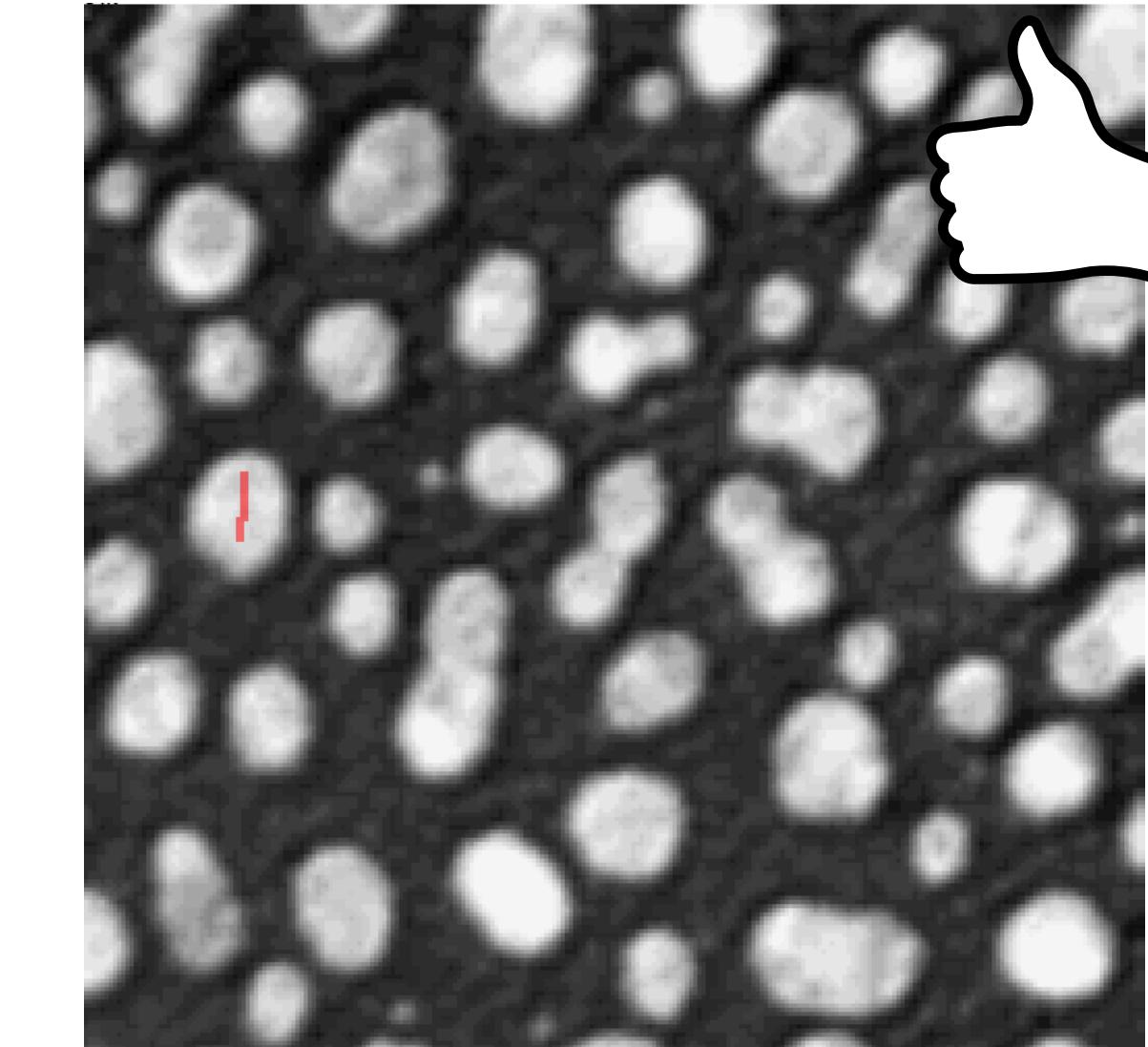
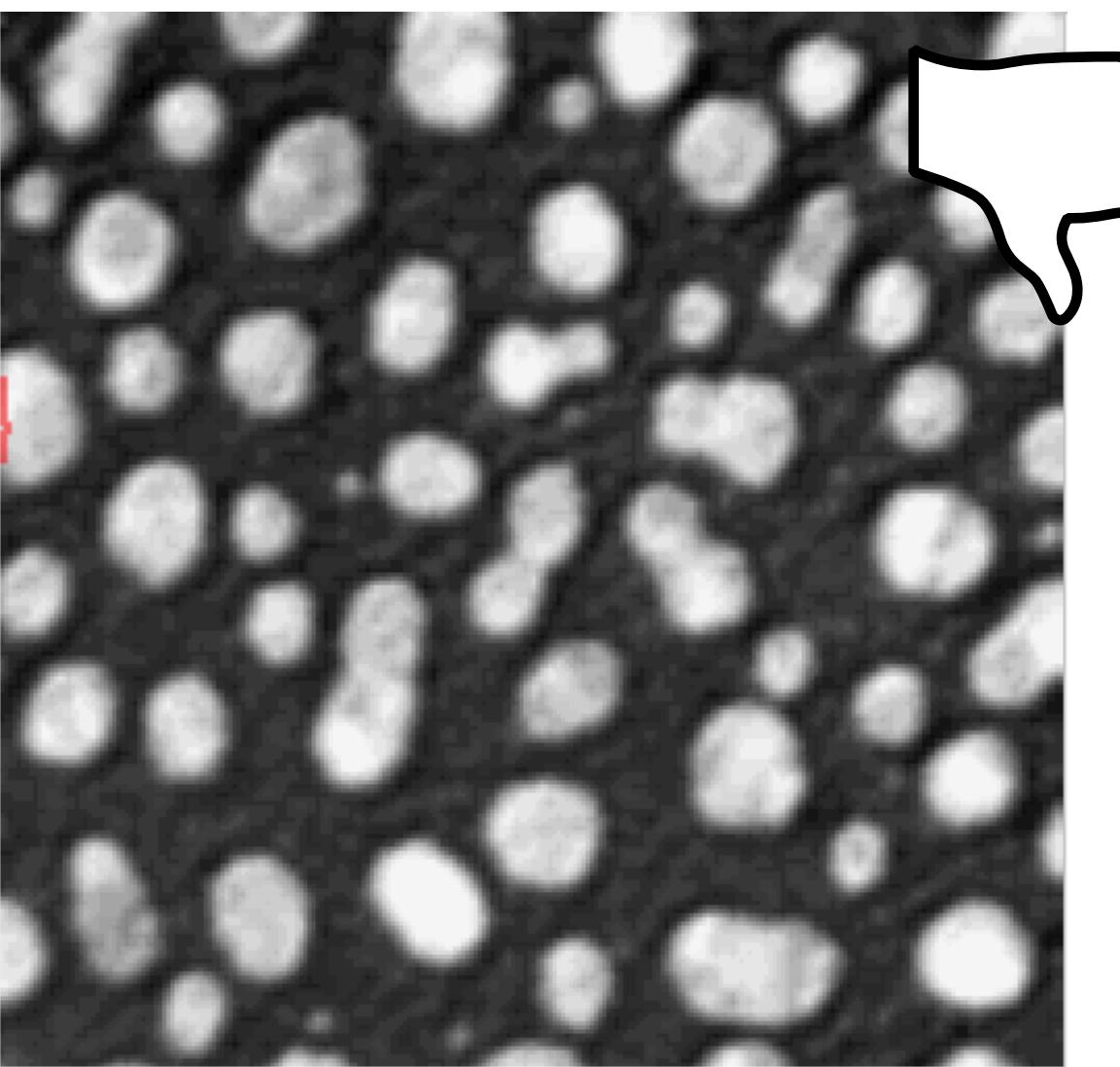
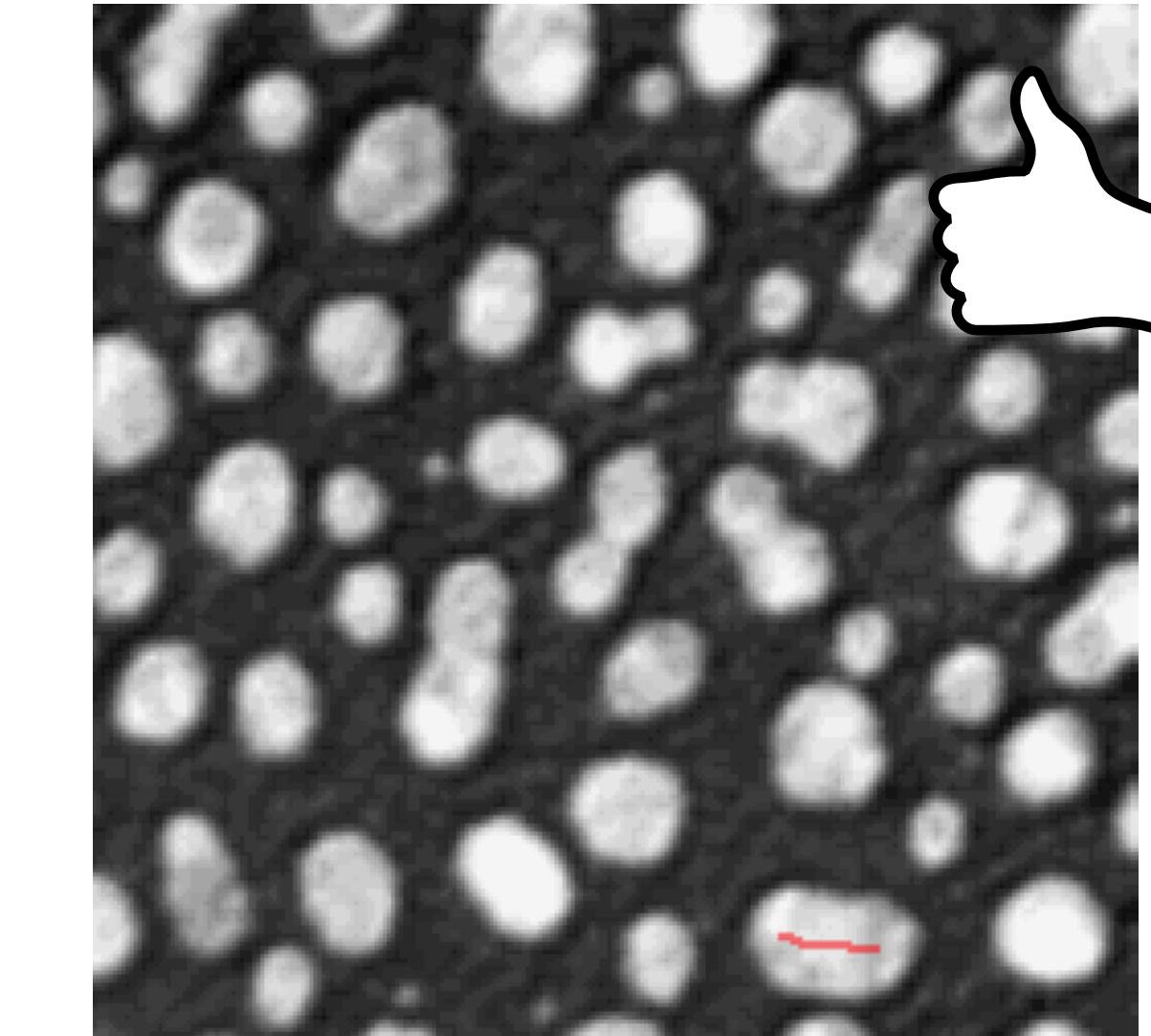
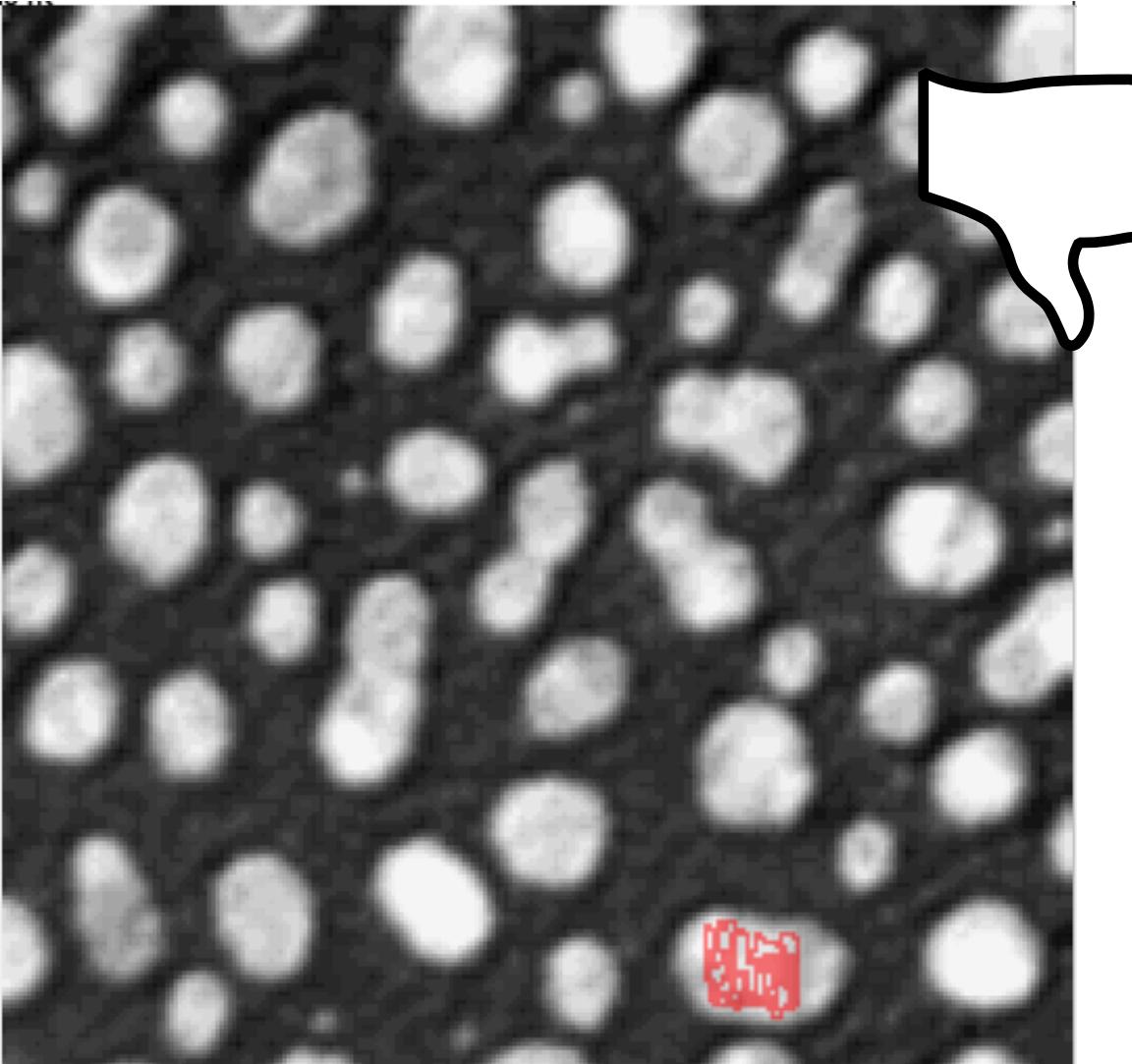
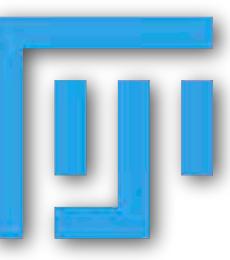
# Labkit Segmentation

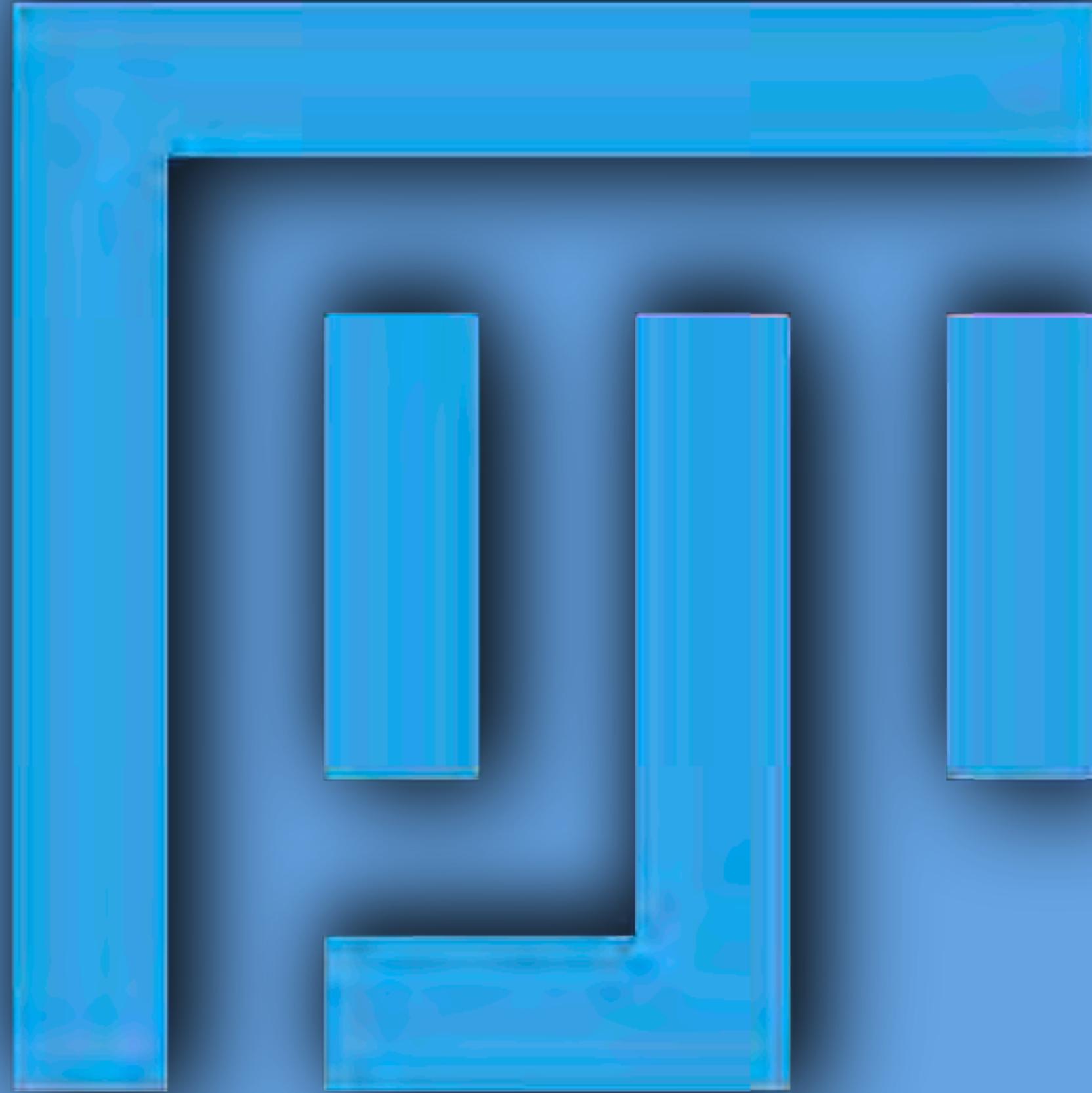


# Labkit Segmentation



# Tips



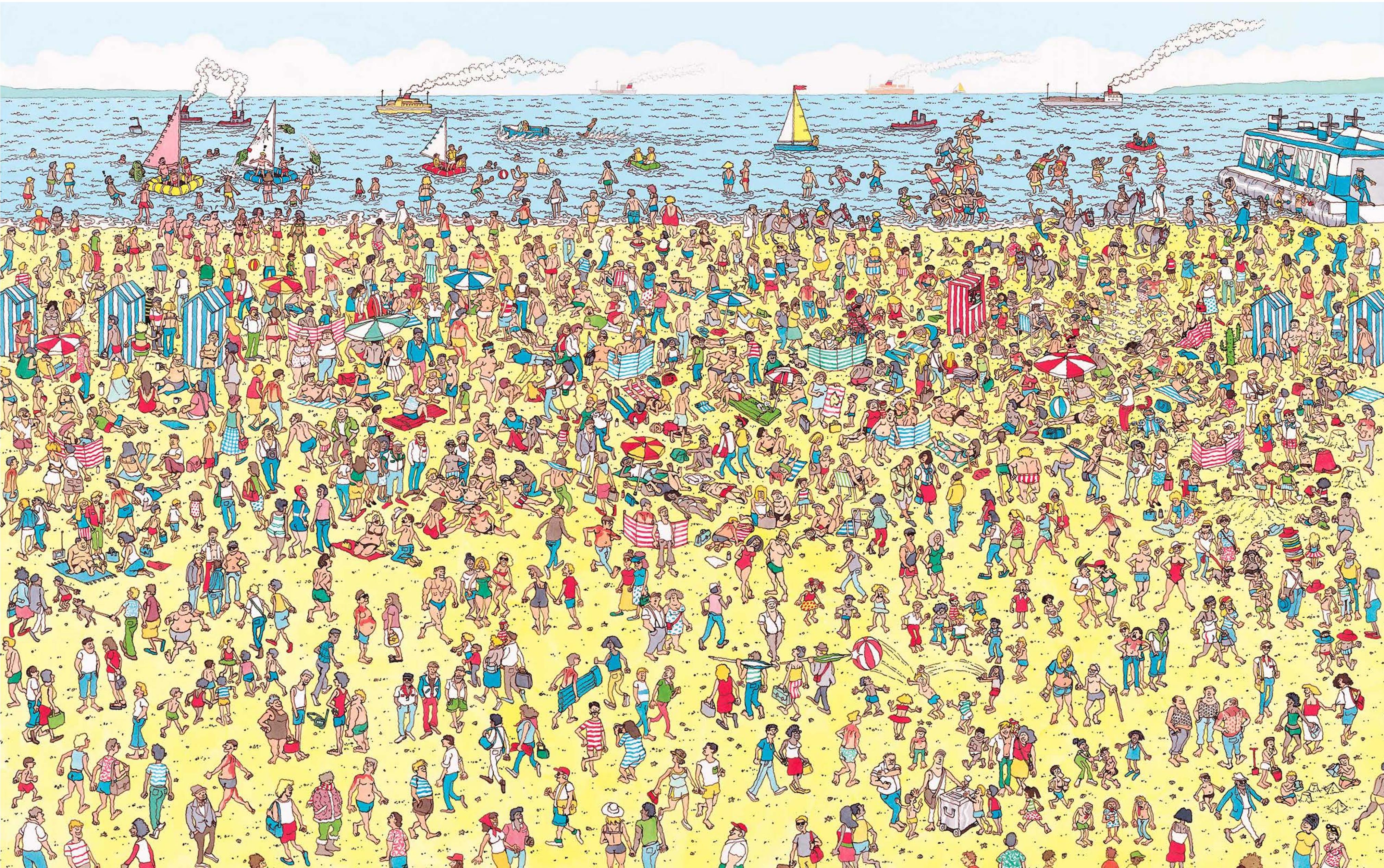


## ***4.3 DAPI segmentation with Labkit***

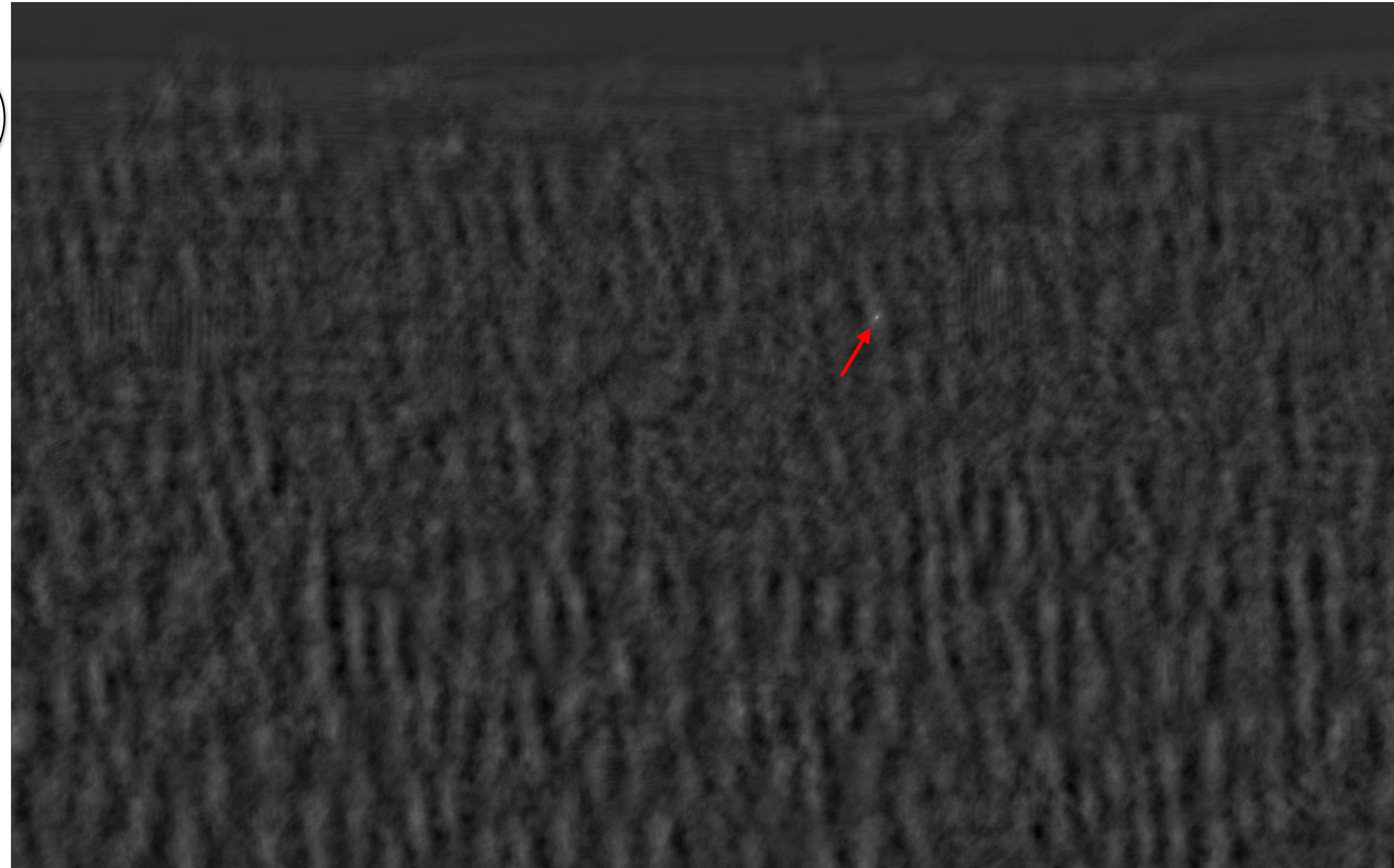
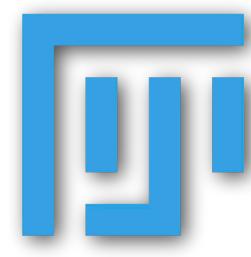
### ***Breakpoint at 12: Create results***

# Filters

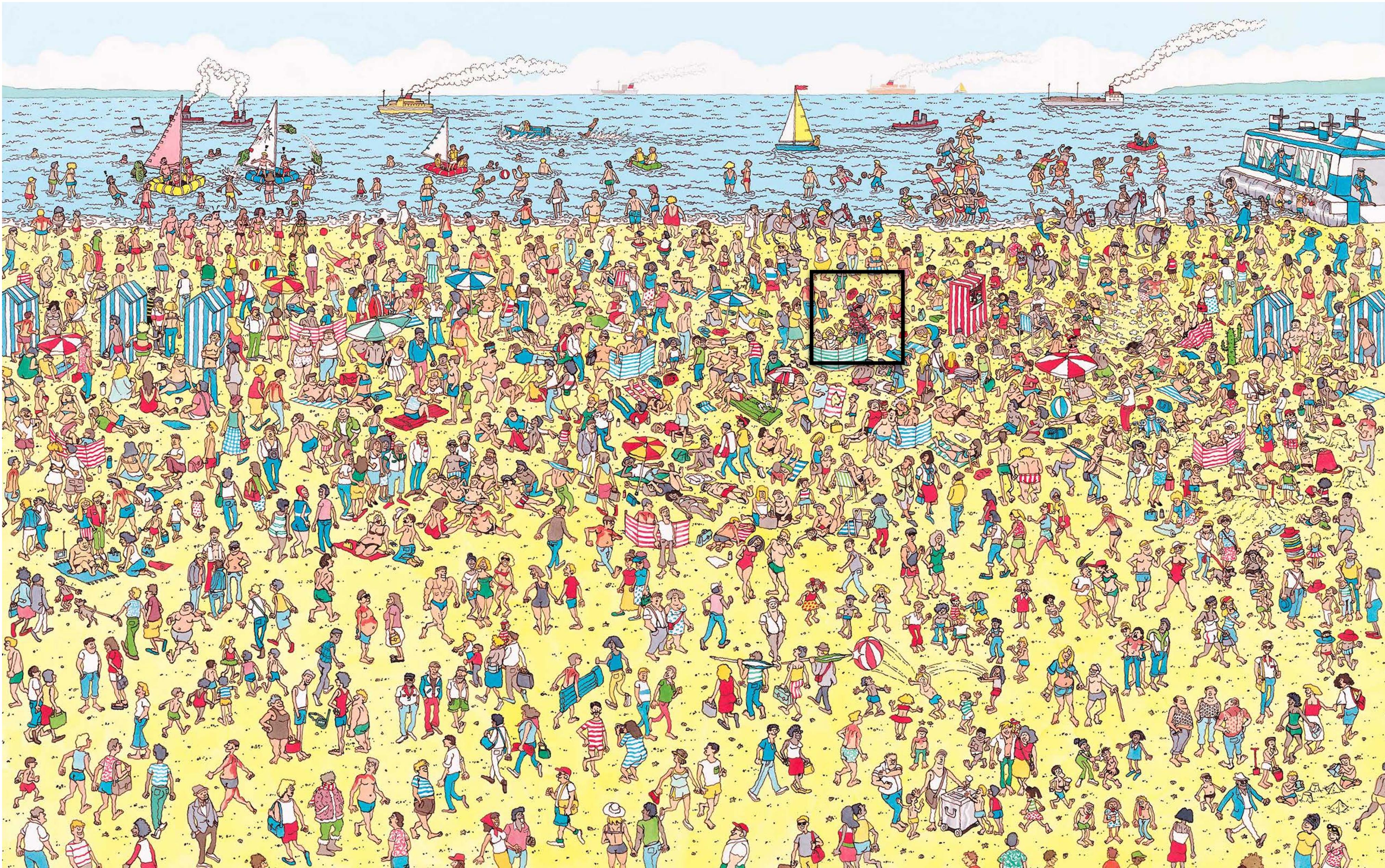
# Where is Waldo?



# Where is Waldo?



# Where is Waldo?



# Filtering

## Neighborhood Transformation

**Linear Transformation**

**Smoothing: Mean, Gaussian**

**Edge detection: Sobel, Canny**

**Non-linear Transformation**

**Smoothing: Median**

**Cleaning: Morphology**

Weights are defined by the kernels

Weighted average = Multiply, add, divide

One neighborhood at a time

Neighborhood defined by the kernel size

**Convolution is weighted average in local neighborhood**

### Terminologies

A **kernel** is a small matrix used in image processing for convolution.

A **filter** is a broader term that refers to any technique that modifies an image.

# Convolution operation

Image

px1	px2	px3	px4	px5	px6	px7	px8	px9
px10	px11	px12	px13	px14	px15	px16	px17	px18
px19	px20	px21	px22	px23	px24	px25	px26	px27
px28	px29	px30	px31	px32	px33	px34	px35	px36
px37	px38	px39	px40	px41	px42	px43	px44	px45
px46	px47	px48	px49	px50	px51	px52	px53	px54
px55	px56	px57	px58	px59	px60	px61	px62	px63
px64	px65	px66	px67	px68	px69	px70	px71	px72
px73	px74	px75	px76	px77	px78	px79	px80	px81

convolution 

k1	k2	k3
k4	k5	k6
k7	k8	k9

$$\text{ksum} = k_1 + k_2 + k_3 + k_4 + k_5 + k_6 + k_7 + k_8 + k_9$$

=

$$\begin{aligned}
 & (\text{px1} * k_1 / \text{ksum}) + \\
 & (\text{px2} * k_2 / \text{ksum}) + \\
 & (\text{px3} * k_3 / \text{ksum}) + \\
 & (\text{px10} * k_4 / \text{ksum}) + \\
 & (\text{px11} * k_5 / \text{ksum}) + \\
 & (\text{px12} * k_6 / \text{ksum}) + \\
 & (\text{px19} * k_7 / \text{ksum}) + \\
 & (\text{px20} * k_8 / \text{ksum}) + \\
 & (\text{px21} * k_9 / \text{ksum}) =
 \end{aligned}$$

c1


# Convolution operation

Image

px1	px2	px3	px4	px5	px6	px7	px8	px9
px10	px11	px12	px13	px14	px15	px16	px17	px18
px19	px20	px21	px22	px23	px24	px25	px26	px27
px28	px29	px30	px31	px32	px33	px34	px35	px36
px37	px38	px39	px40	px41	px42	px43	px44	px45
px46	px47	px48	px49	px50	px51	px52	px53	px54
px55	px56	px57	px58	px59	px60	px61	px62	px63
px64	px65	px66	px67	px68	px69	px70	px71	px72
px73	px74	px75	px76	px77	px78	px79	px80	px81

convolution  
★

k1	k2	k3
k4	k5	k6
k7	k8	k9

3 x 3 kernel

$$\begin{aligned}
 & (\text{px2} * \text{k1} / \text{ksum}) + \\
 & (\text{px3} * \text{k2} / \text{ksum}) + \\
 & (\text{px4} * \text{k3} / \text{ksum}) + \\
 & (\text{px11} * \text{k4} / \text{ksum}) + \\
 & (\text{px12} * \text{k5} / \text{ksum}) + \\
 & (\text{px13} * \text{k6} / \text{ksum}) + \\
 & (\text{px20} * \text{k7} / \text{ksum}) + \\
 & (\text{px21} * \text{k8} / \text{ksum}) + \\
 & (\text{px22} * \text{k9} / \text{ksum}) = \\
 \end{aligned}$$

c2

$$\text{ksum} = \text{k1} + \text{k2} + \text{k3} + \text{k4} + \text{k5} + \text{k6} + \text{k7} + \text{k8} + \text{k9}$$

	c1	?						

# Convolution operation

Image

px1	px2	px3	px4	px5	px6	px7	px8	px9
px10	px11	px12	px13	px14	px15	px16	px17	px18
px19	px20	px21	px22	px23	px24	px25	px26	px27
px28	px29	px30	px31	px32	px33	px34	px35	px36
px37	px38	px39	px40	px41	px42	px43	px44	px45
px46	px47	px48	px49	px50	px51	px52	px53	px54
px55	px56	px57	px58	px59	px60	px61	px62	px63
px64	px65	px66	px67	px68	px69	px70	px71	px72
px73	px74	px75	px76	px77	px78	px79	px80	px81

convolution  
★

3 x 3 kernel

k1	k2	k3
k4	k5	k6
k7	k8	k9

$$\text{ksum} = k_1 + k_2 + k_3 + k_4 + k_5 + k_6 + k_7 + k_8 + k_9$$

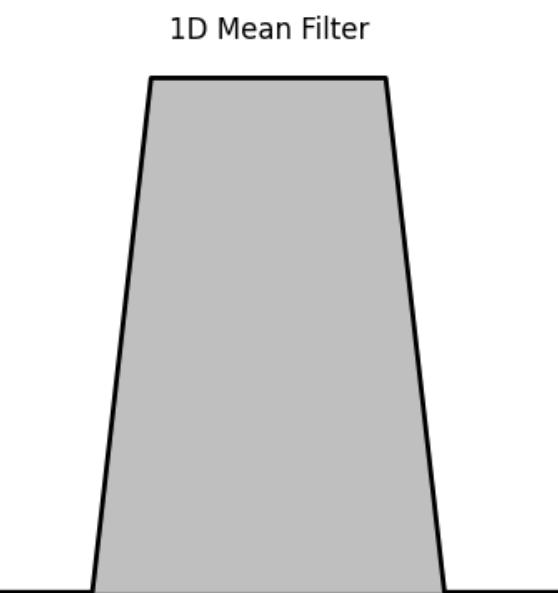
$$\begin{aligned}
 & (\text{px}_3 * \text{k}_1 / \text{ksum}) + \\
 & (\text{px}_4 * \text{k}_2 / \text{ksum}) + \\
 & (\text{px}_5 * \text{k}_3 / \text{ksum}) + \\
 & (\text{px}_{12} * \text{k}_4 / \text{ksum}) + \\
 & (\text{px}_{13} * \text{k}_5 / \text{ksum}) + \\
 & (\text{px}_{14} * \text{k}_6 / \text{ksum}) + \\
 & (\text{px}_{21} * \text{k}_7 / \text{ksum}) + \\
 & (\text{px}_{22} * \text{k}_8 / \text{ksum}) + \\
 & (\text{px}_{23} * \text{k}_9 / \text{ksum}) =
 \end{aligned}$$

c<sub>3</sub>

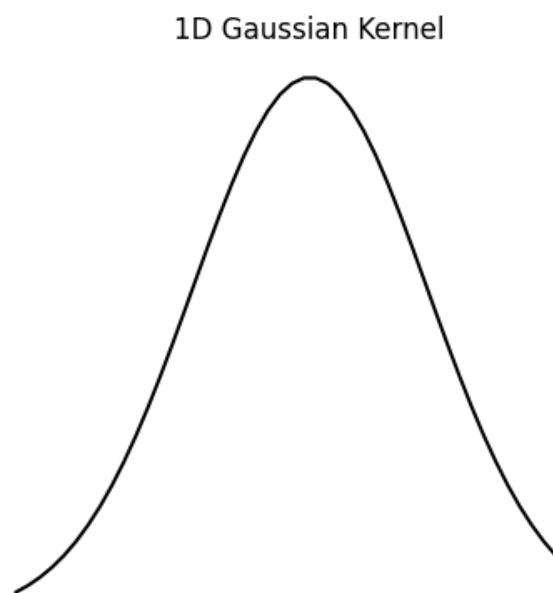
	c <sub>1</sub>	c <sub>2</sub>	?					

# Different Types of Kernel

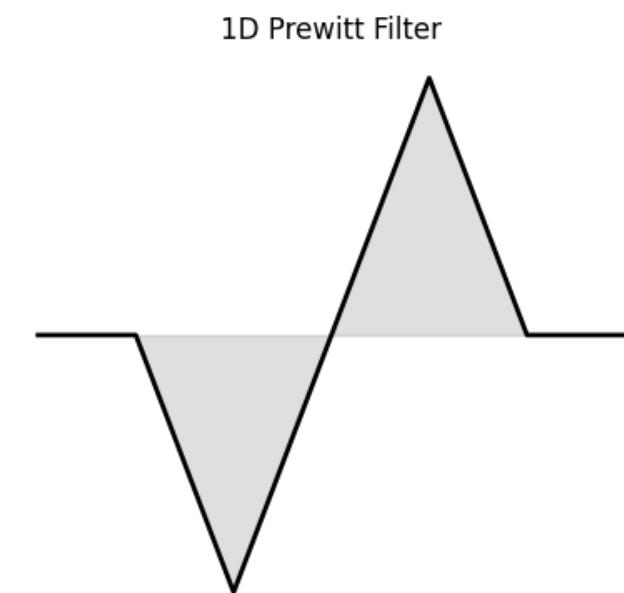
0	0	0	0	0	0
0	1	1	1	1	0
0	1	1	1	1	0
0	1	1	1	1	0
0	0	0	0	0	0



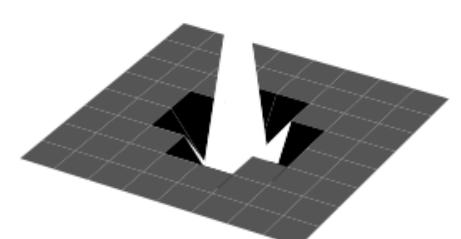
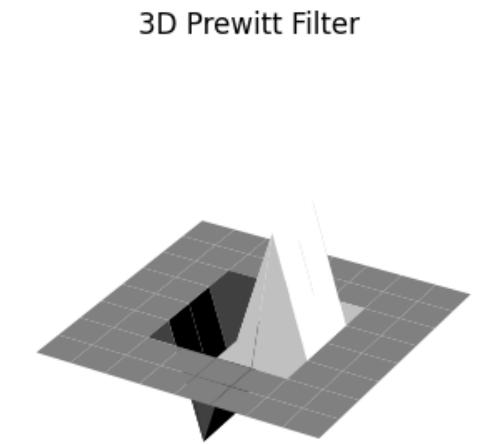
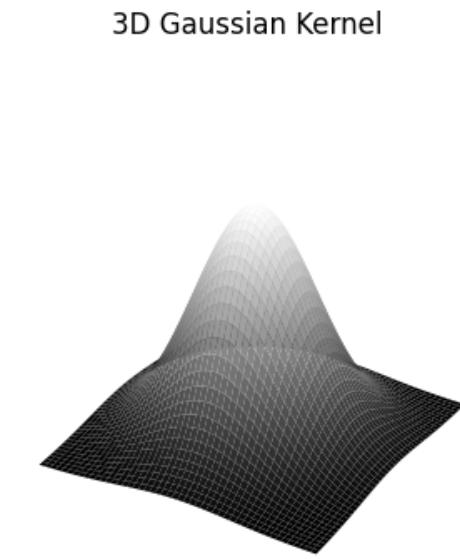
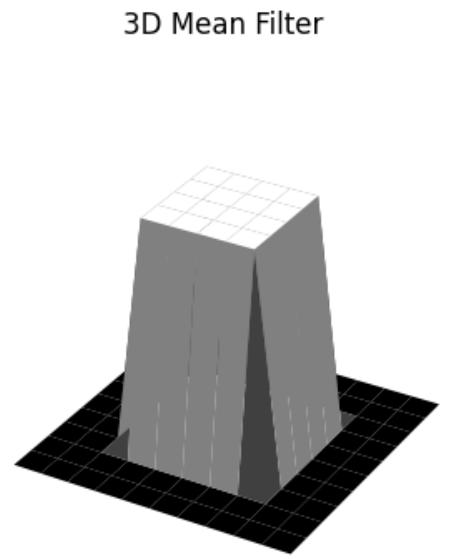
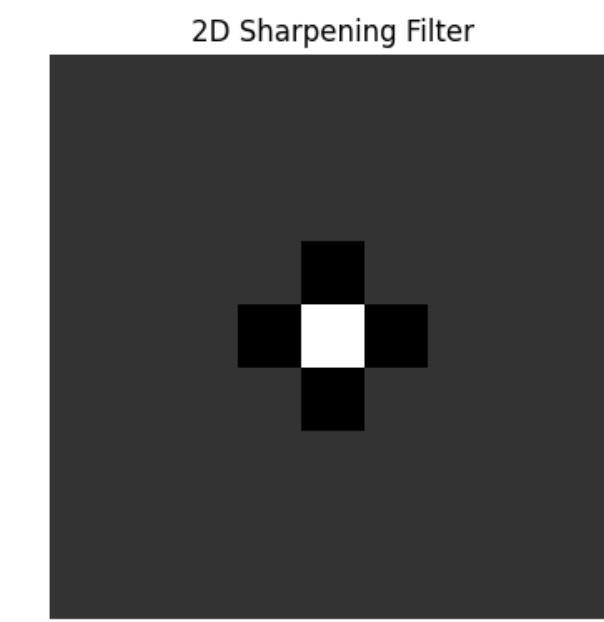
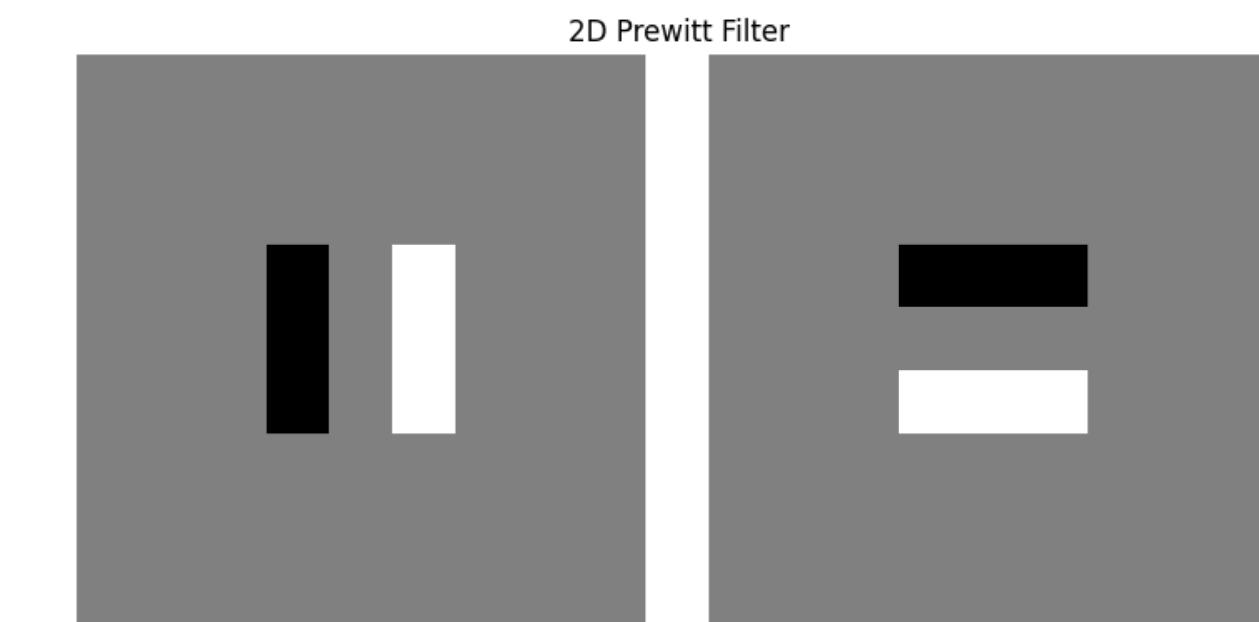
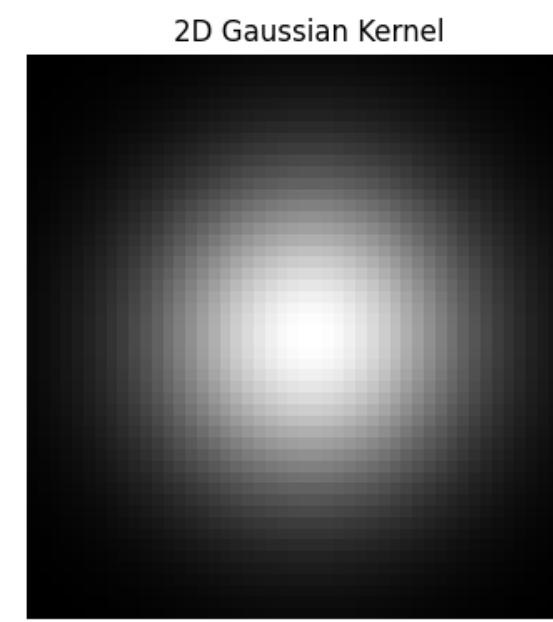
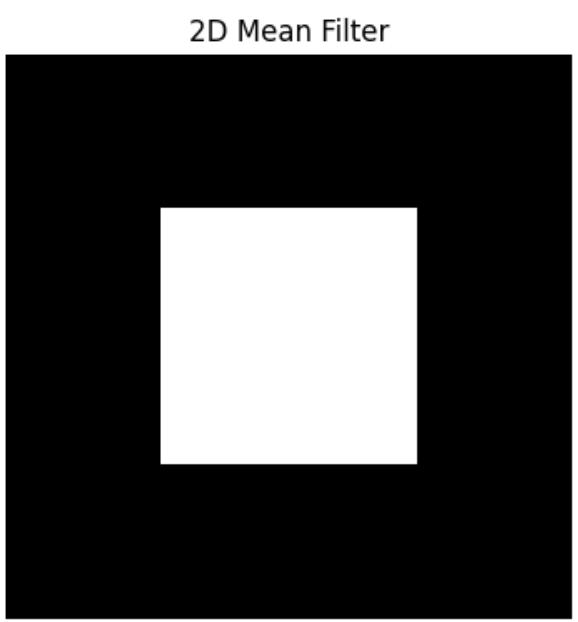
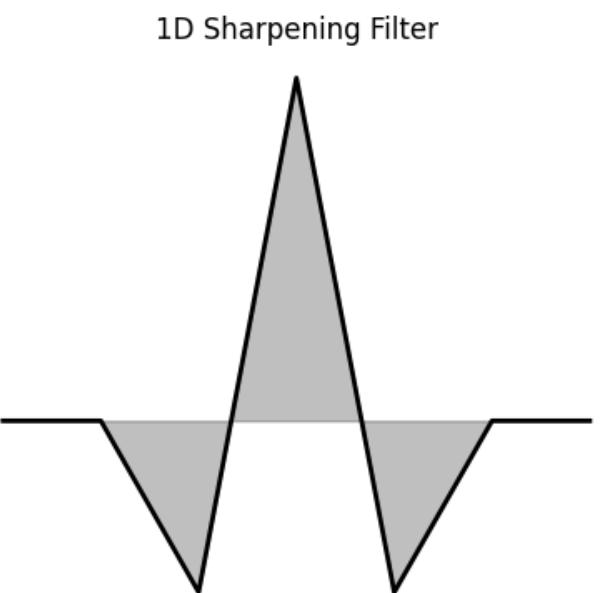
0	0	0	0	0	0
0	1	2	1	0	0
0	2	5	2	0	0
0	1	2	1	0	0
0	0	0	0	0	0



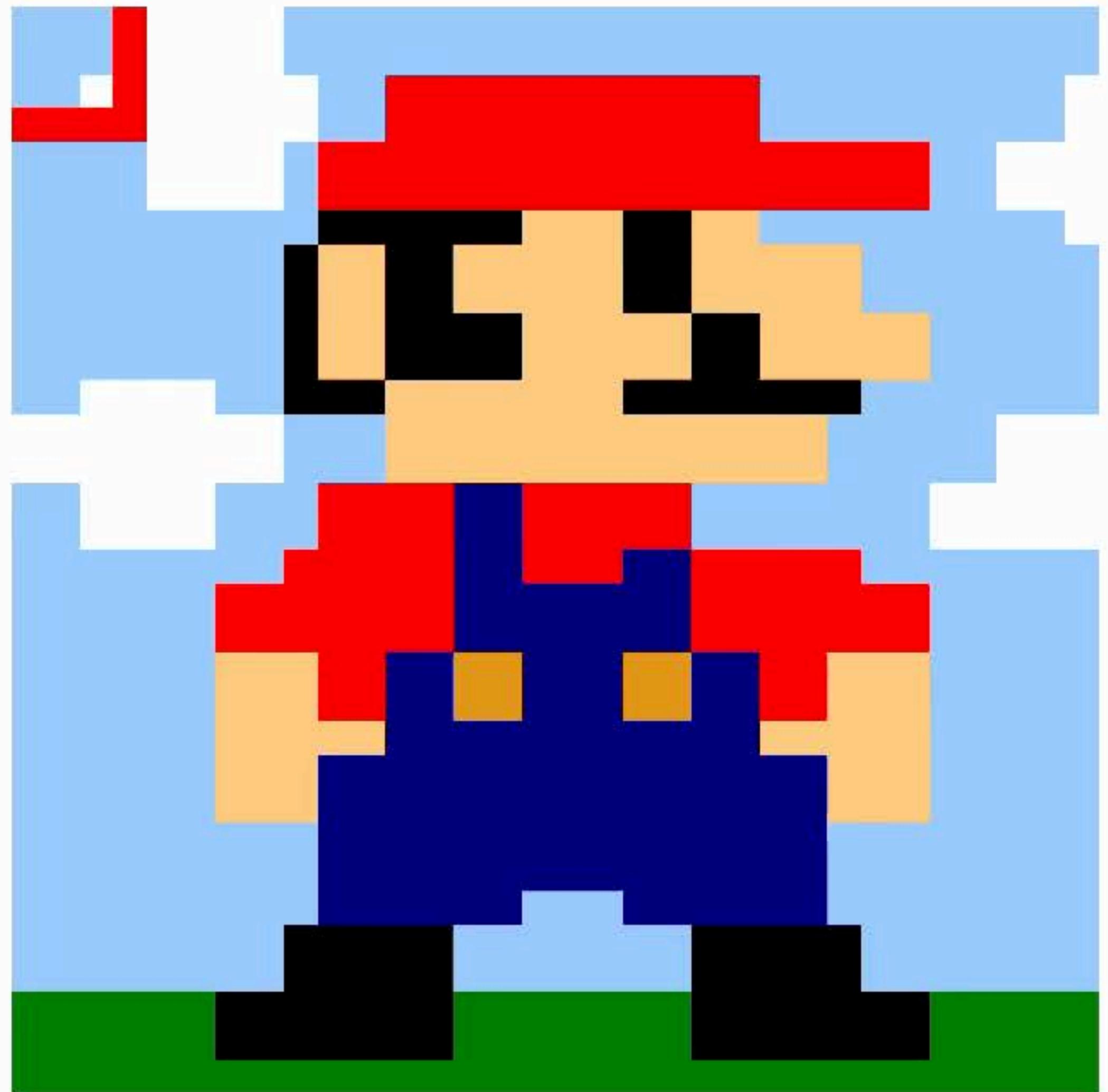
0	0	0	0	0	0
0	-1	0	1	0	0
0	-1	0	1	0	0
0	-1	0	1	0	0
0	1	1	1	0	0



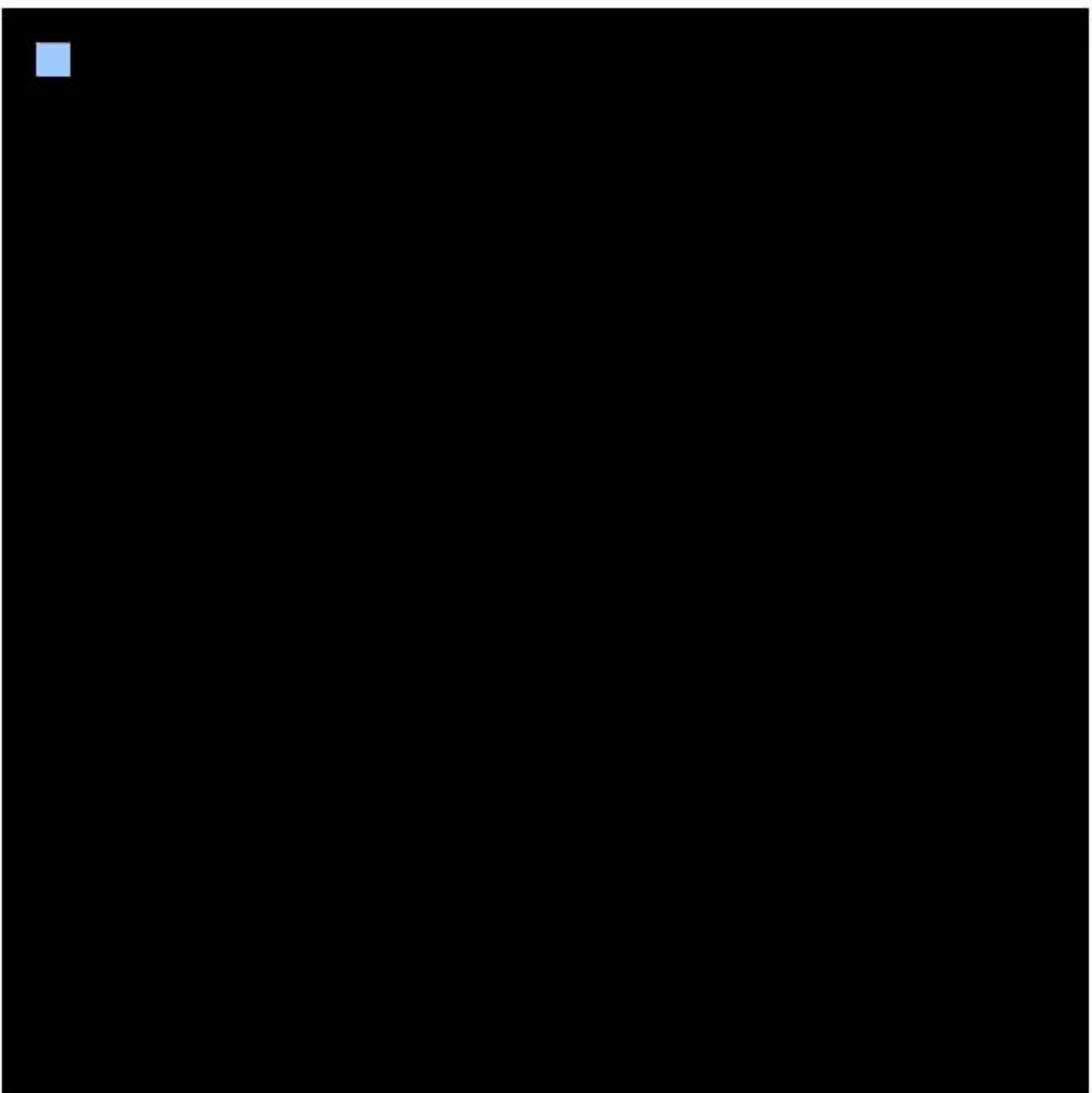
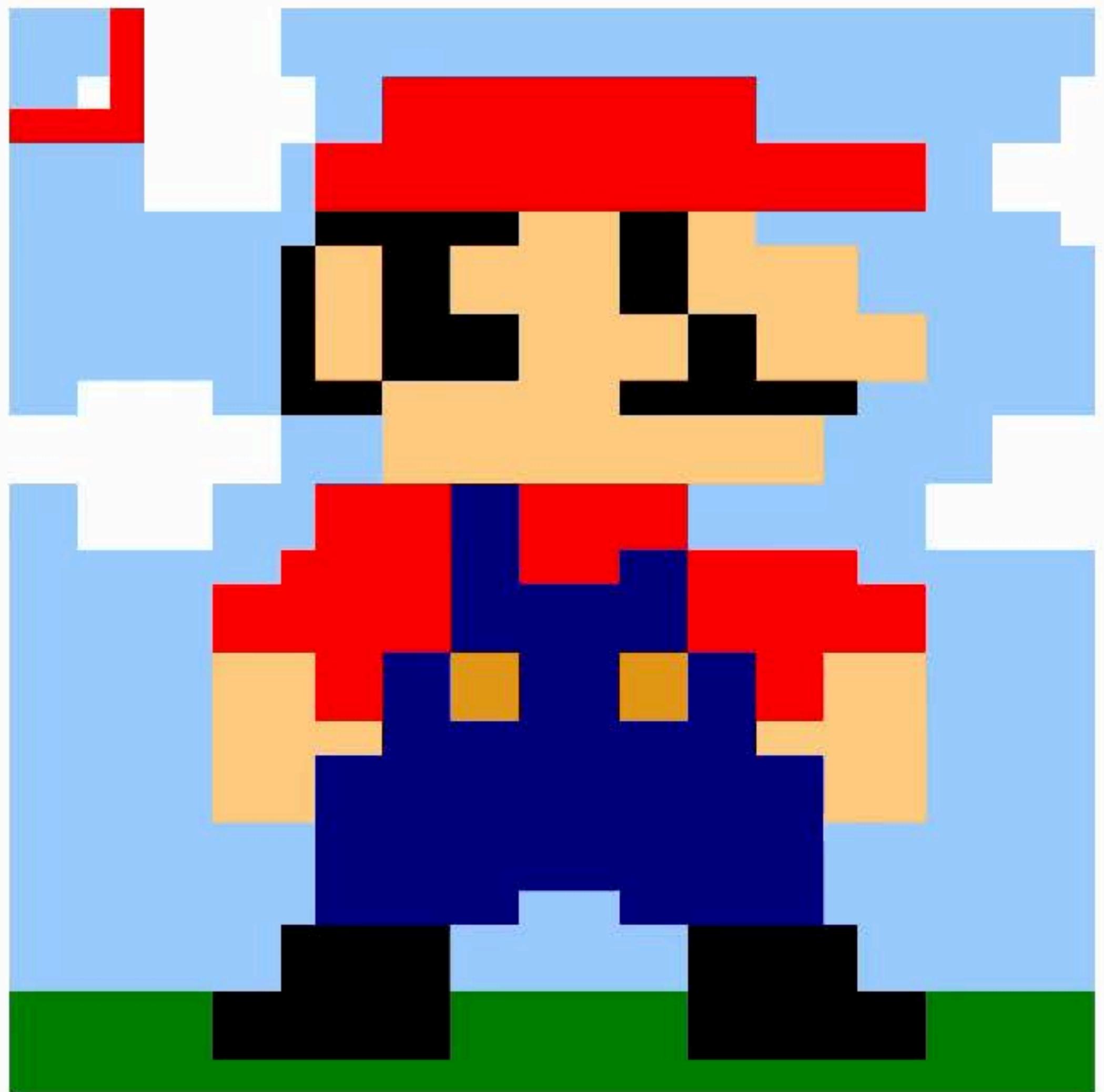
0	1	-1	1	0	0
0	-1	2	-1	0	0
0	1	-1	1	0	0
0	0	0	0	0	0



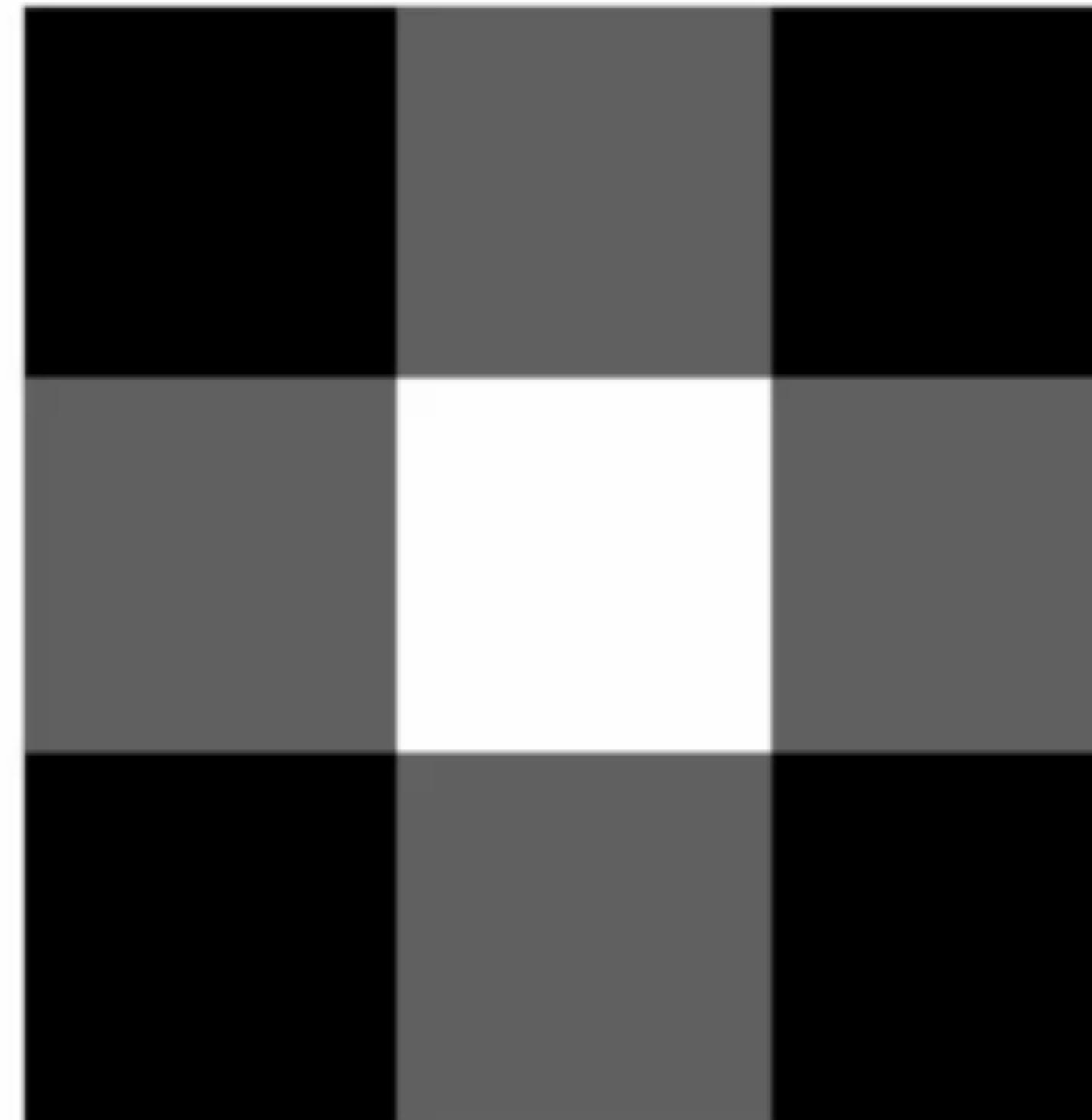
# Mean Filter



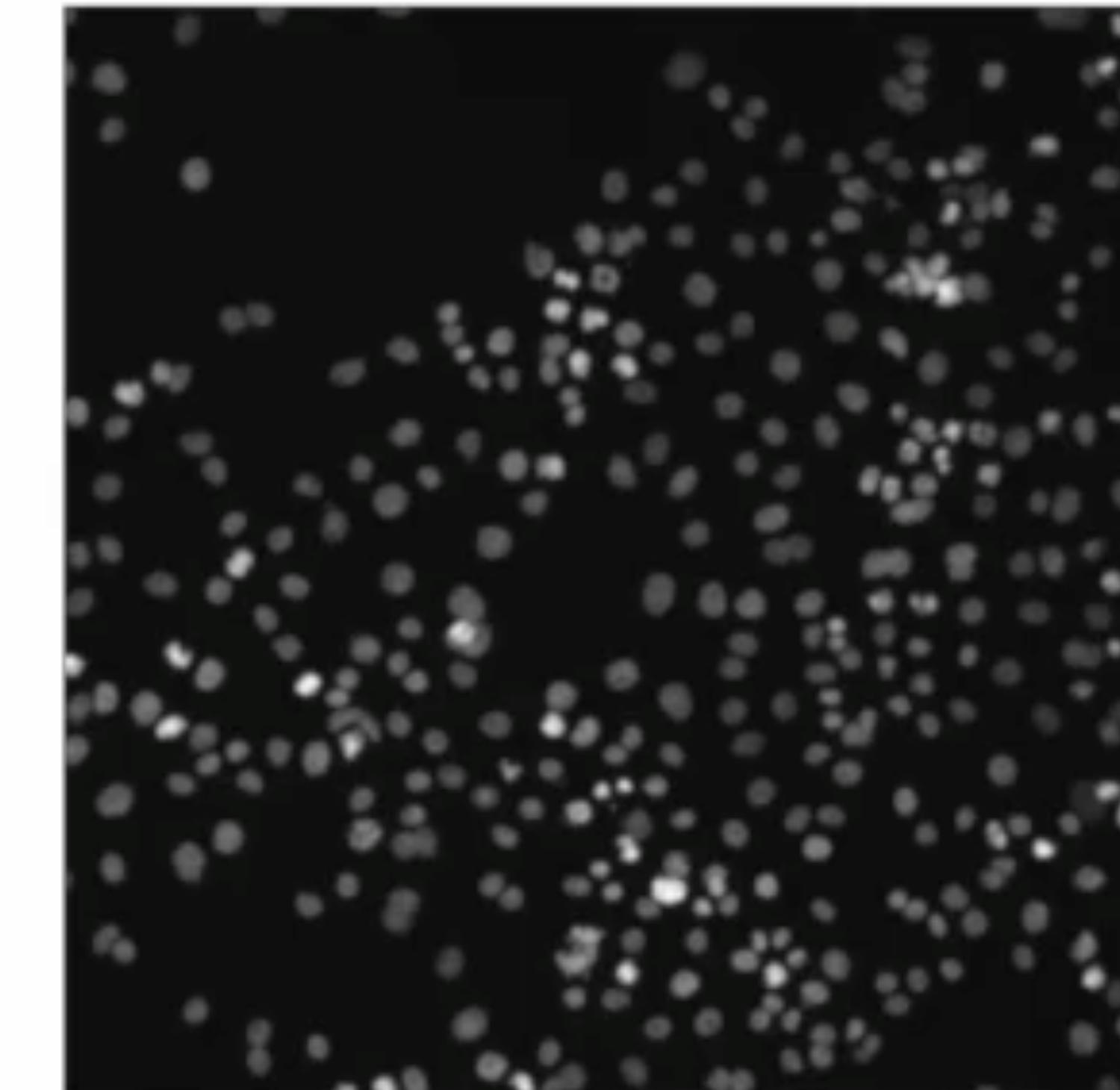
# Gaussian Filter



# Effects of Different Gaussian Kernel



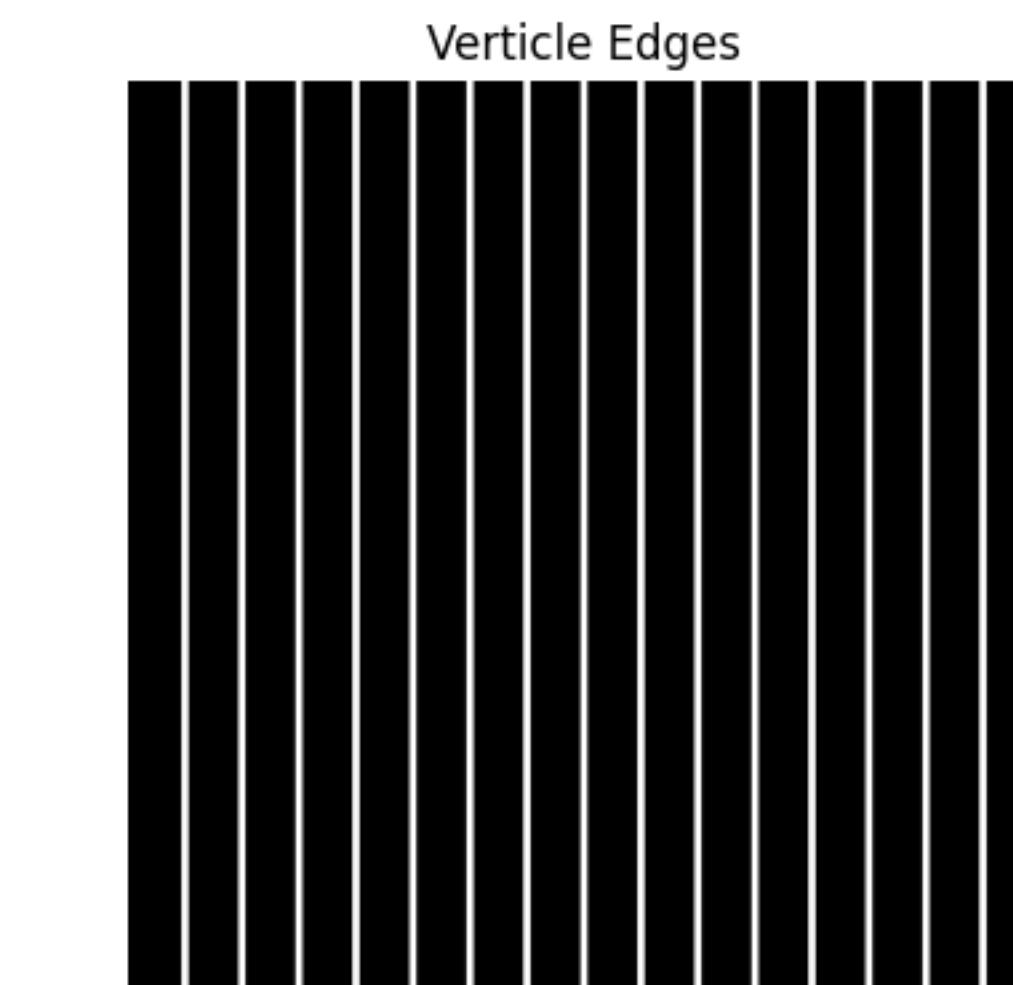
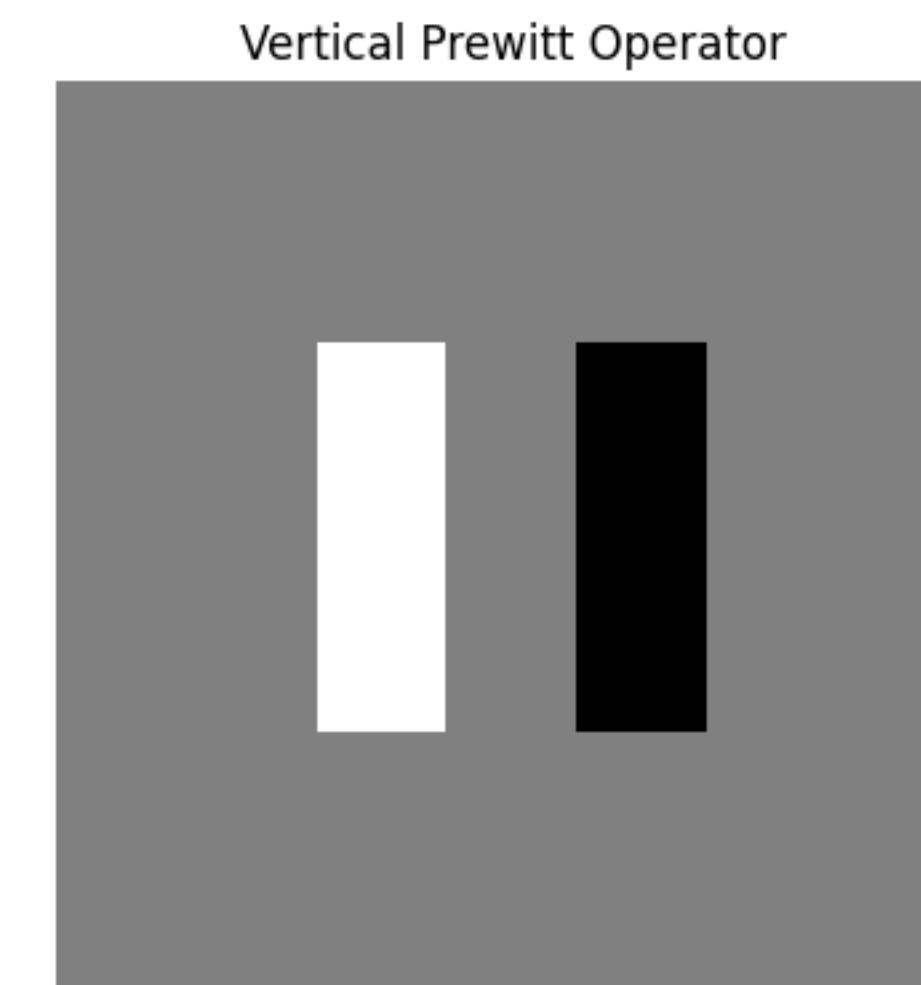
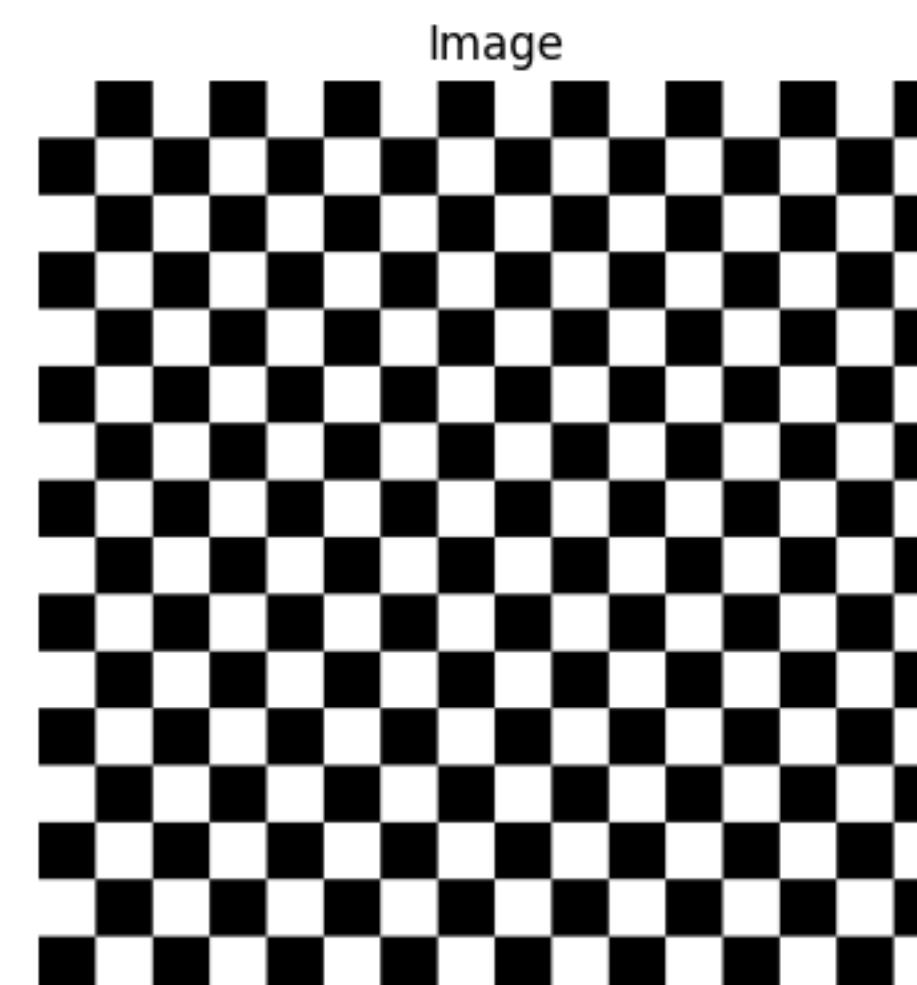
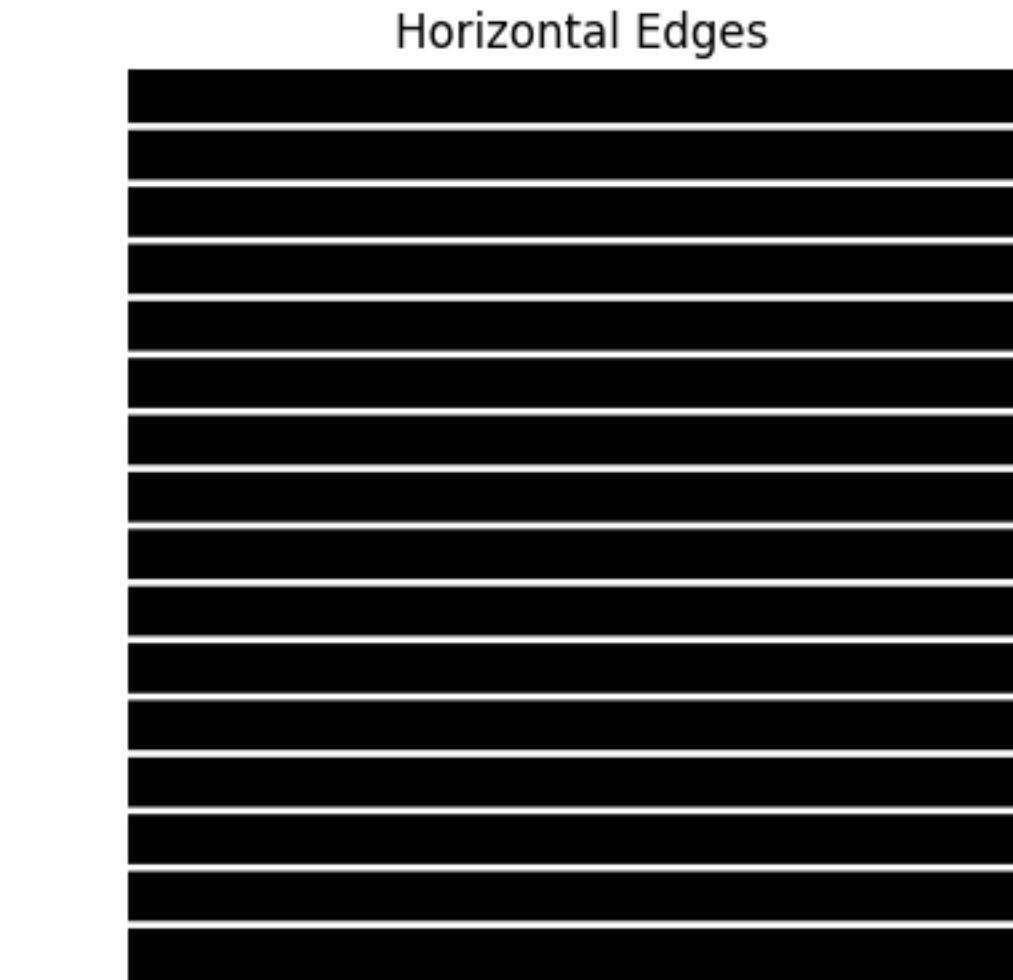
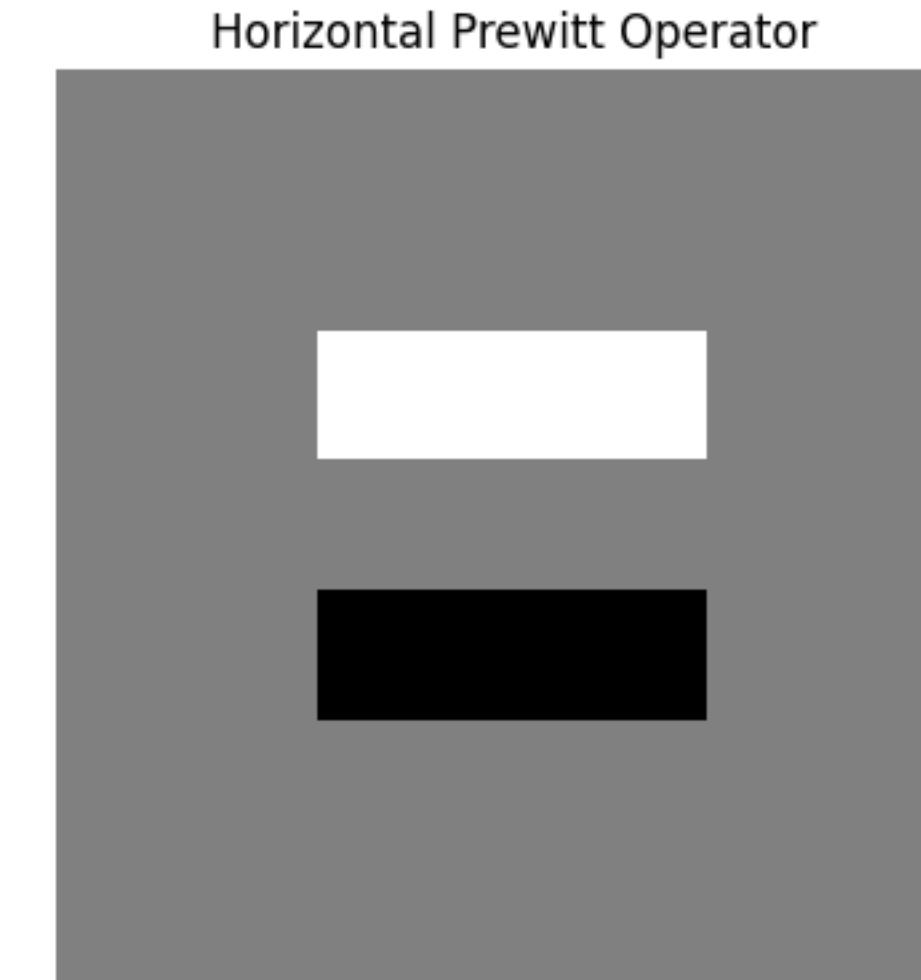
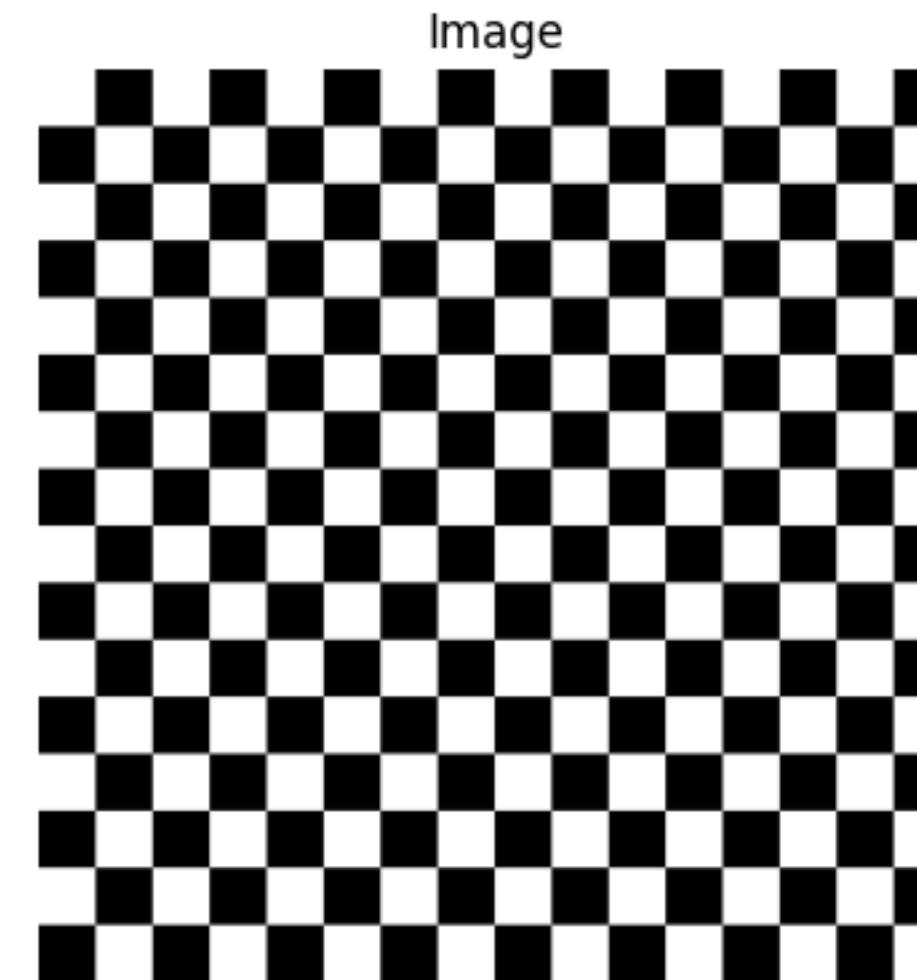
Kernel



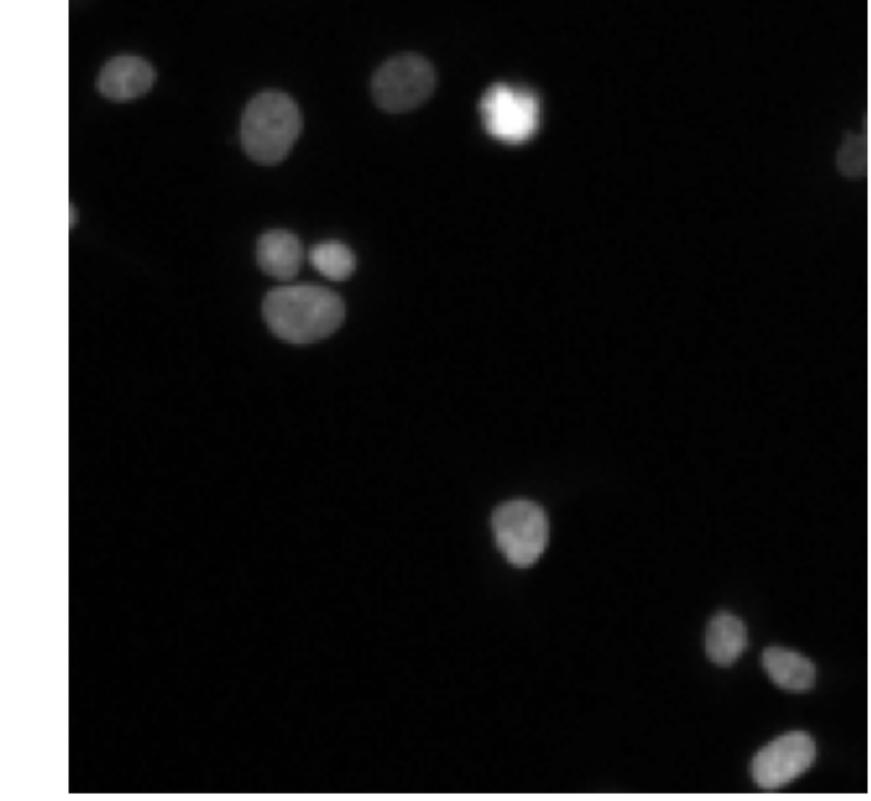
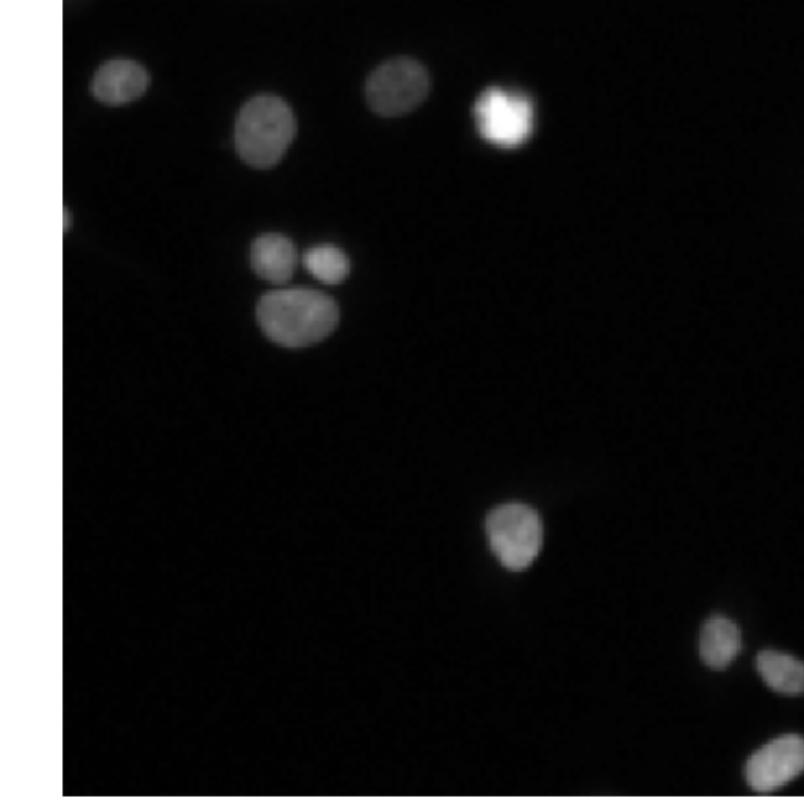
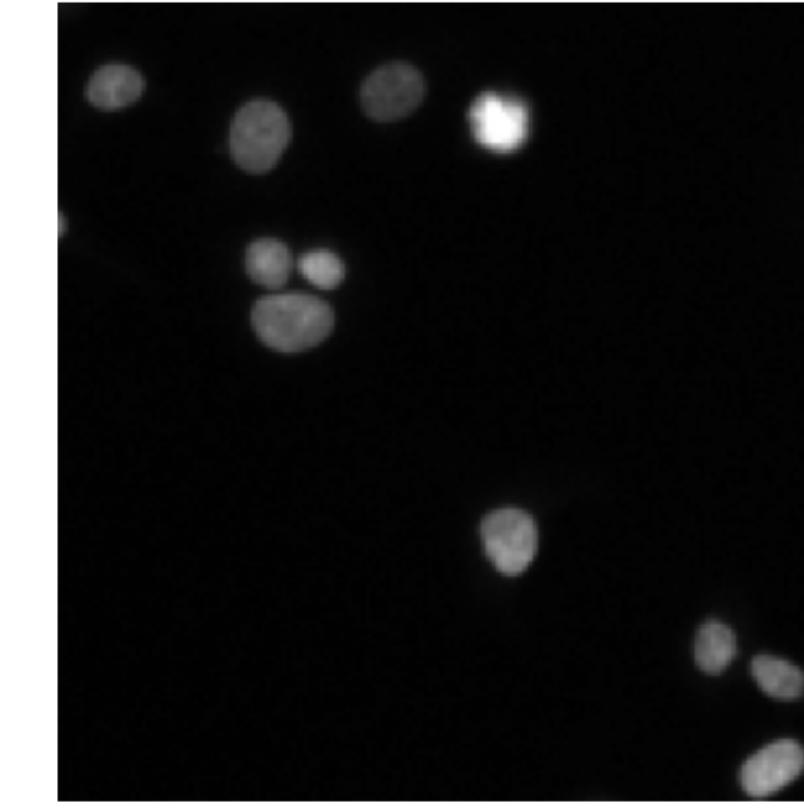
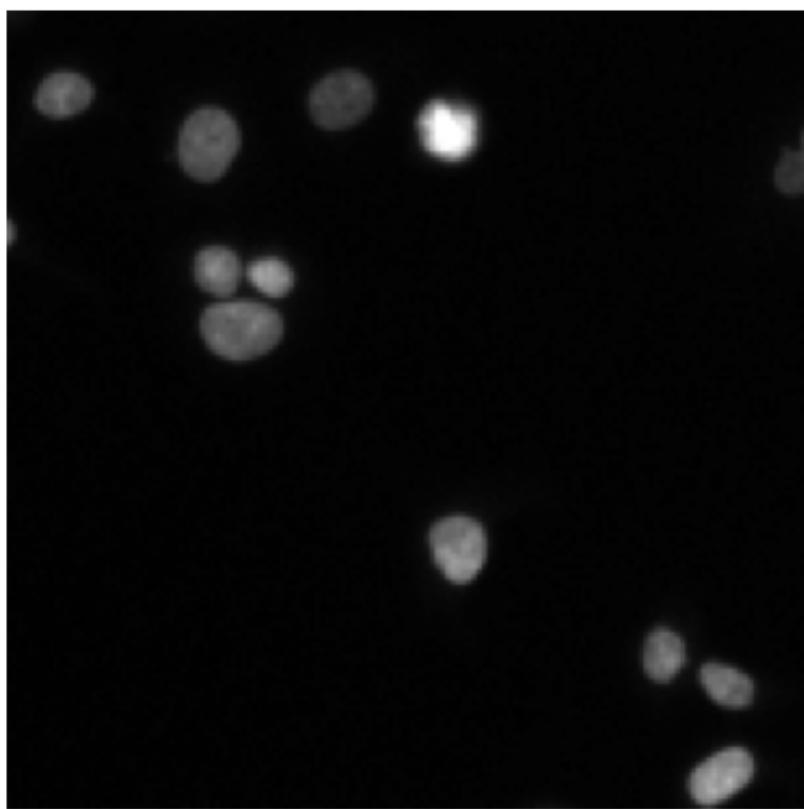
Blurred image

# Edge Filters

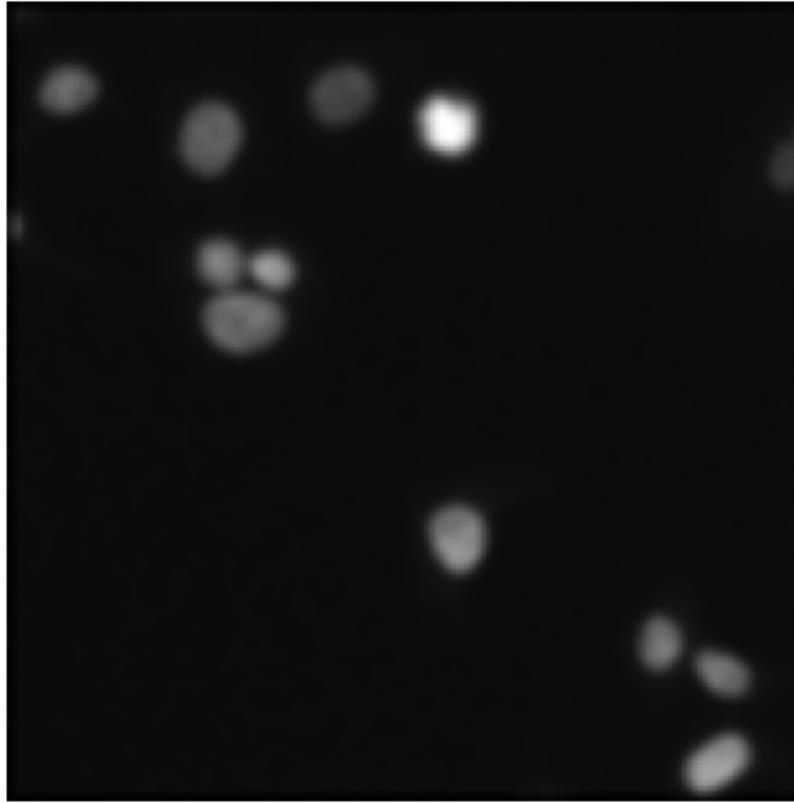
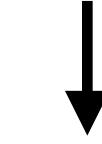
## Prewitt Operator



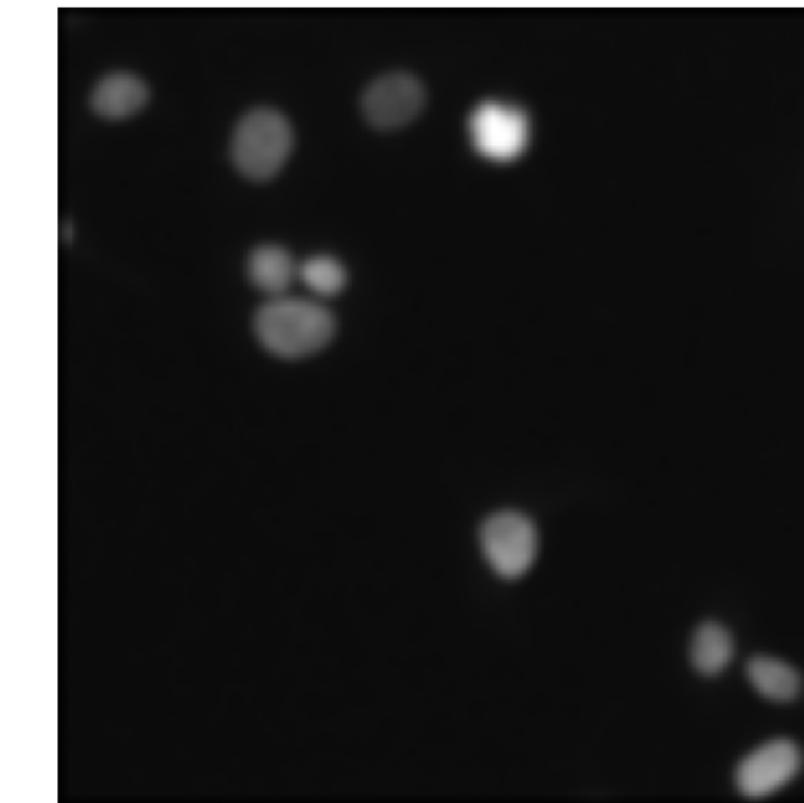
# Different Filters in Action



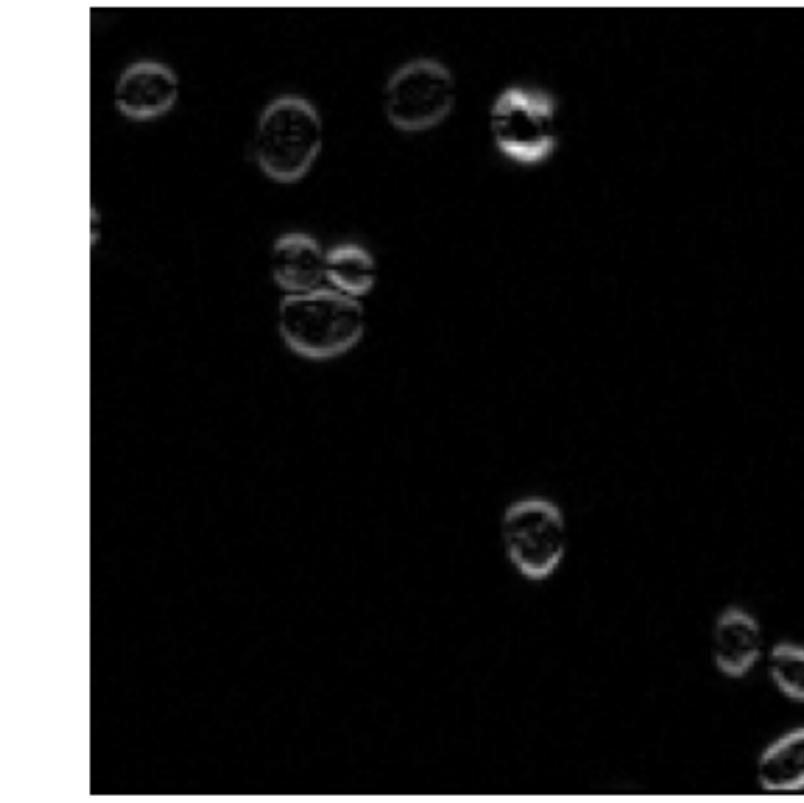
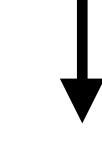
0	0	0	0	0	0
0	1	1	1	1	0
0	1	1	1	1	0
0	1	1	1	1	0
0	0	0	0	0	0



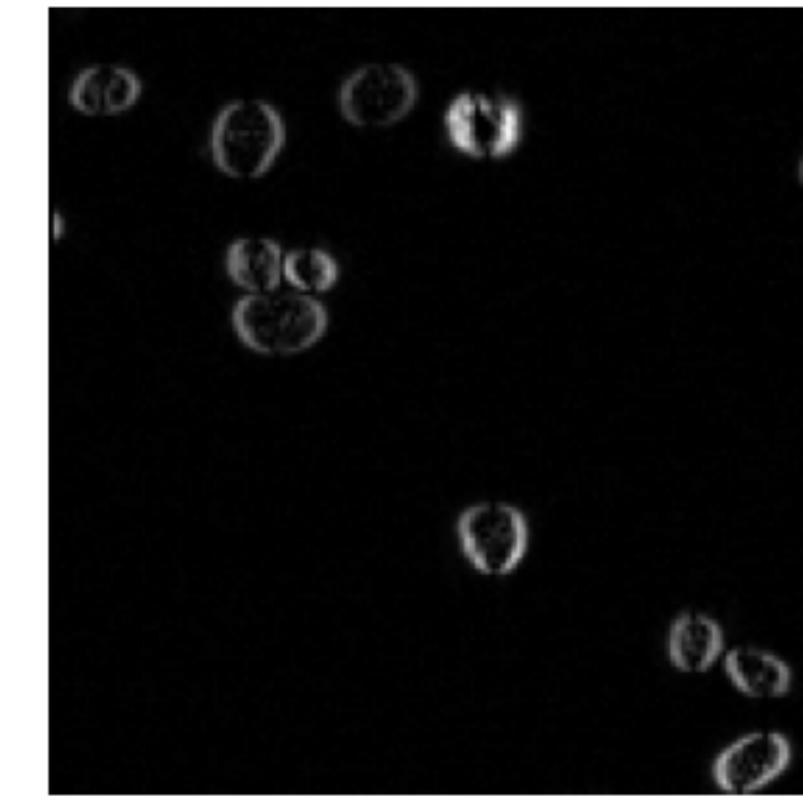
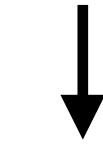
0	0	0	0	0	0
0	1	2	1	0	0
0	2	5	2	0	0
0	1	2	1	0	0
0	0	0	0	0	0



0	0	0	0	0	0
0	-1	-1	-1	0	0
0	0	0	0	0	0
0	1	1	1	0	0
0	0	0	0	0	0



0	0	0	0	0	0
0	1	-1	1	0	0
0	-1	2	-1	0	0
0	1	-1	1	0	0
0	0	0	0	0	0



Mean Filter

Gaussian Filter

Prewitt Filter

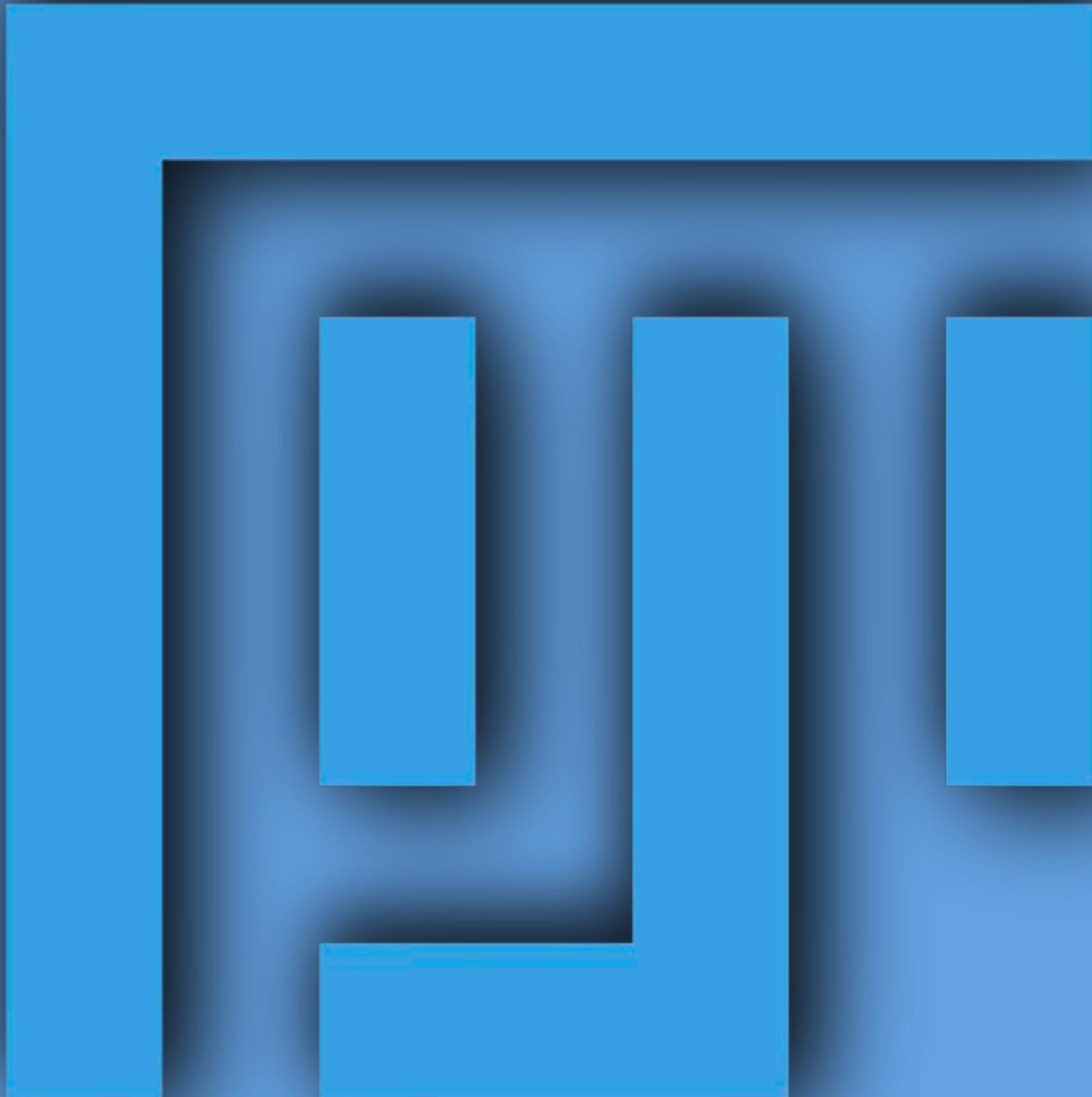
Sharpening Filter

# Filtering — exercises



- 3.1 edge filters - vertical stripes**
- 3.2 edge filters - horizontal stripes**
- 3.3 edge filters - checkerboard**

# Filtering — exercises

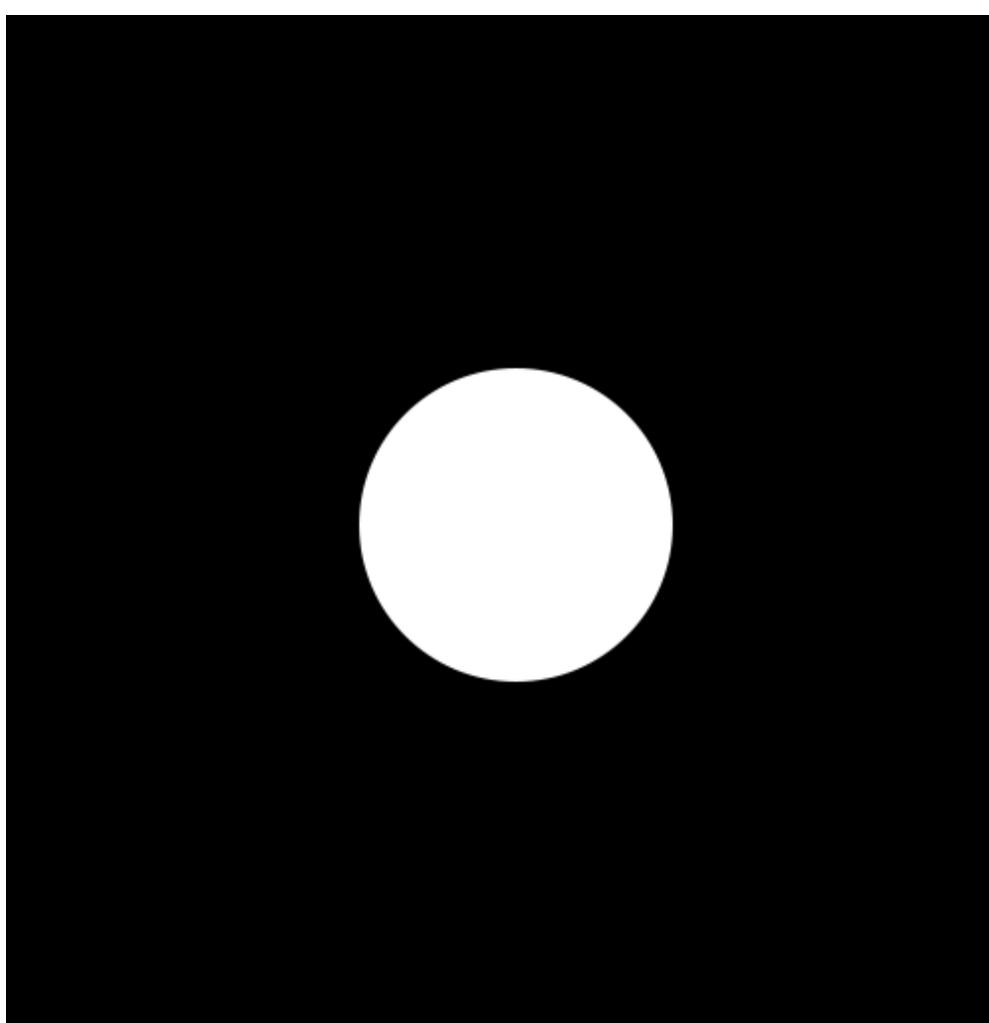


## 4.2 DAPI segmentation with filters and thresholding

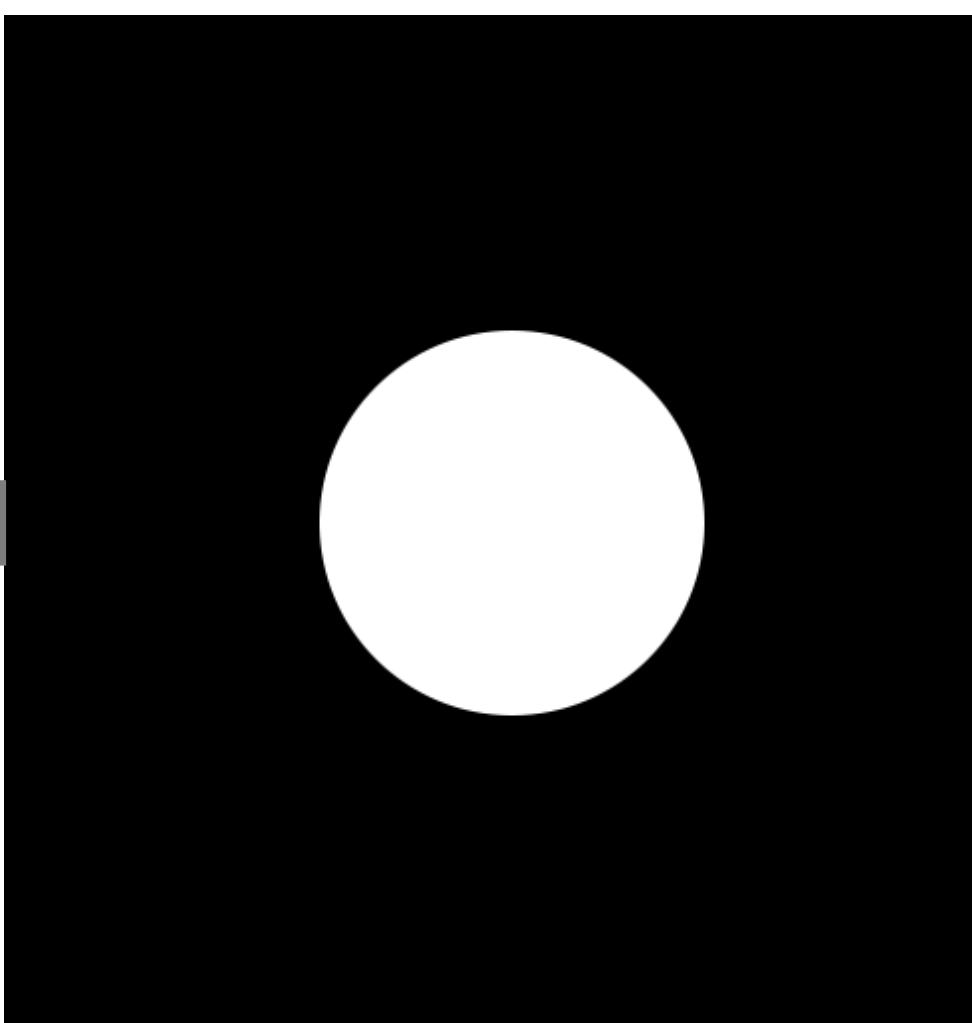
# Morphological Filters

# Two Primary Morphological Operations

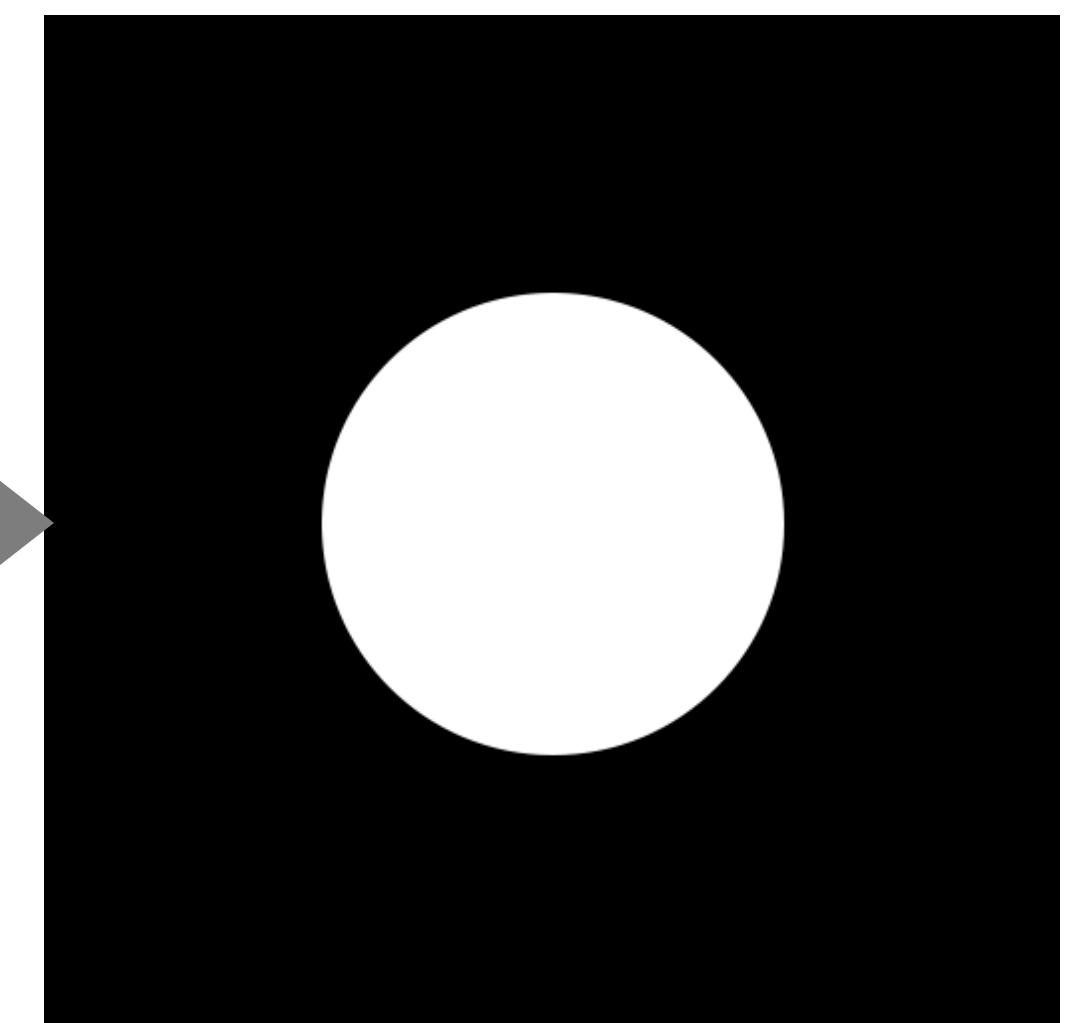
Erosion



Erosion

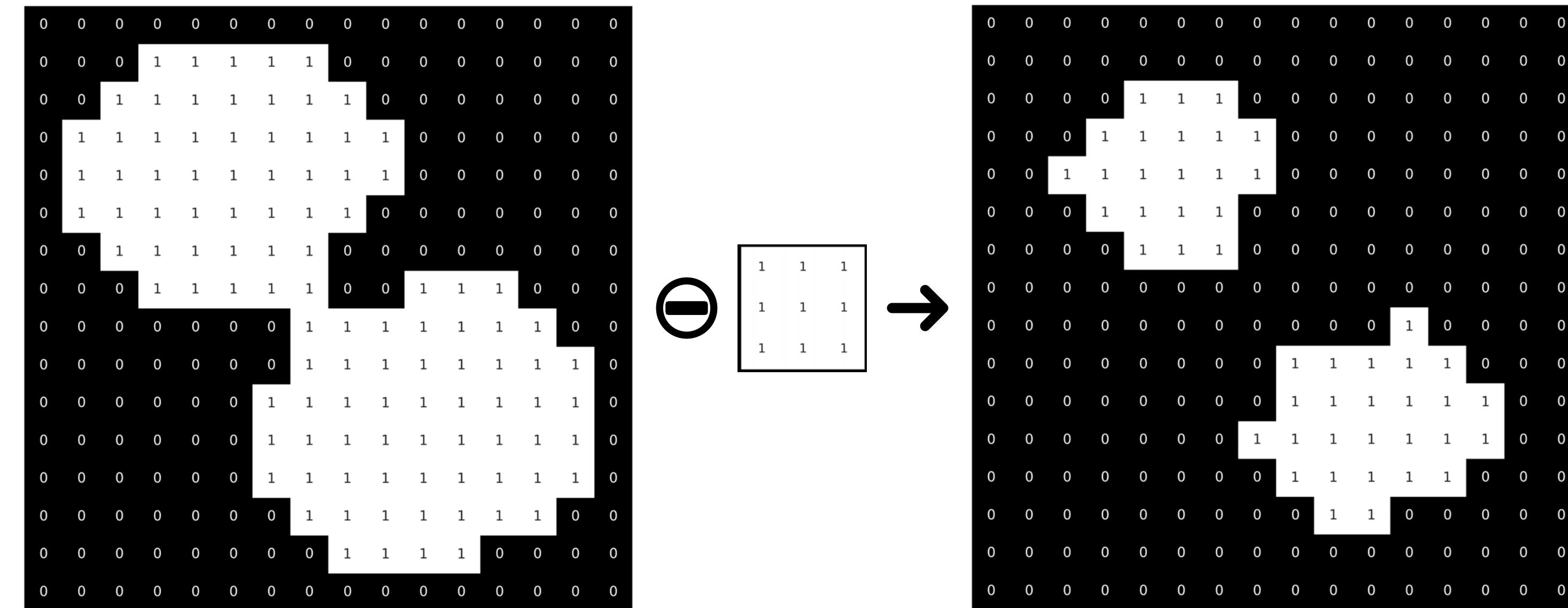


Dilation

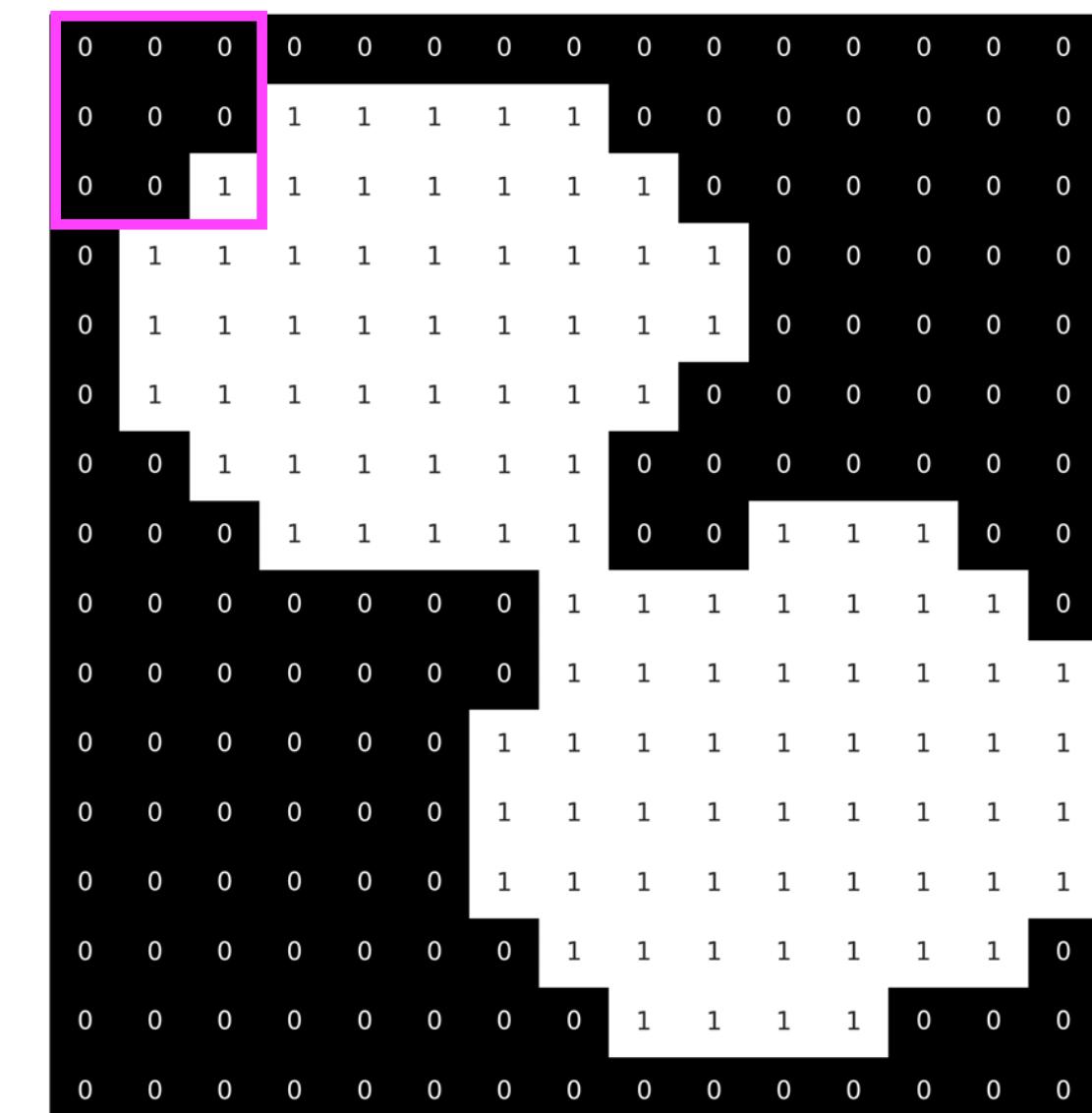


Dilation

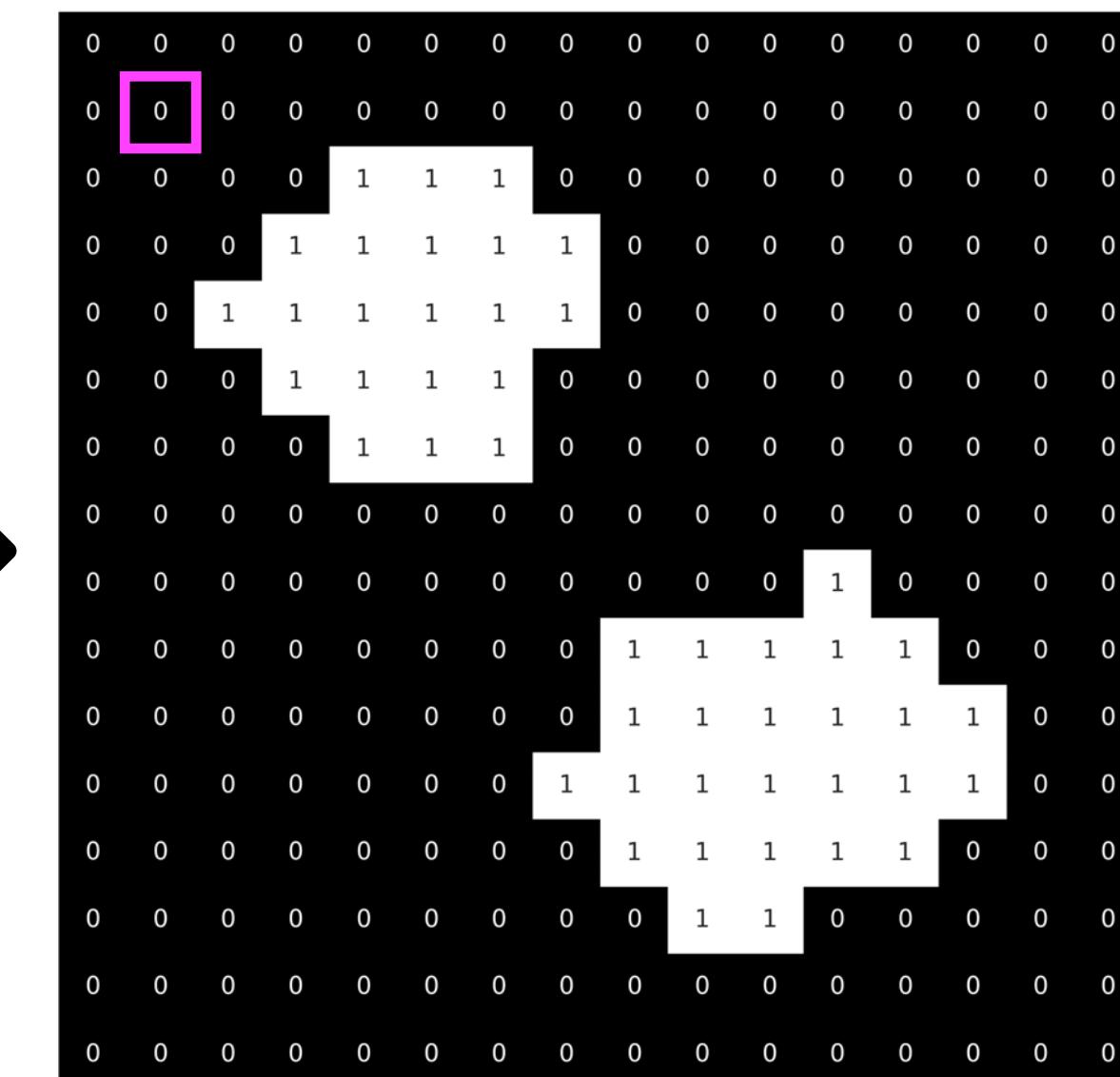
# Erosion Operation



# Erosion Operation

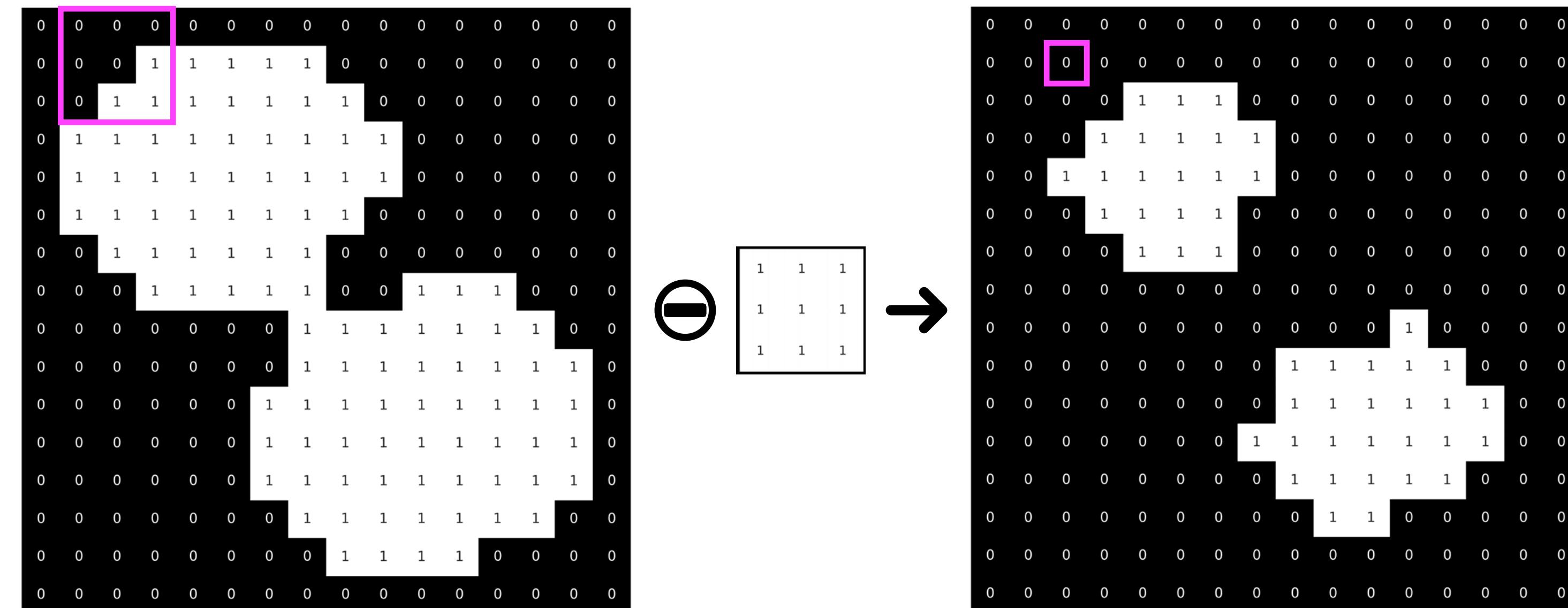


$$\ominus$$



$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \ominus \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \rightarrow 0$$

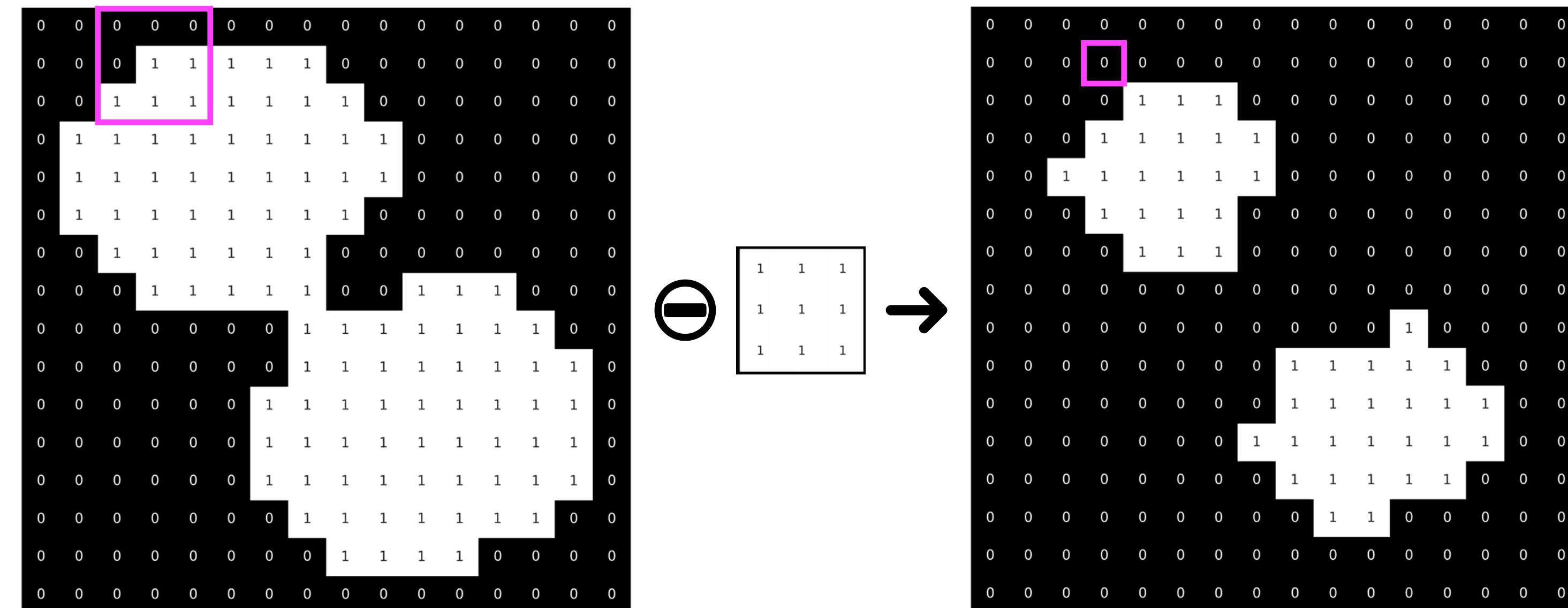
# Erosion Operation



$$\begin{array}{ccc}
 \begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{matrix} & \ominus & \begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix} \\
 \begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{matrix} & \rightarrow & 0
 \end{array}$$
  

$$\begin{array}{ccc}
 \begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{matrix} & \ominus & \begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix} \\
 \begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{matrix} & \rightarrow & 0
 \end{array}$$

# Erosion Operation

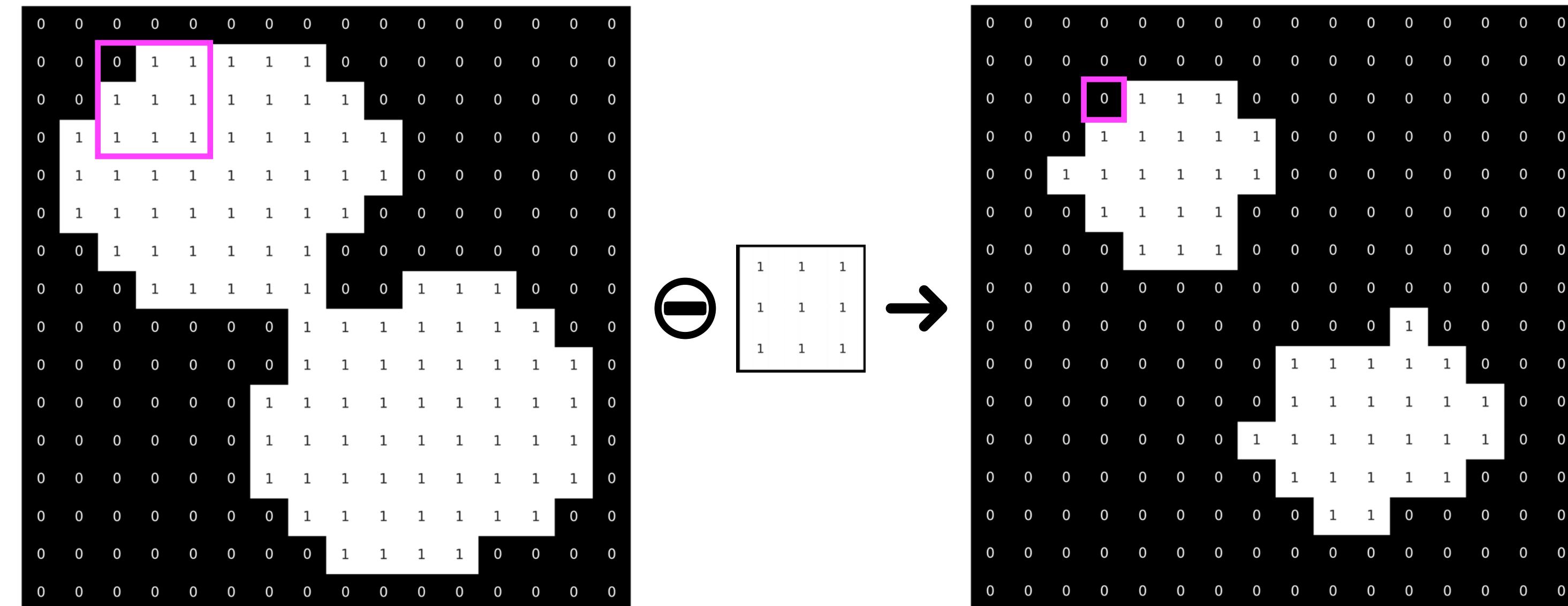


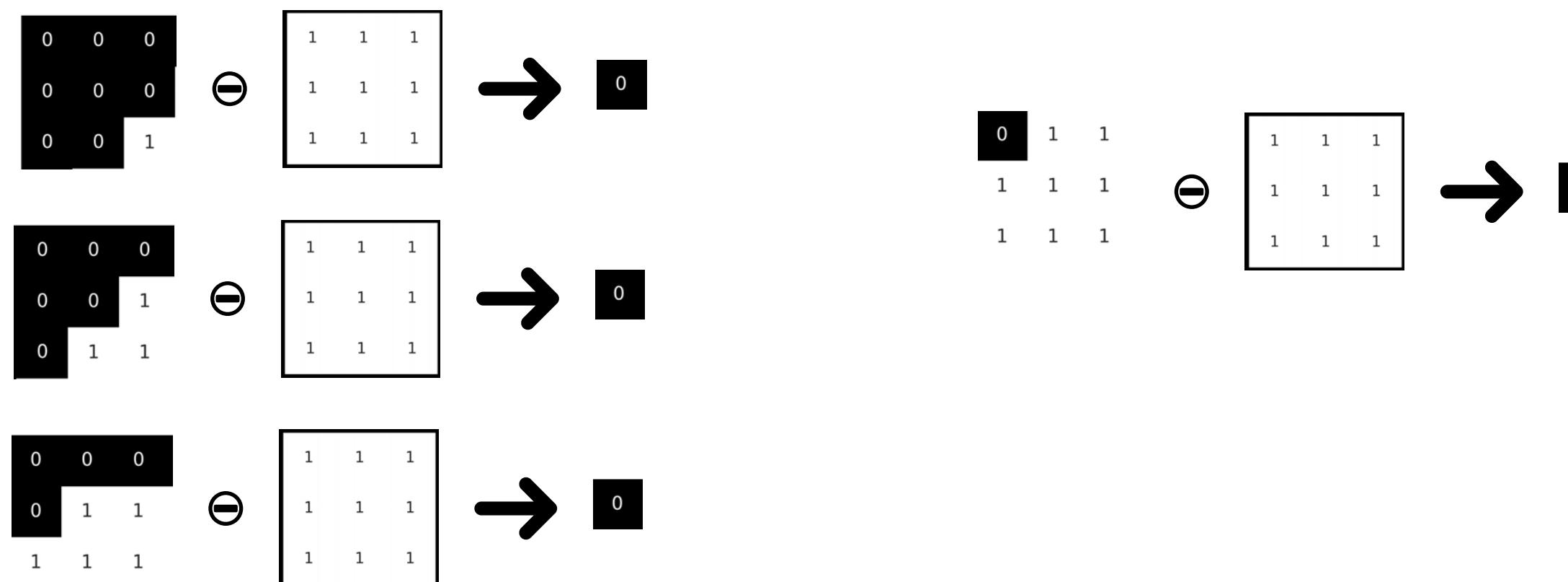
$$\begin{array}{ccc}
 \begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{matrix} & \ominus & \begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix} \\
 \begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{matrix} & \rightarrow & 0
 \end{array}$$
  

$$\begin{array}{ccc}
 \begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{matrix} & \ominus & \begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix} \\
 \begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \end{matrix} & \rightarrow & 0
 \end{array}$$
  

$$\begin{array}{ccc}
 \begin{matrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{matrix} & \ominus & \begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix} \\
 \begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{matrix} & \rightarrow & 0
 \end{array}$$

# Erosion Operation

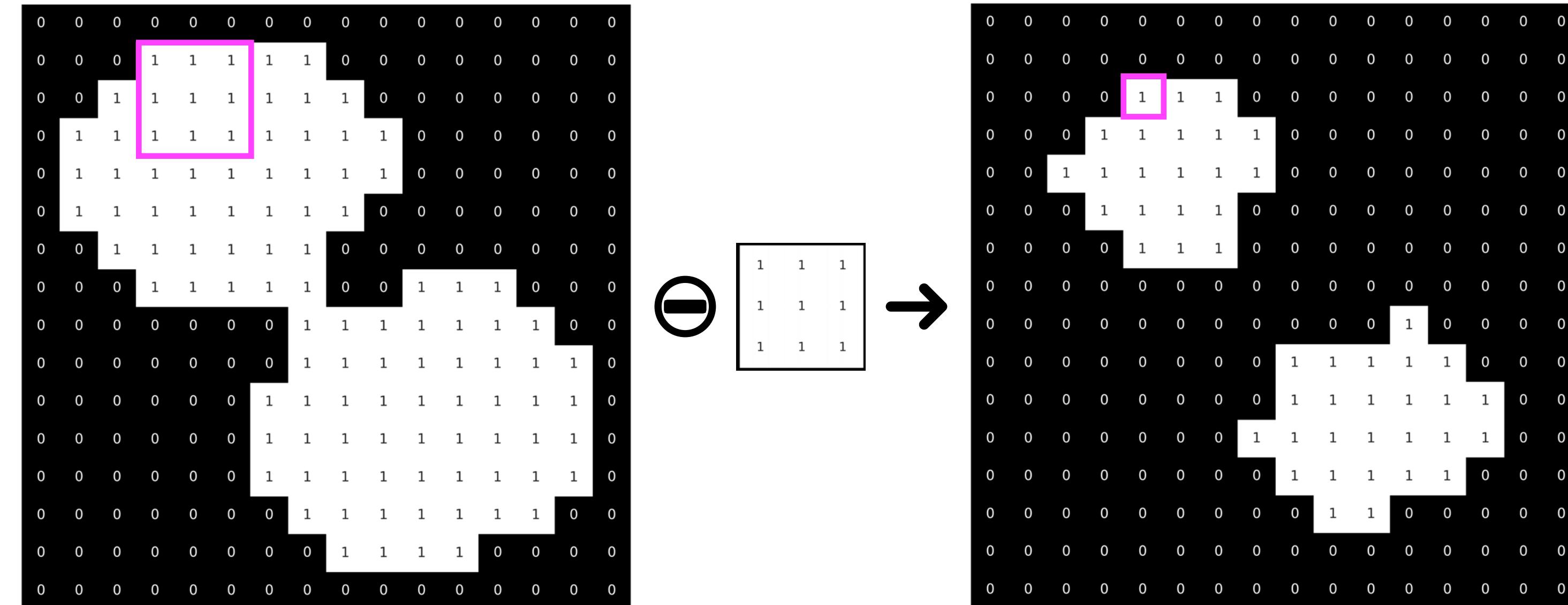




Three examples of erosion operations are shown, each consisting of an input image, an erosion operator symbol ( $\ominus$ ), a 3x3 kernel, and the resulting output image.

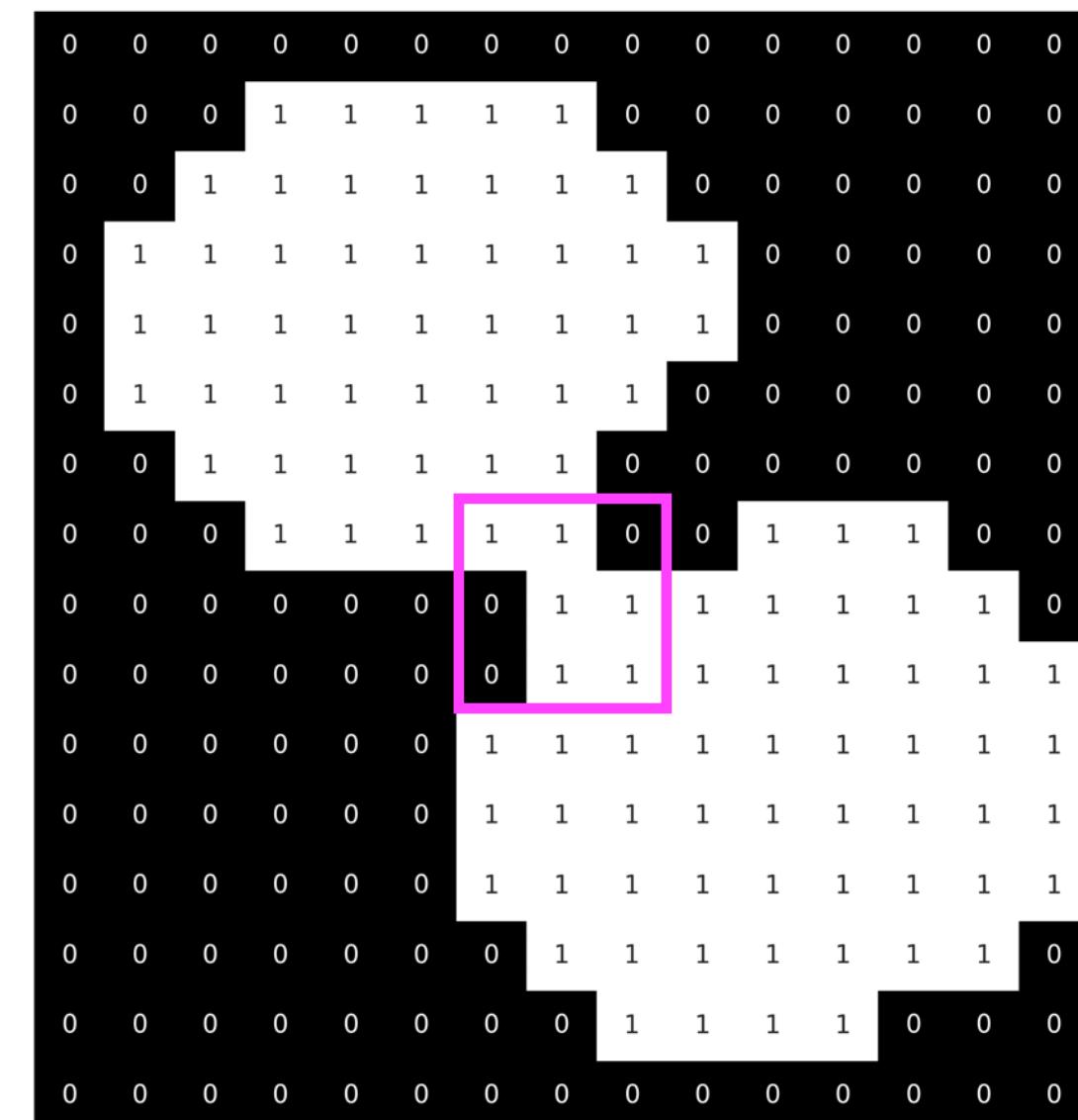
- Example 1:** Input image has a central white 3x3 block. Erosion with a 3x3 kernel results in a smaller white 3x3 block at the center of the output.
- Example 2:** Input image has a white vertical strip. Erosion with a 3x3 kernel results in a shorter white vertical strip in the center of the output.
- Example 3:** Input image has a white horizontal strip. Erosion with a 3x3 kernel results in a shorter white horizontal strip in the center of the output.

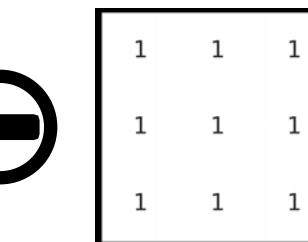
# Erosion Operation

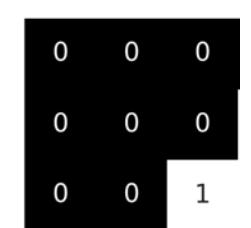
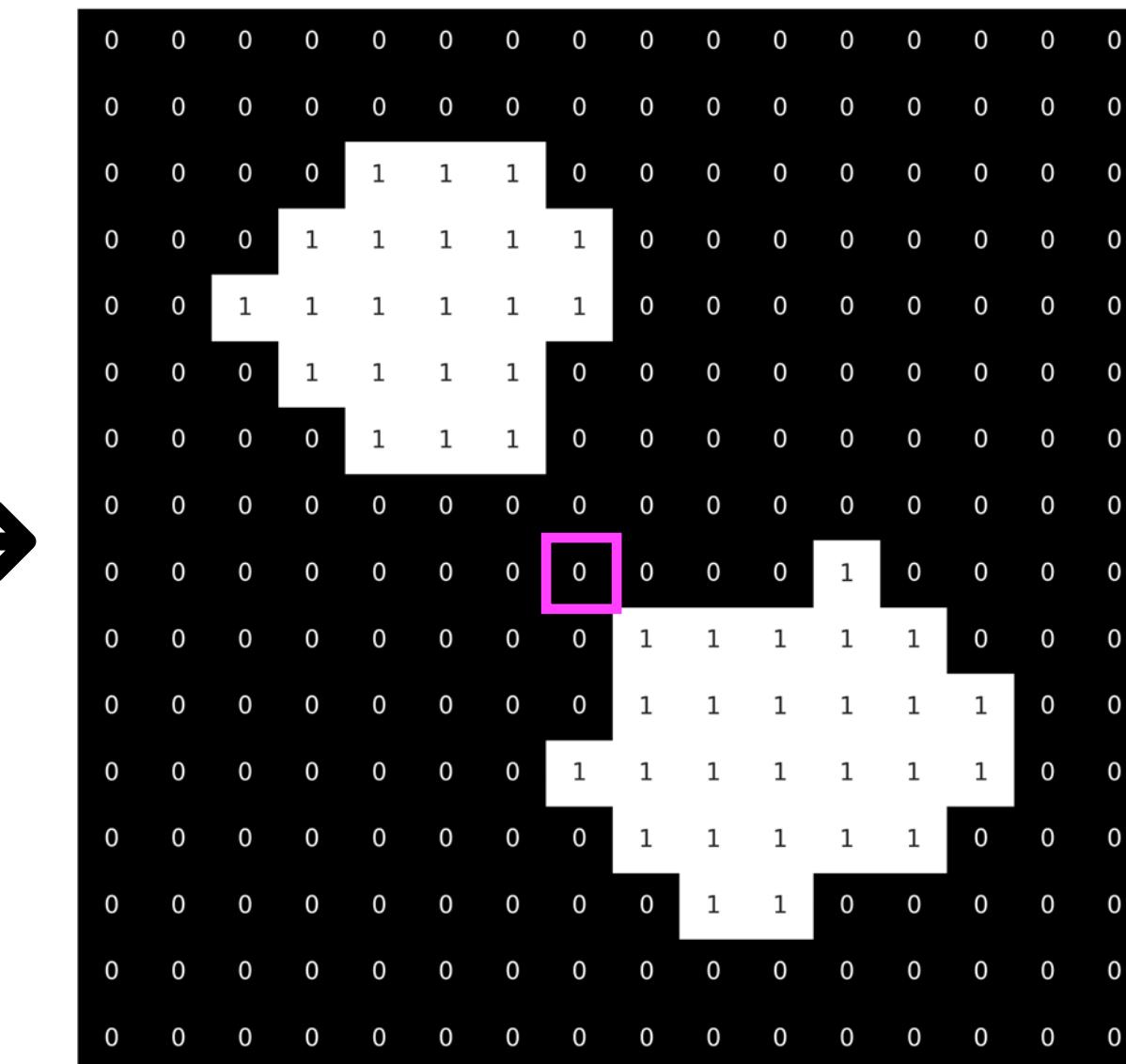


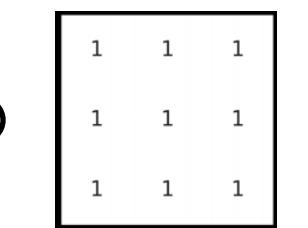
$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\ominus$	$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	$\rightarrow$	$0$
$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$	$\ominus$	$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	$\rightarrow$	$0$
$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	$\ominus$	$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	$\rightarrow$	$0$
$\begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	$\ominus$	$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	$\rightarrow$	$0$
$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	$\ominus$	$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	$\rightarrow$	$1$

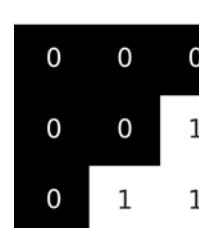
# Erosion Operation

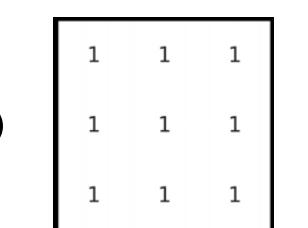


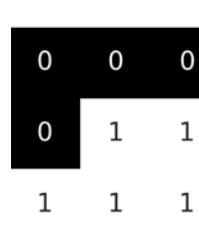
$$\ominus$$


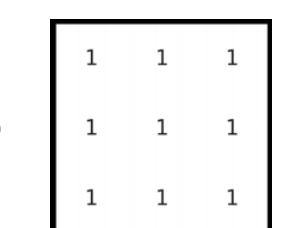


$$\ominus$$


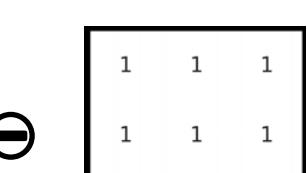


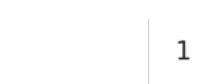
$$\ominus$$


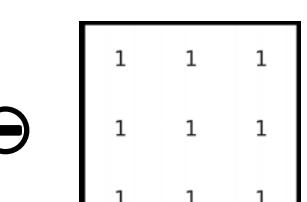


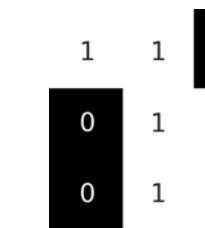
$$\ominus$$


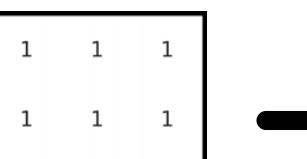


$$\ominus$$




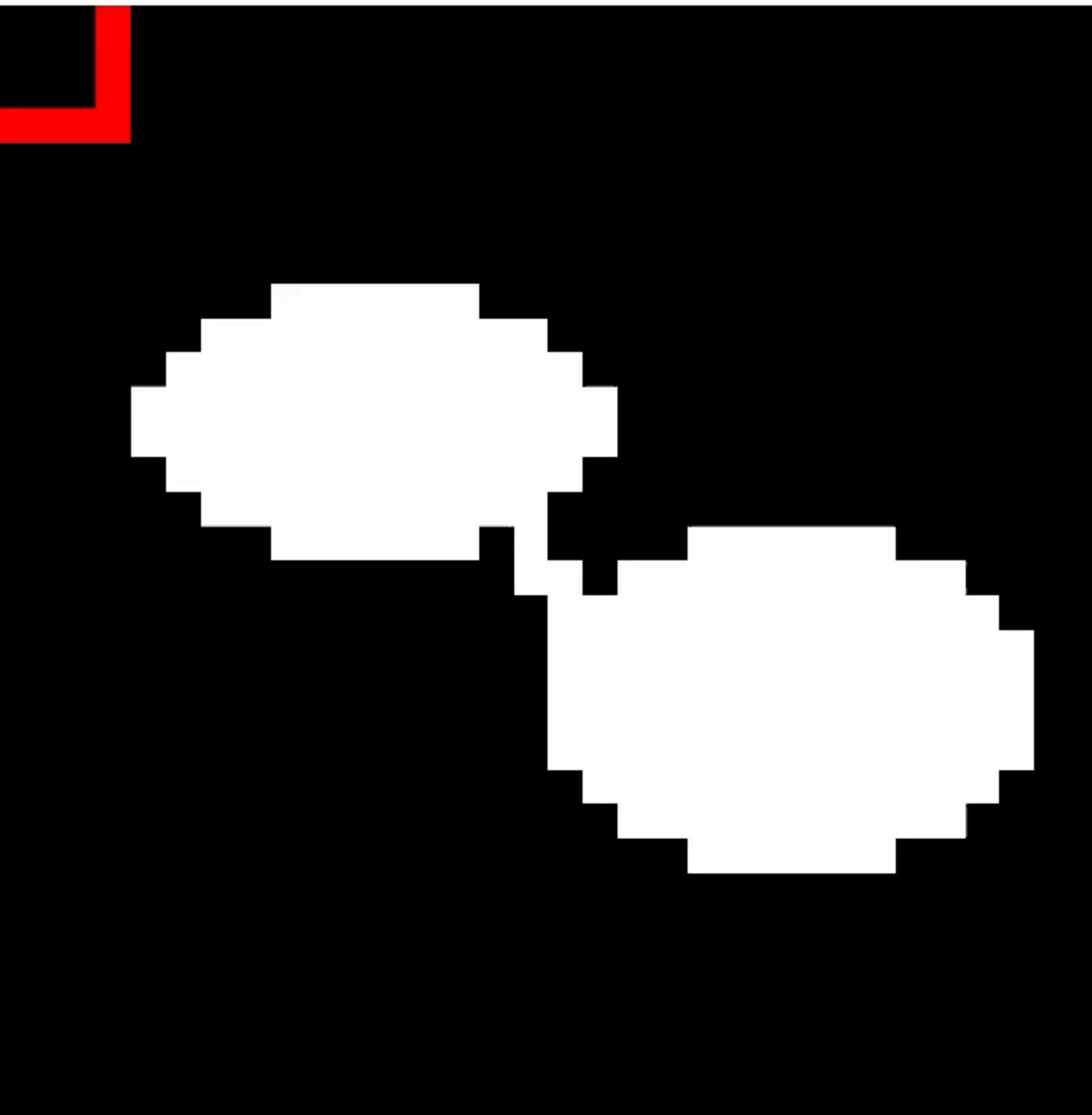
$$\ominus$$




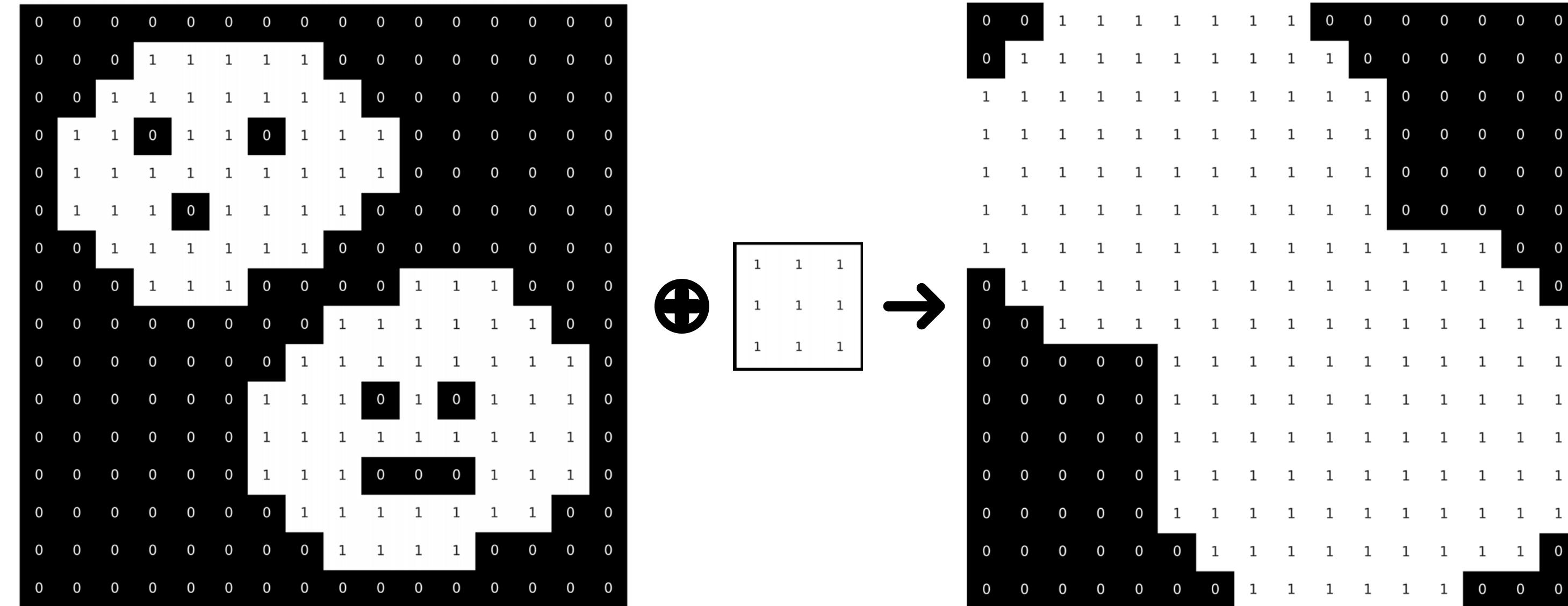
$$\ominus$$




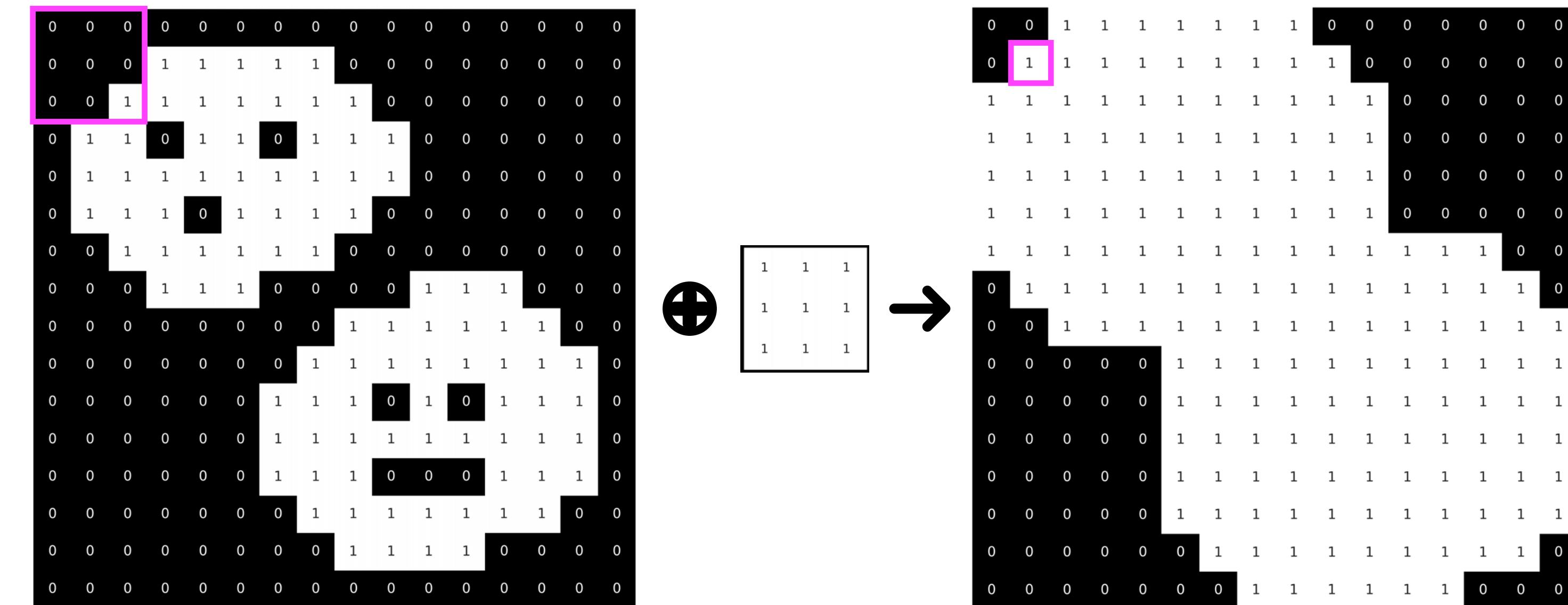
# Erosion - The Movie



# Dilation Operation

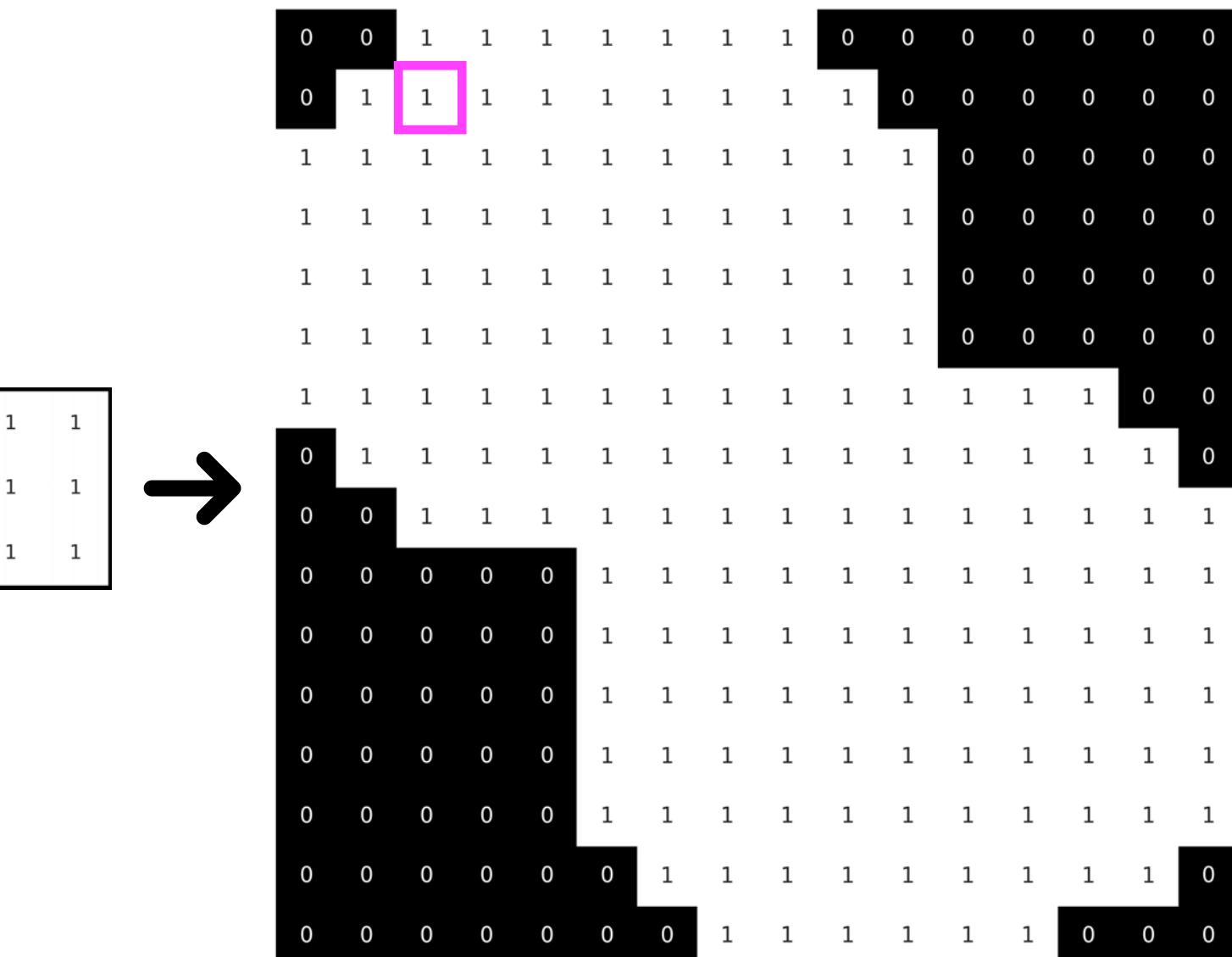
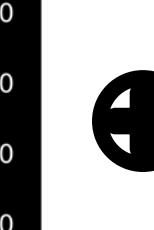
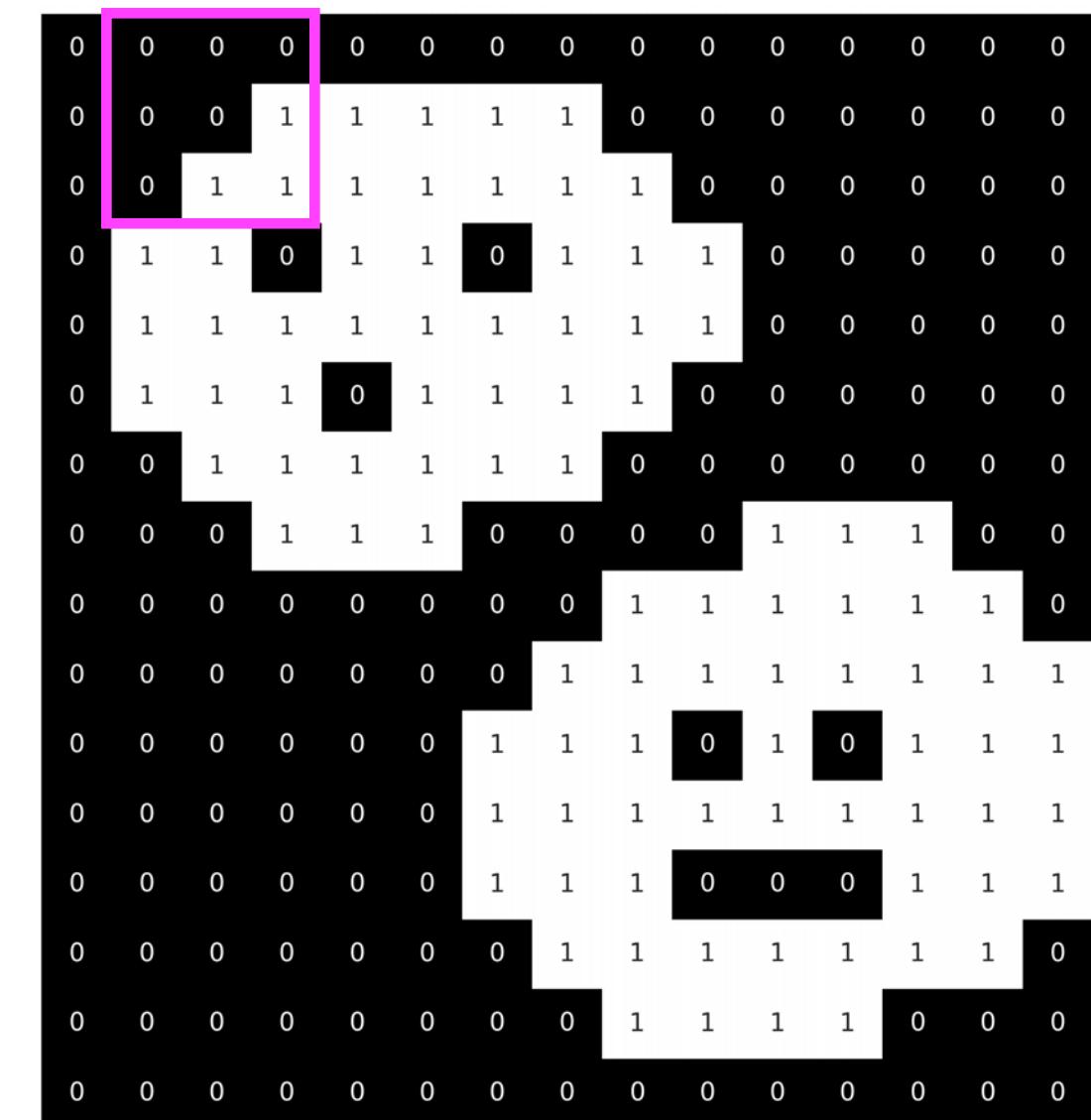


# Dilation Operation



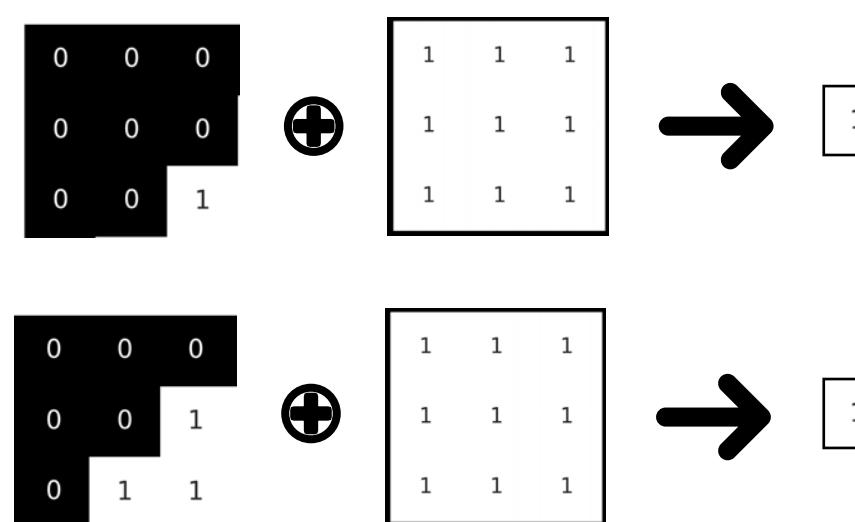
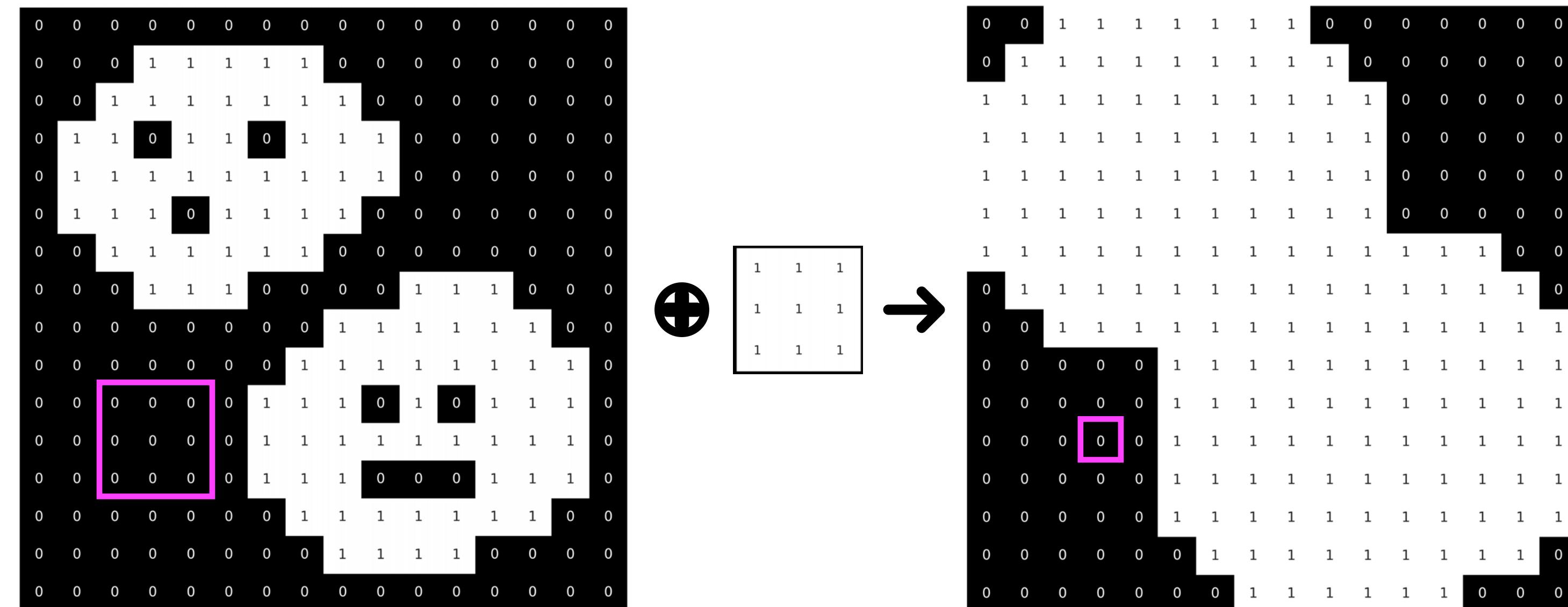
$$\begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{matrix} + \begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix} \rightarrow \begin{matrix} \dots \end{matrix}$$

# Dilation Operation

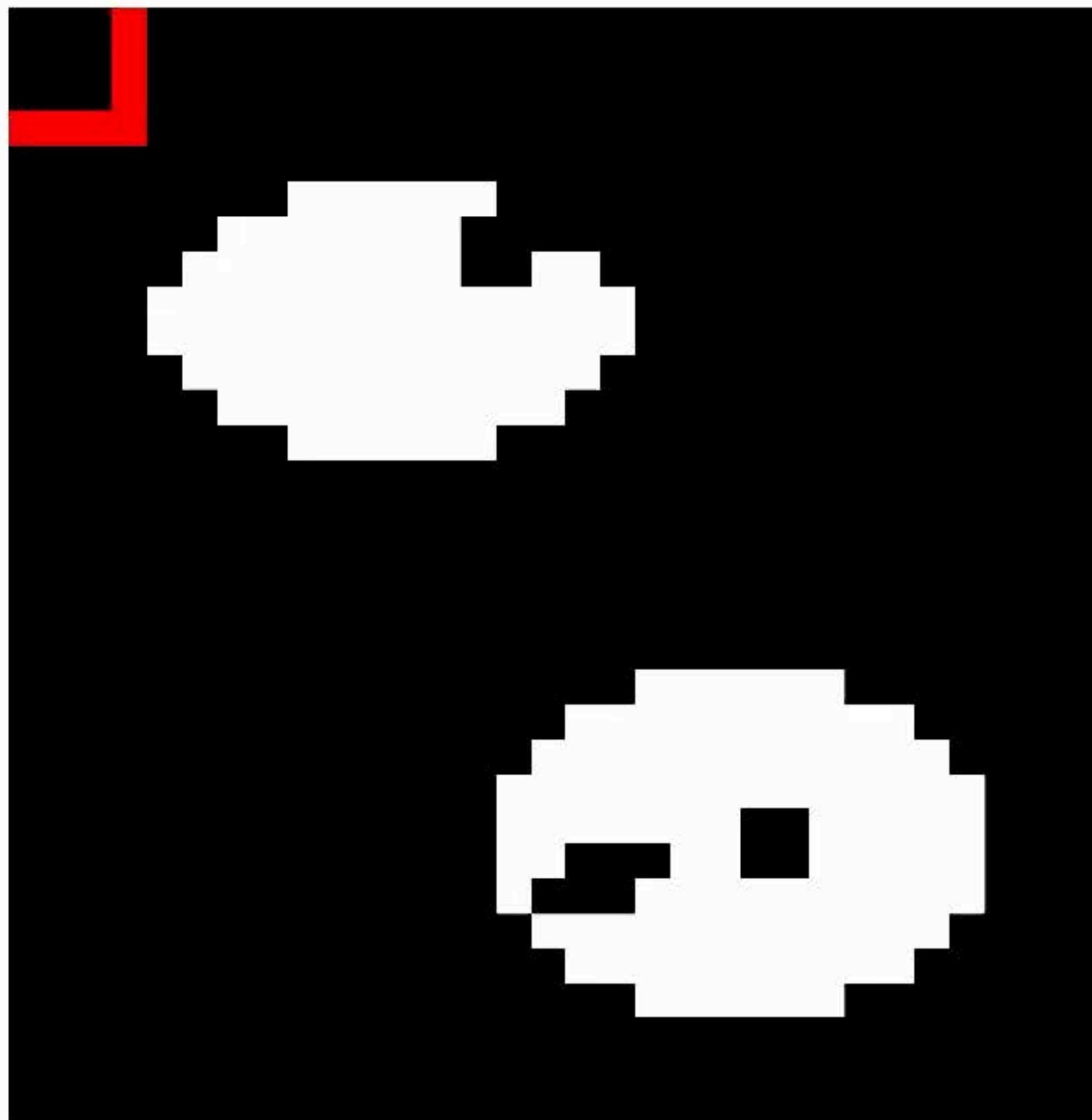


$$\begin{matrix}
 0 & 0 & 0 \\
 0 & 0 & 0 \\
 0 & 0 & 1
 \end{matrix} \oplus \begin{matrix}
 1 & 1 & 1 \\
 1 & 1 & 1 \\
 1 & 1 & 1
 \end{matrix} \rightarrow \boxed{1}$$

# Dilation Operation



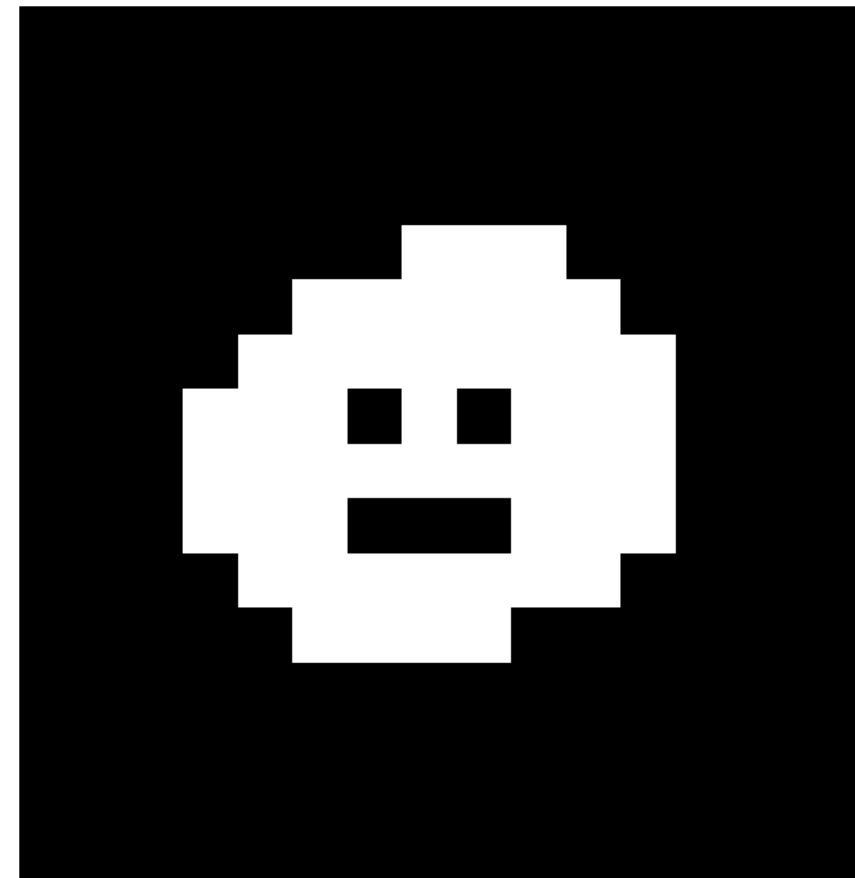
# Dilation - The Movie



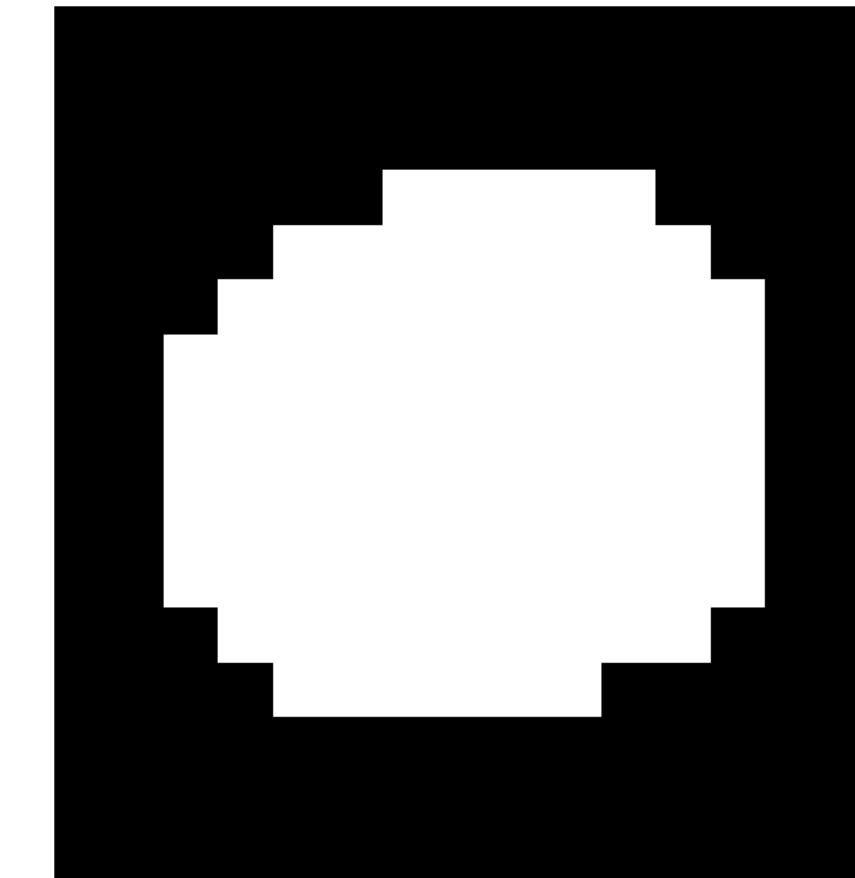
But wait,  
there is a problem.

# Together We Stand

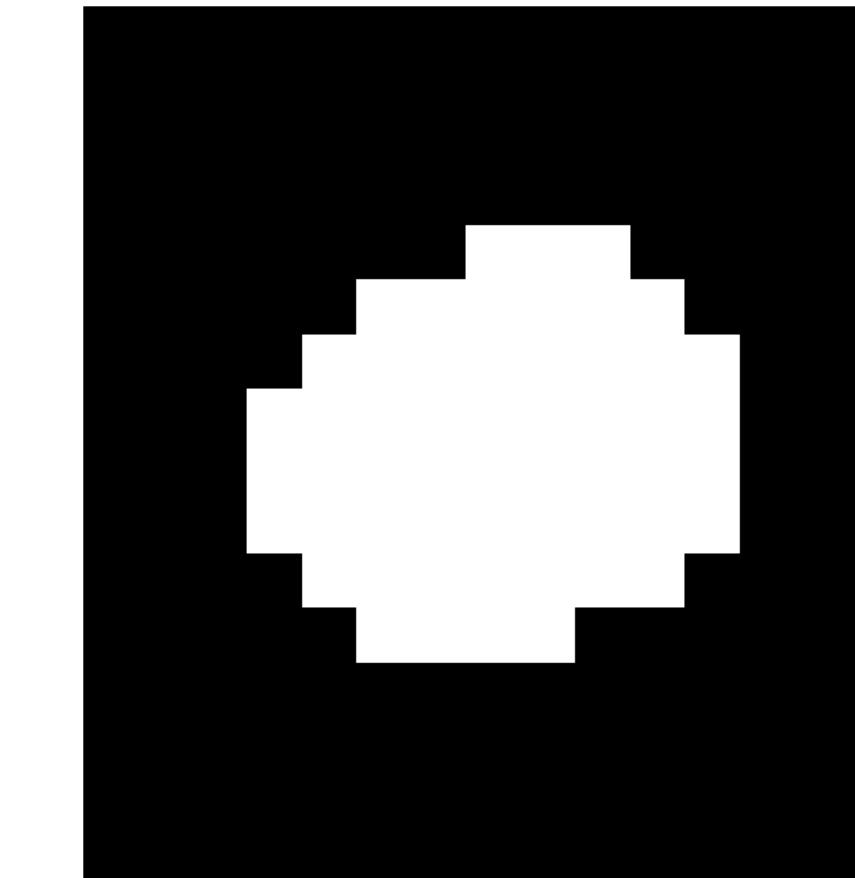
Main image



Dilated image



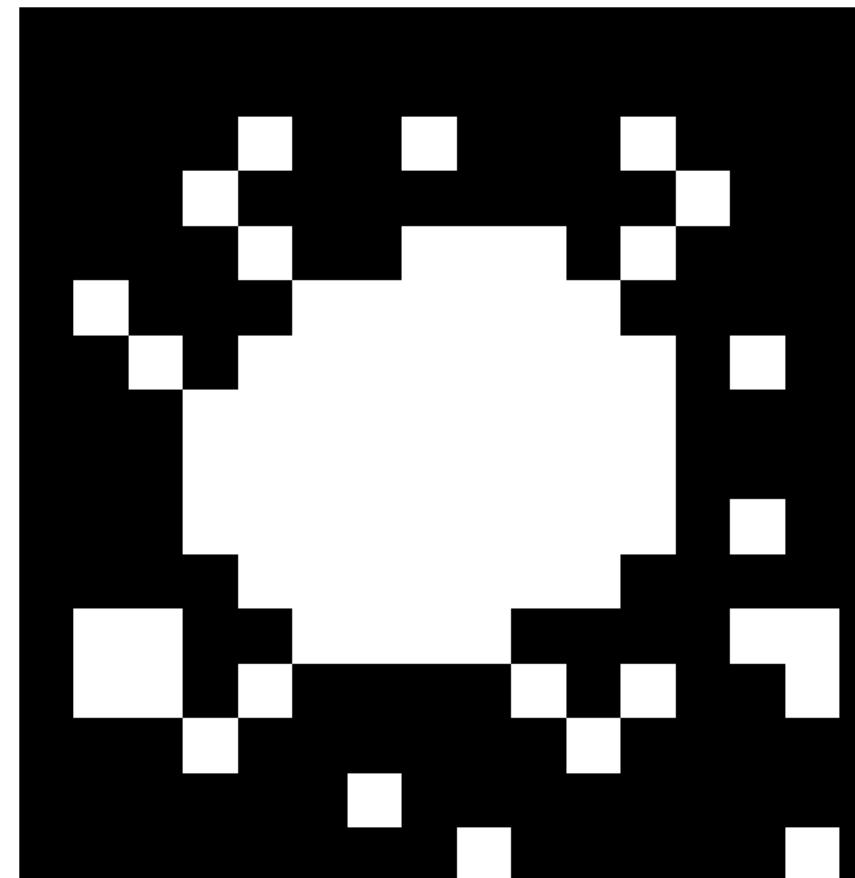
Closed image



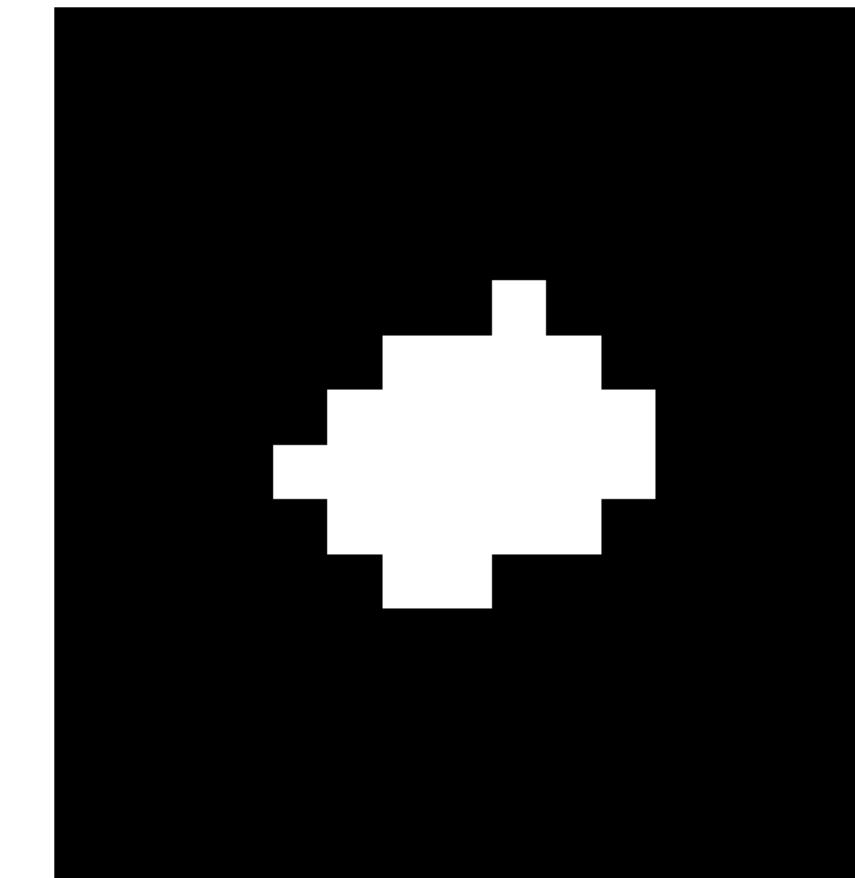
**Closing:**

Erosion of the dilation.

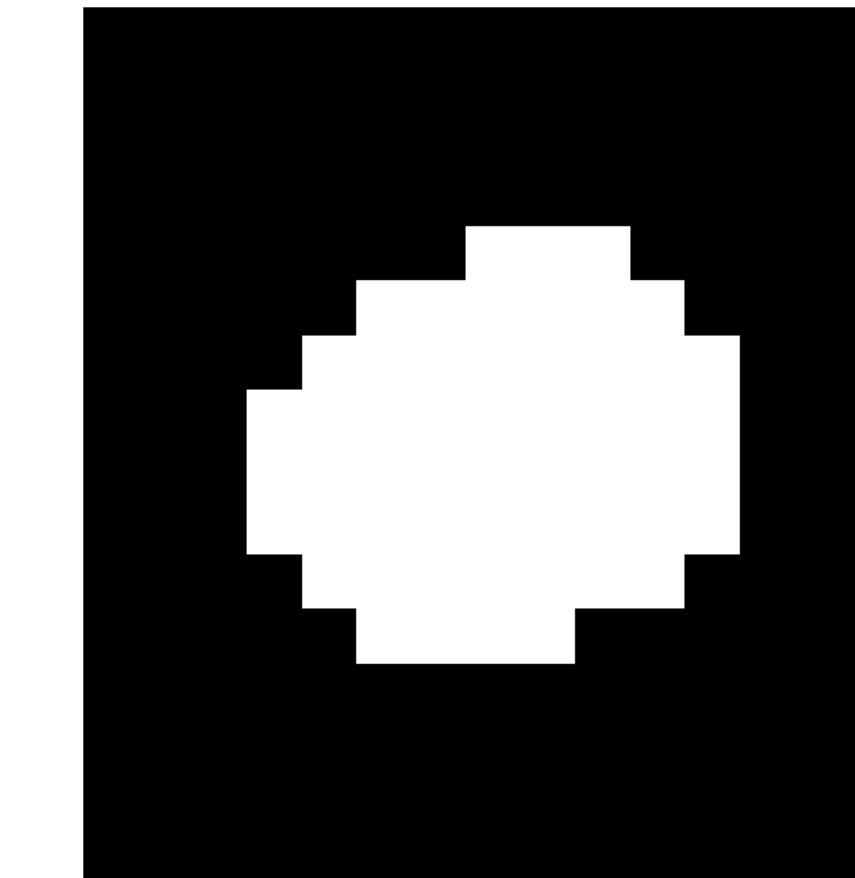
Main image



Eroded image



Opened image

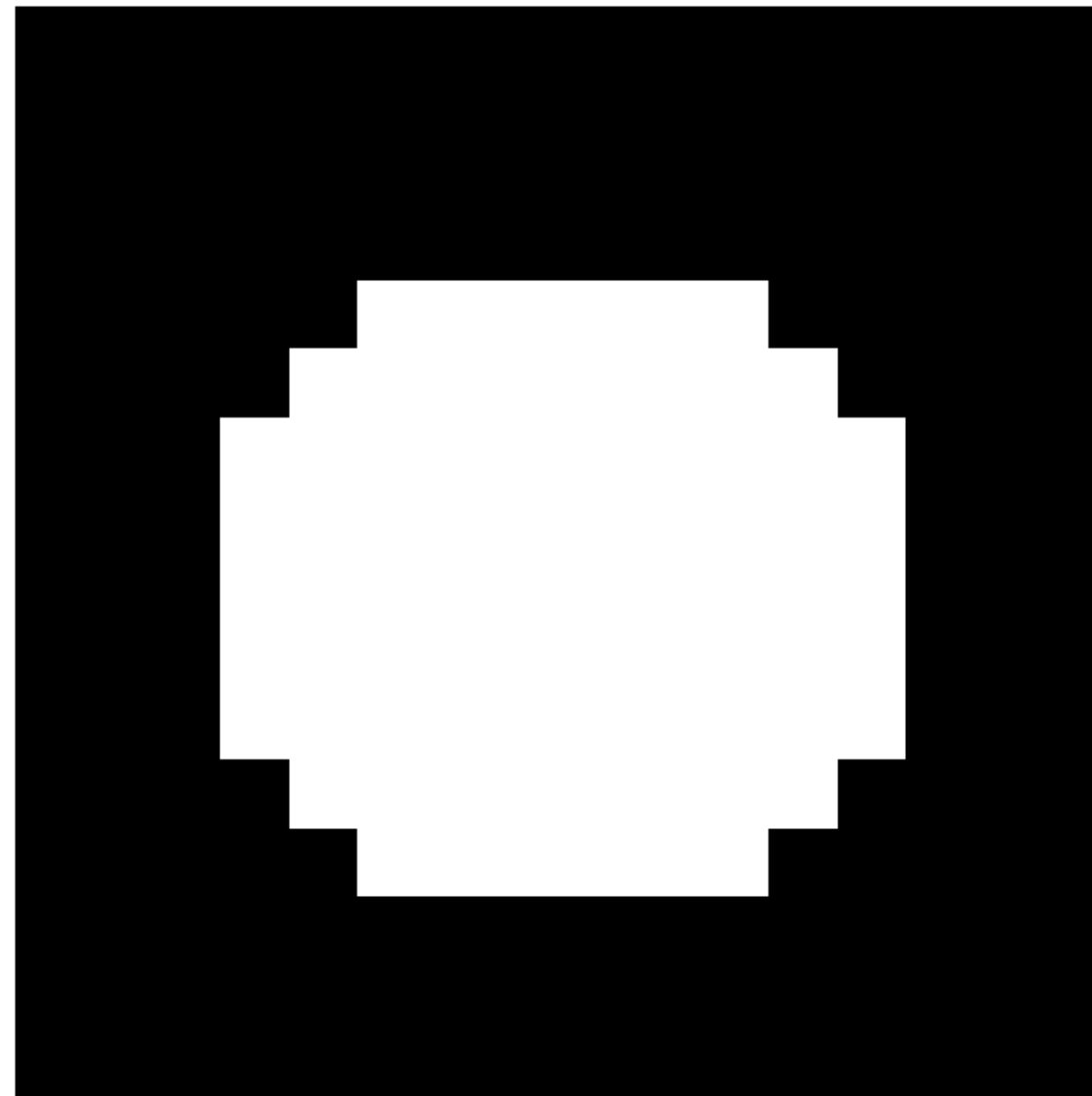


**Opening:**

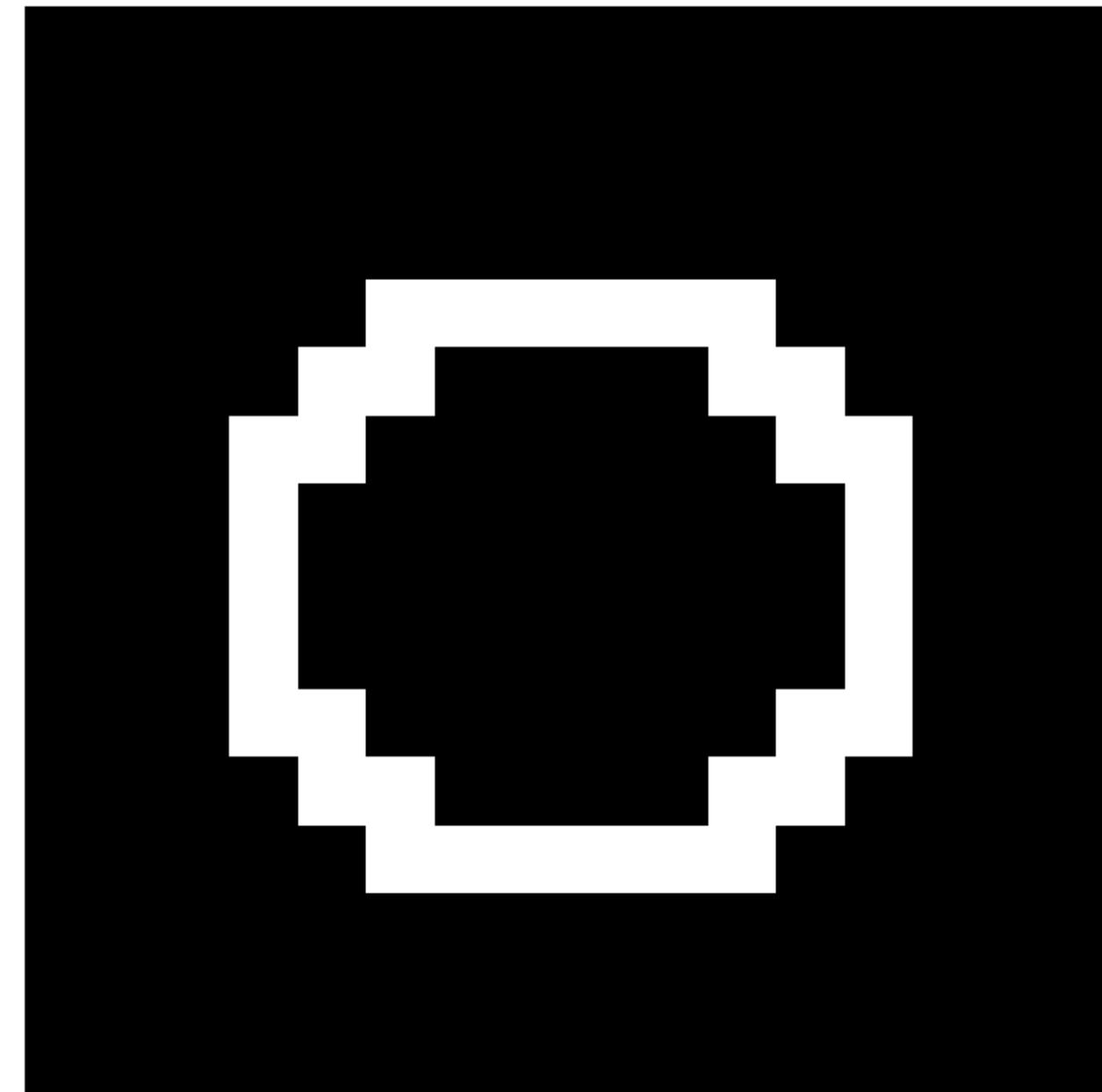
Dilation of the erosion.

# Bonus Example

Main image

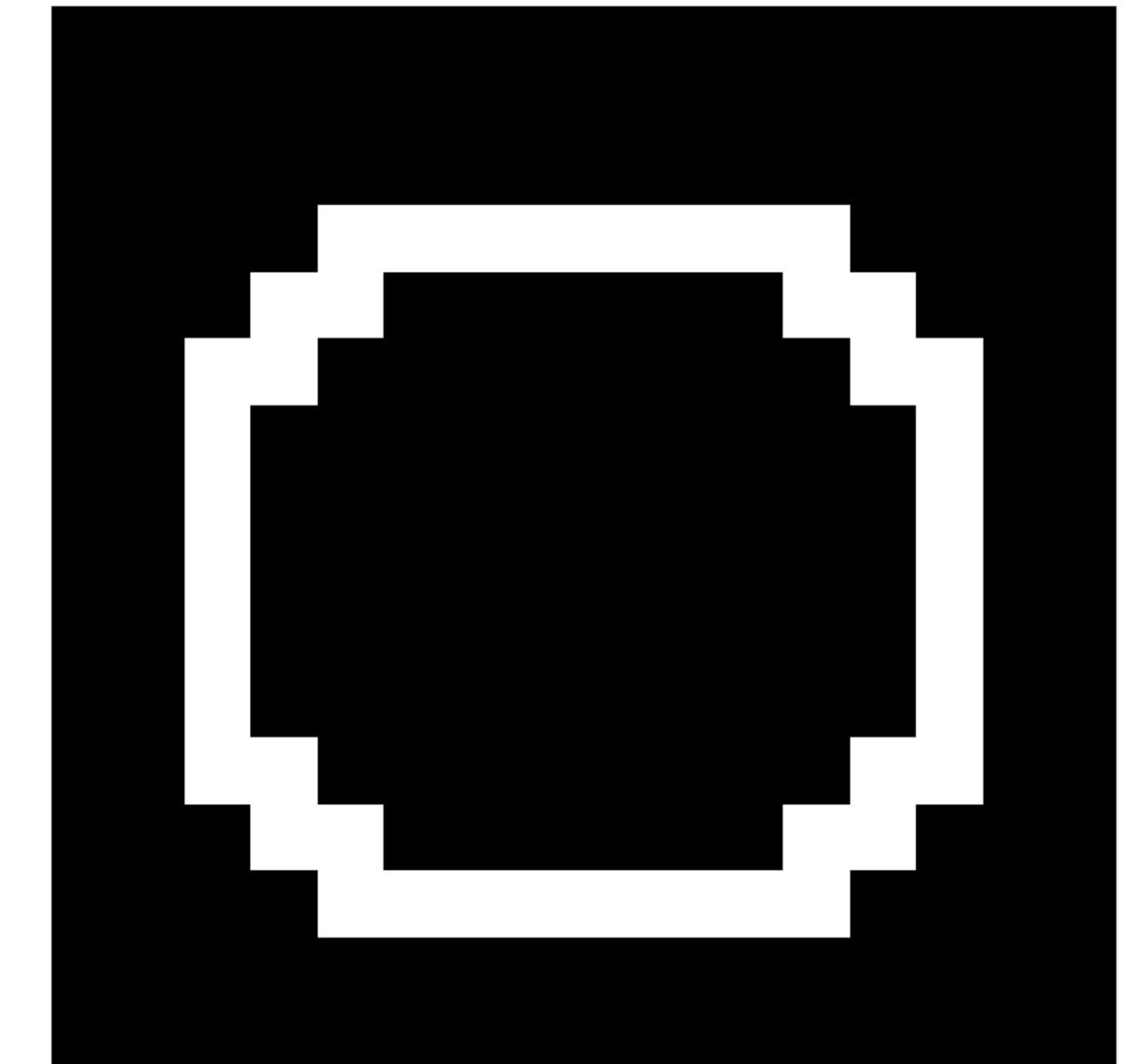


Inner edge



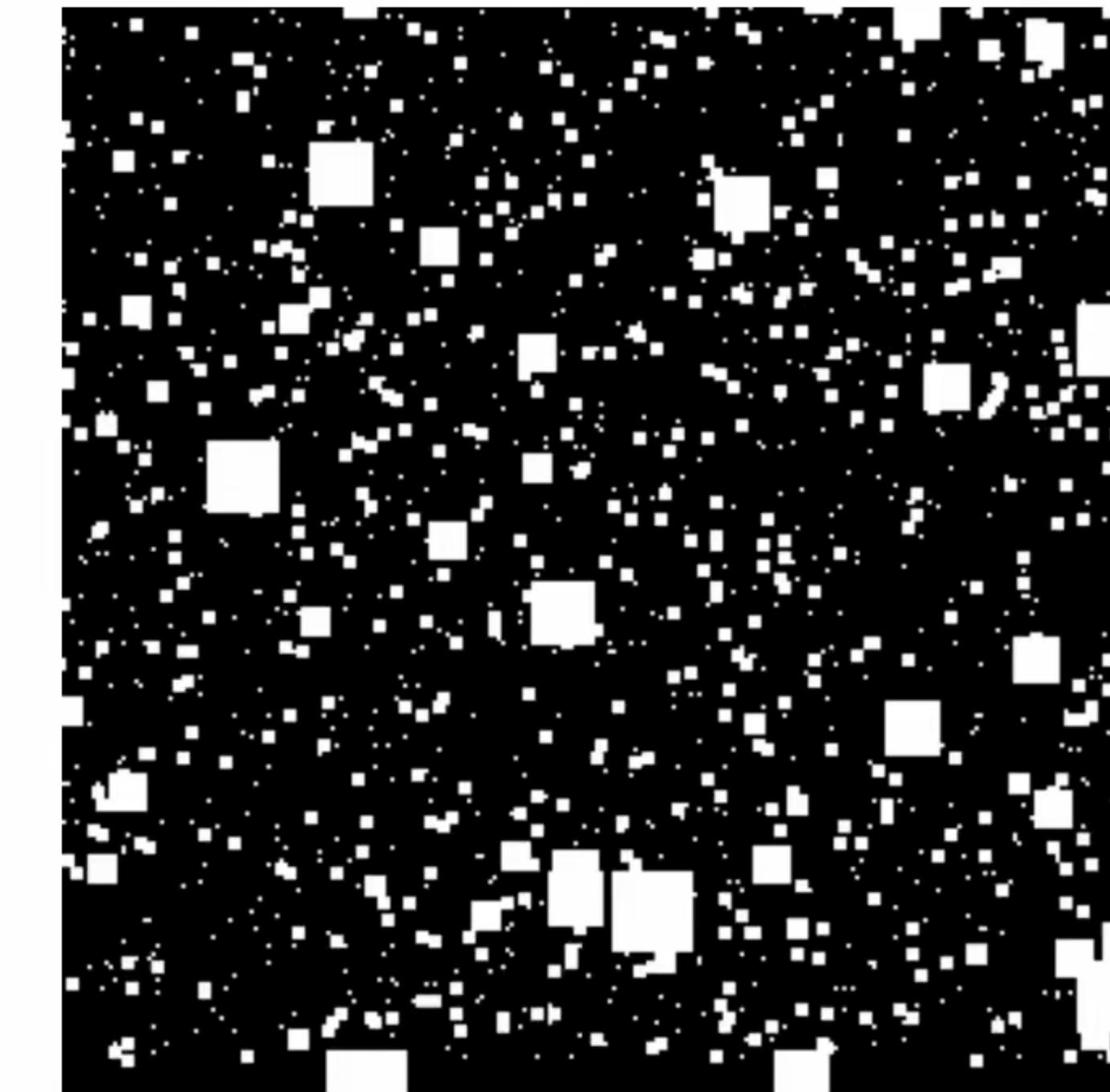
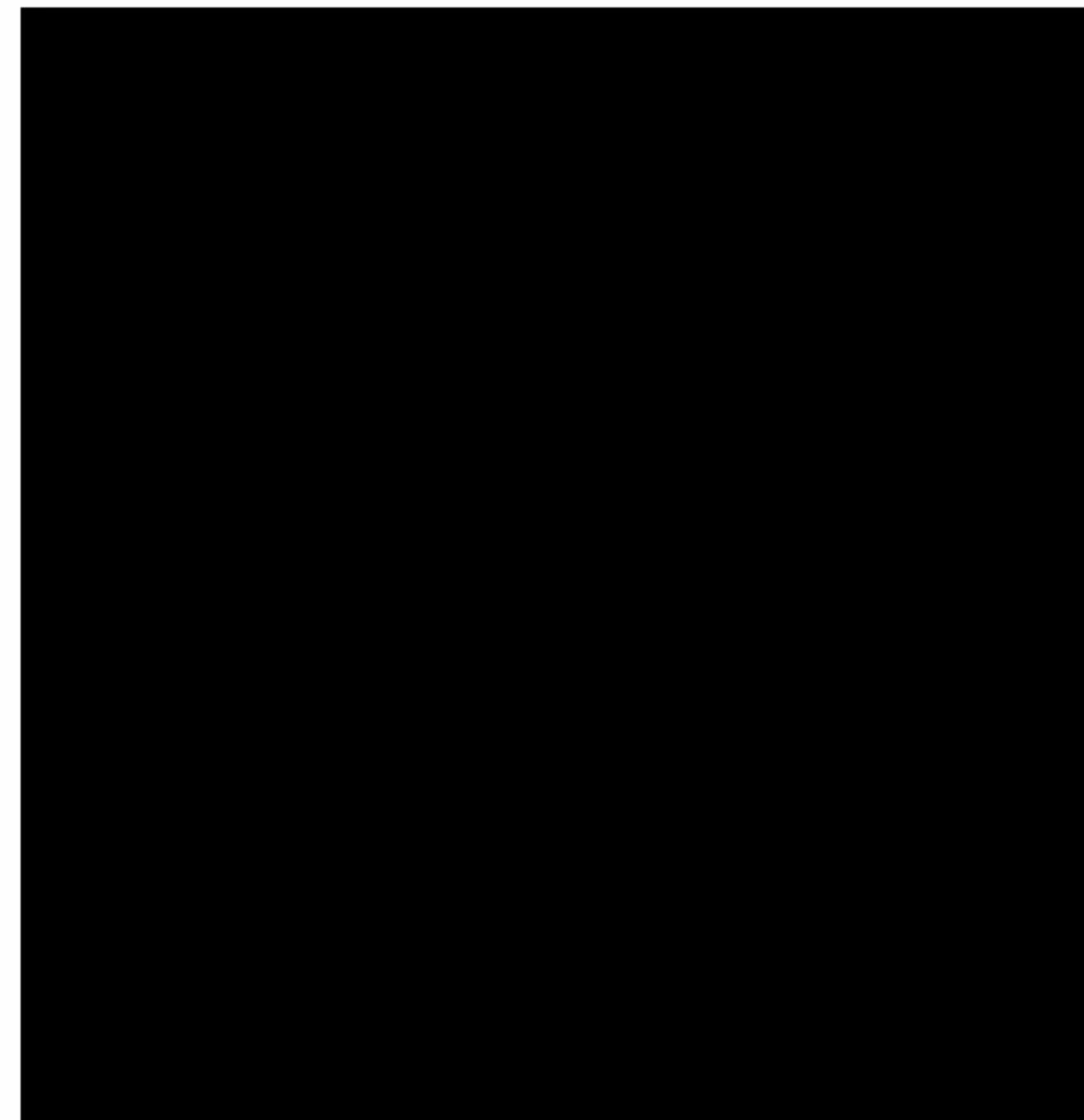
Erode more, dilate less, then subtract

Outer edge

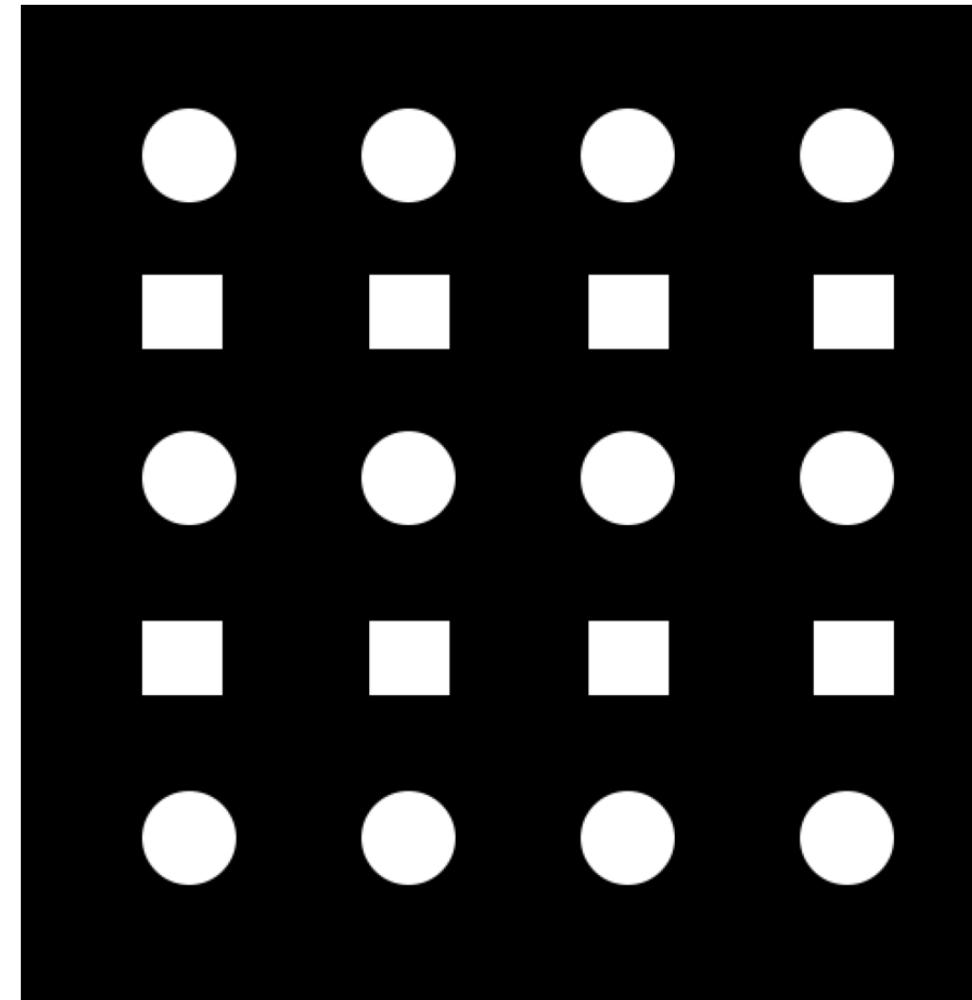


Dilate more, erode less, then subtract

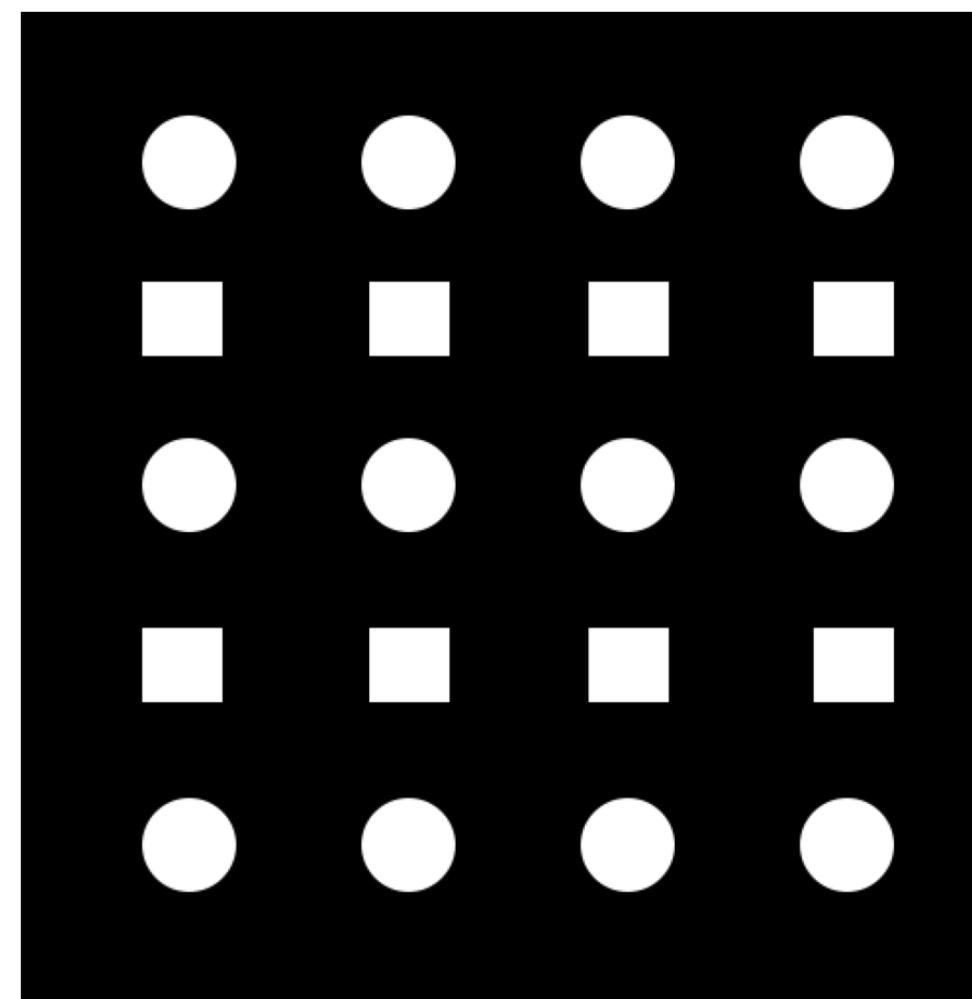
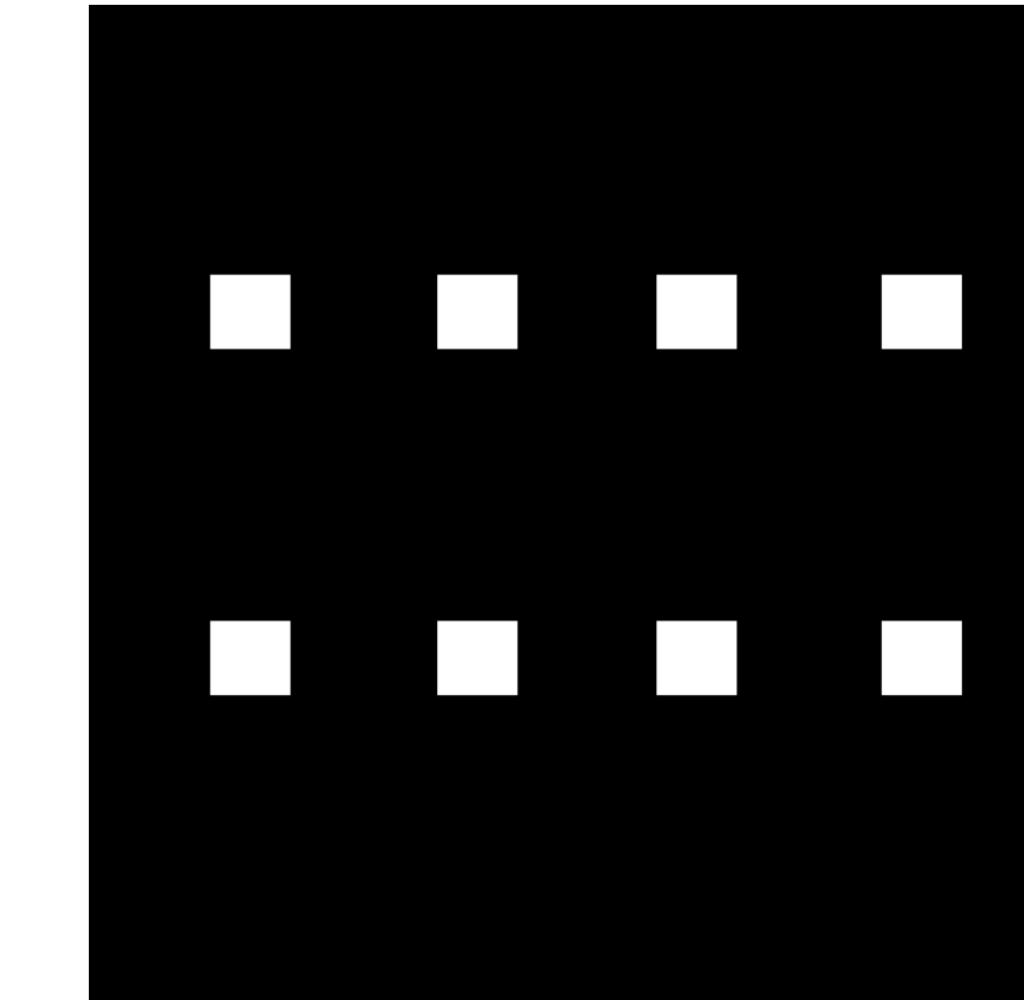
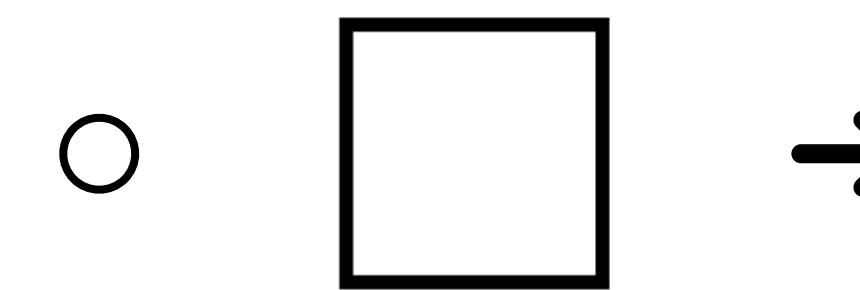
# Structuring Element Size



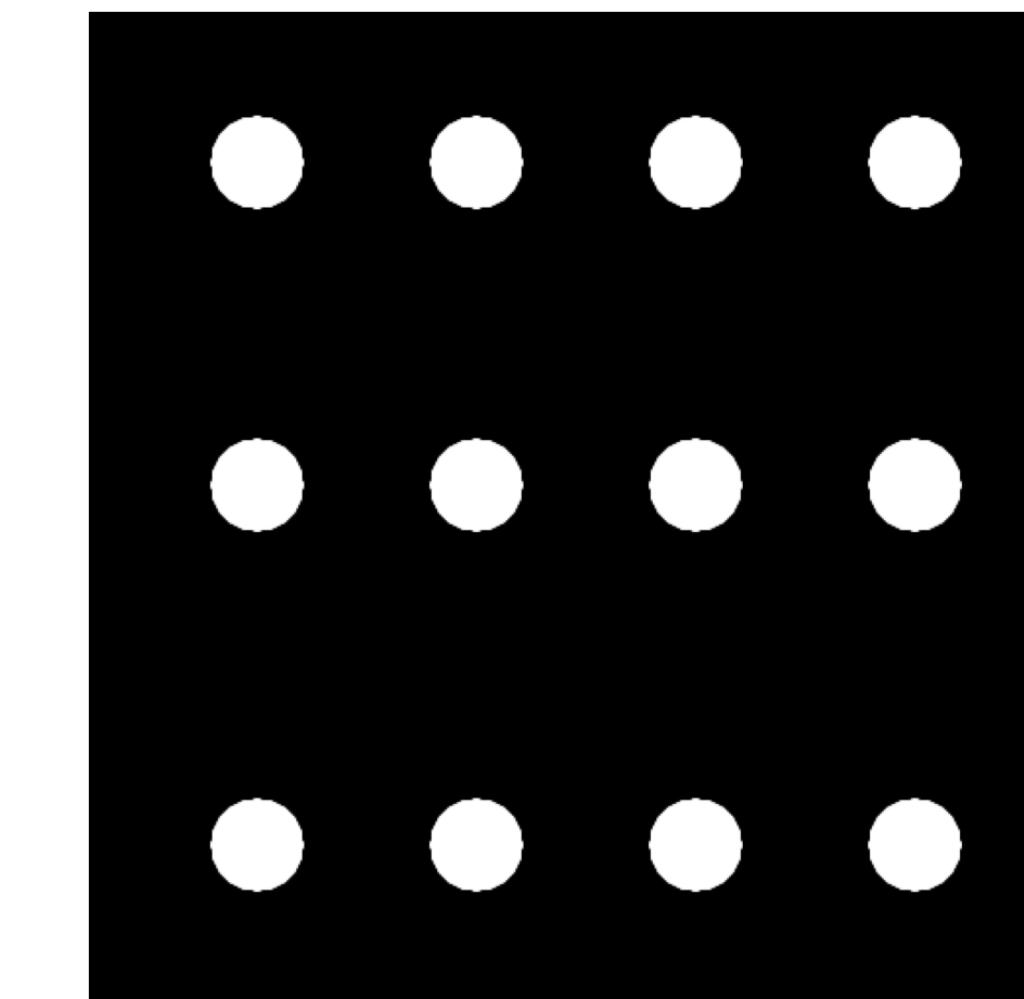
# Structuring Element Shape



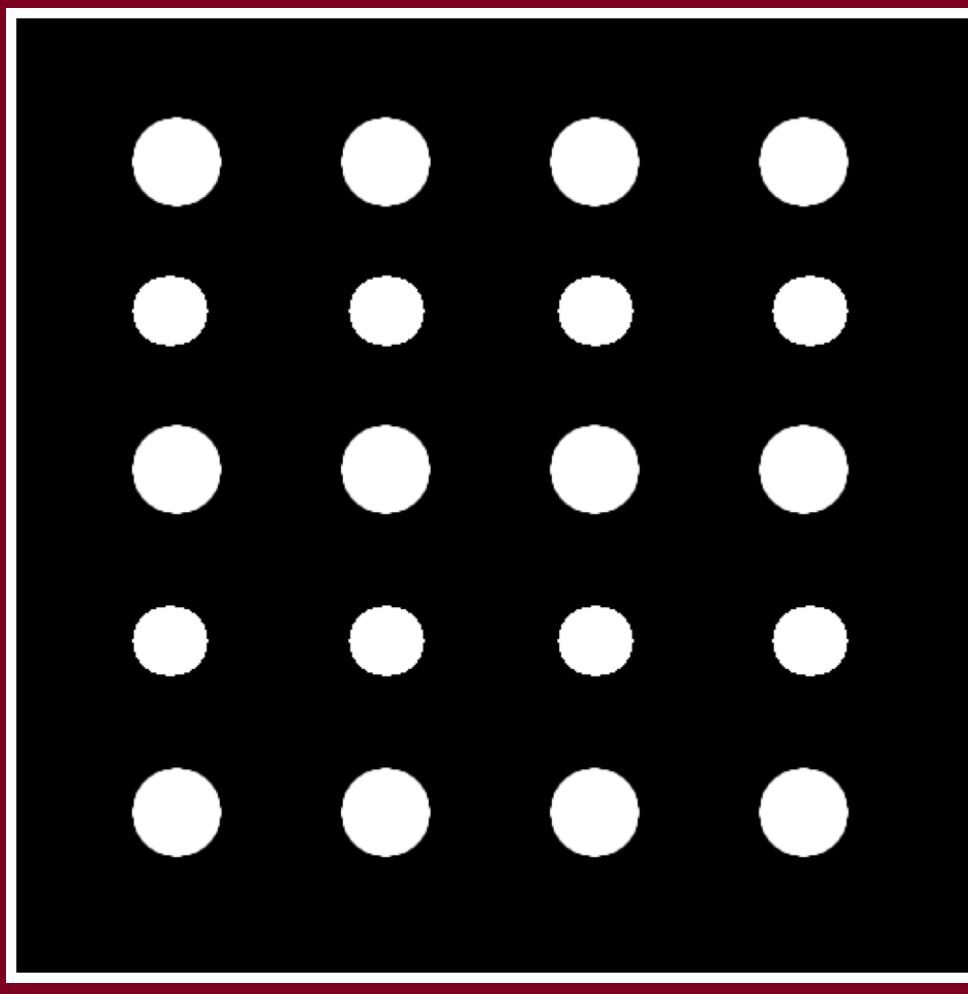
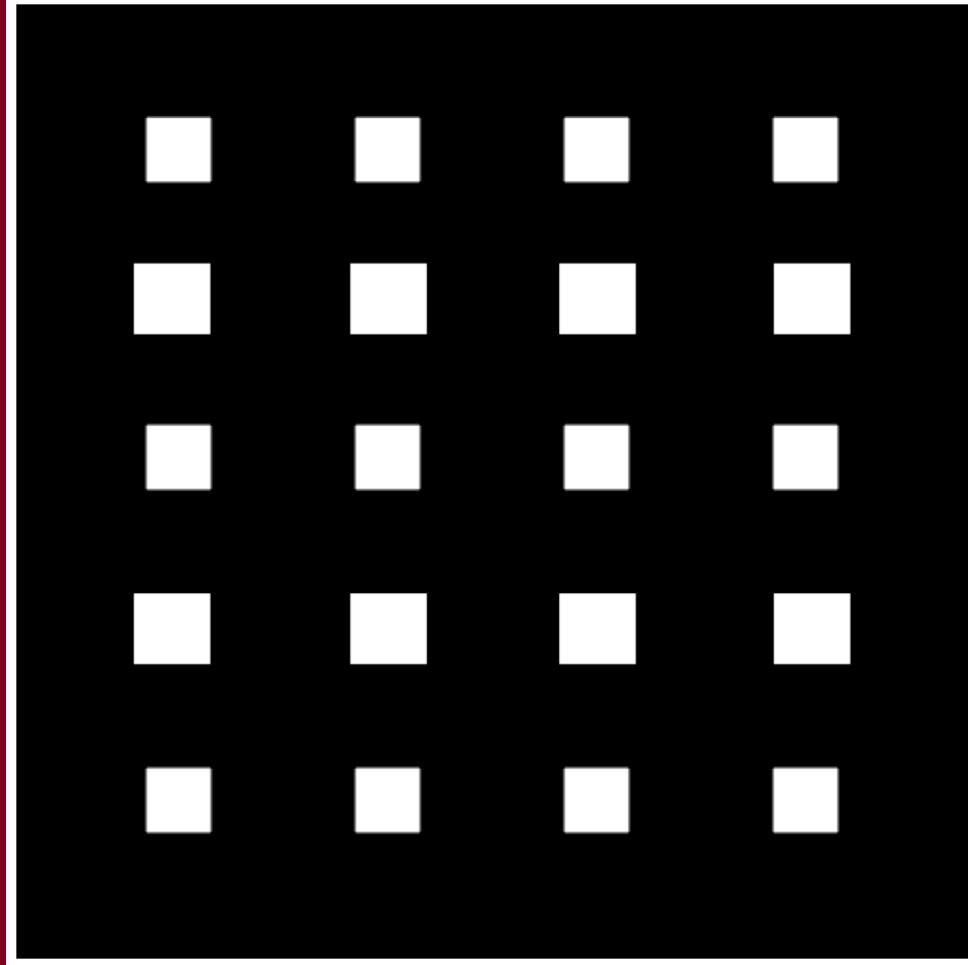
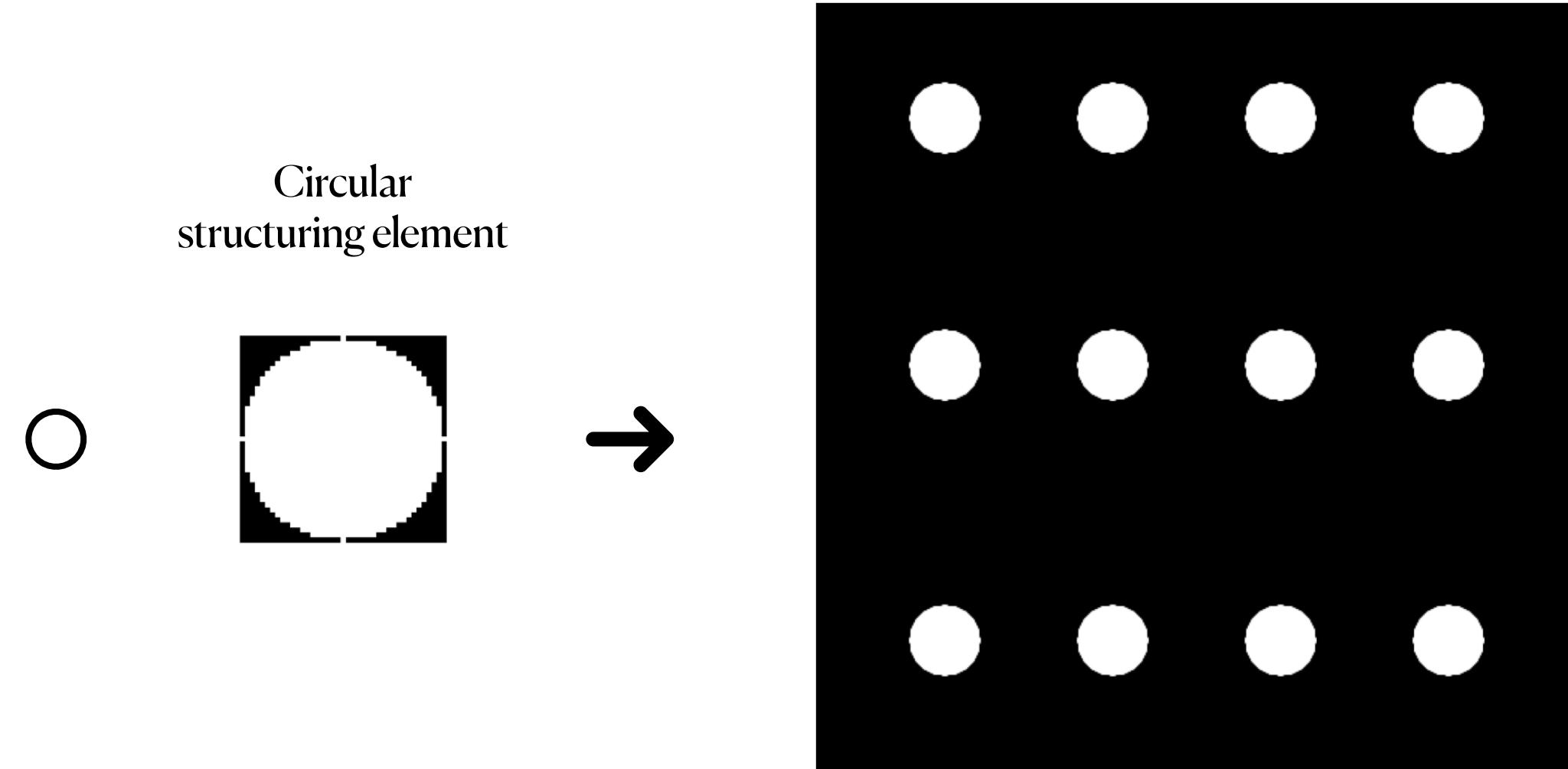
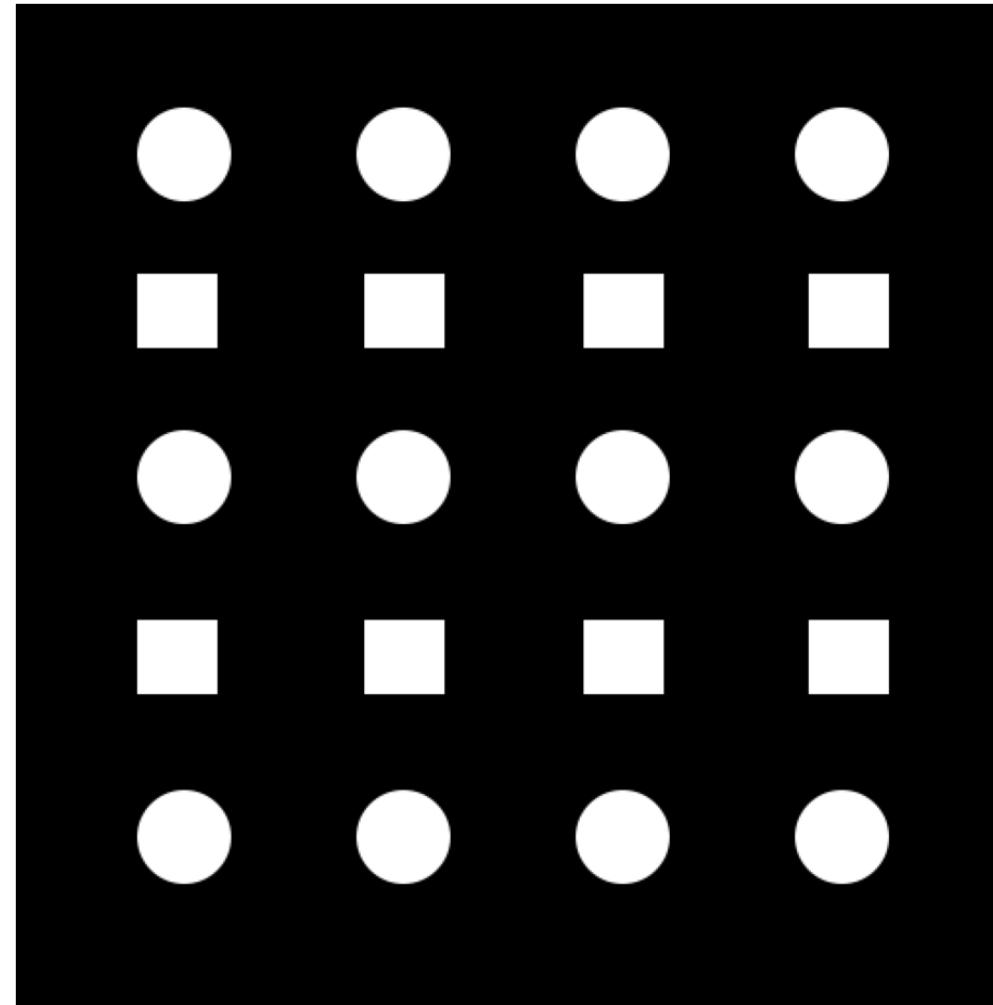
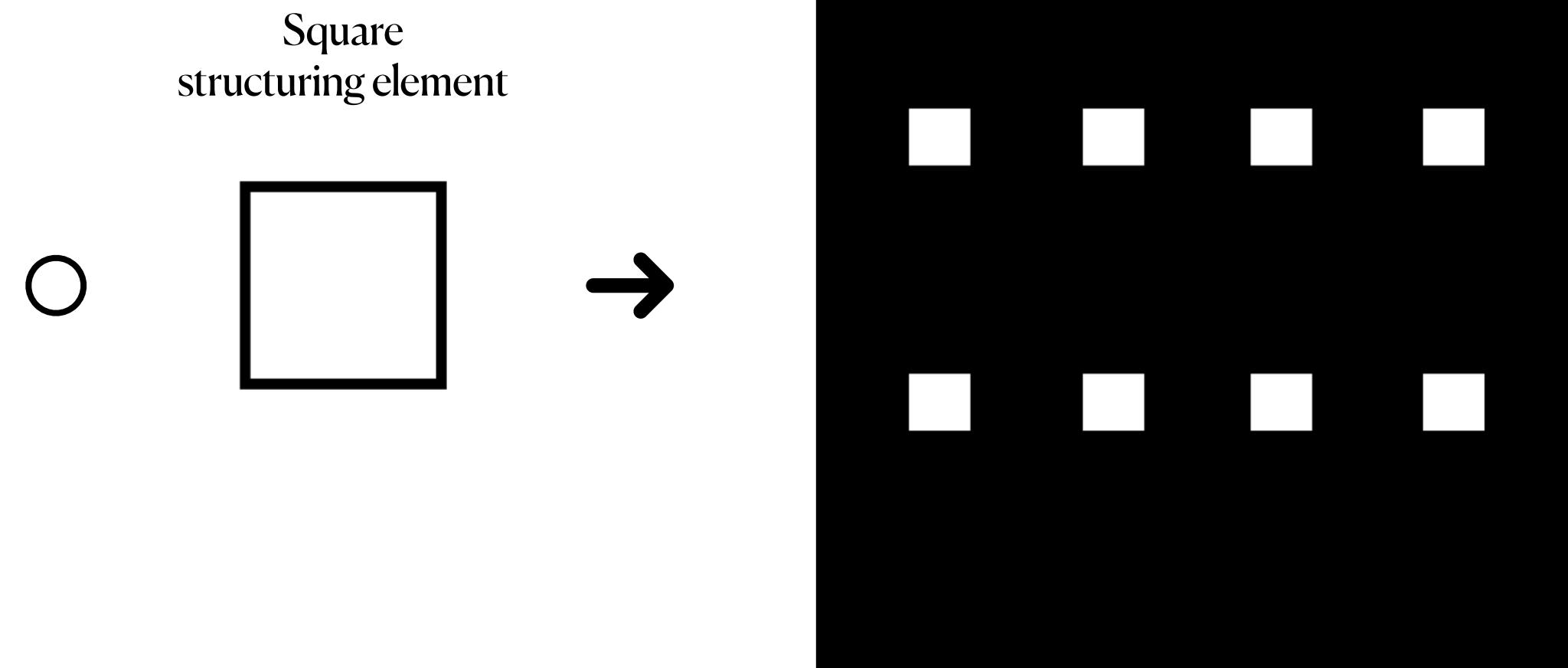
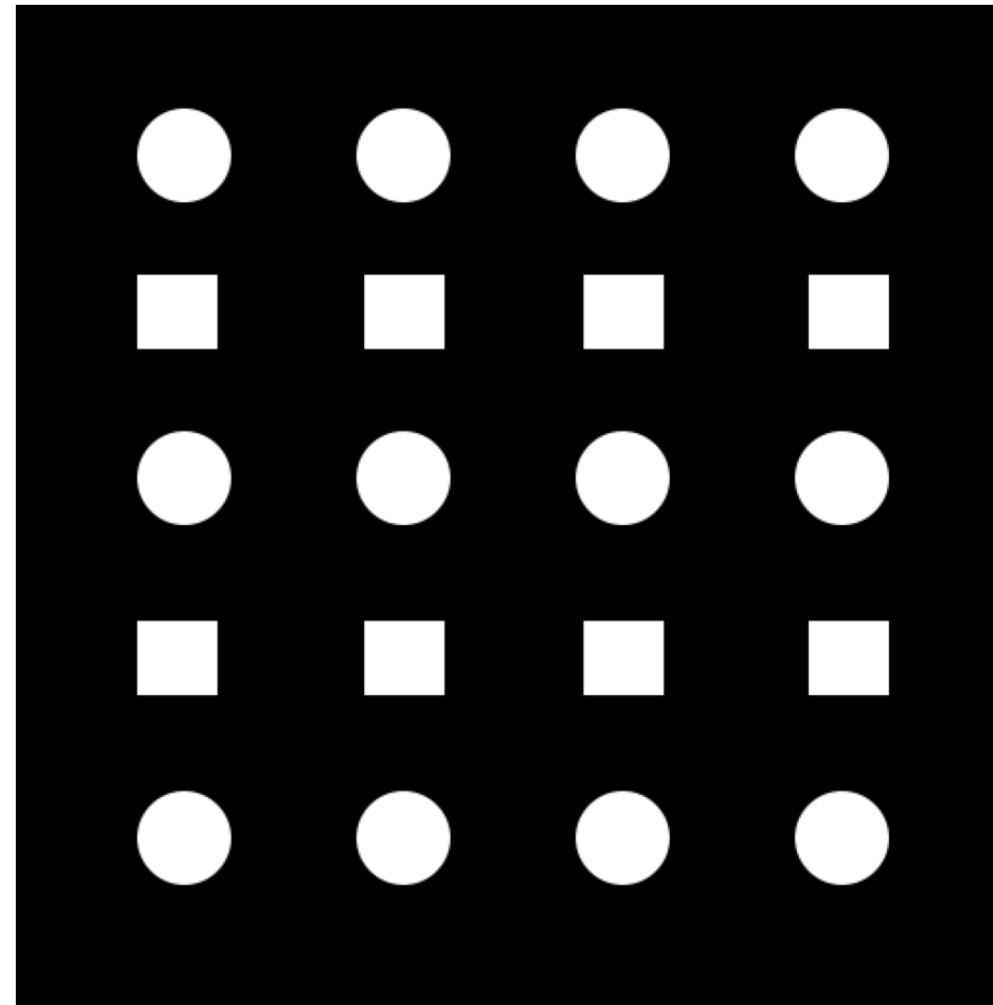
Square  
structuring element



Circular  
structuring element

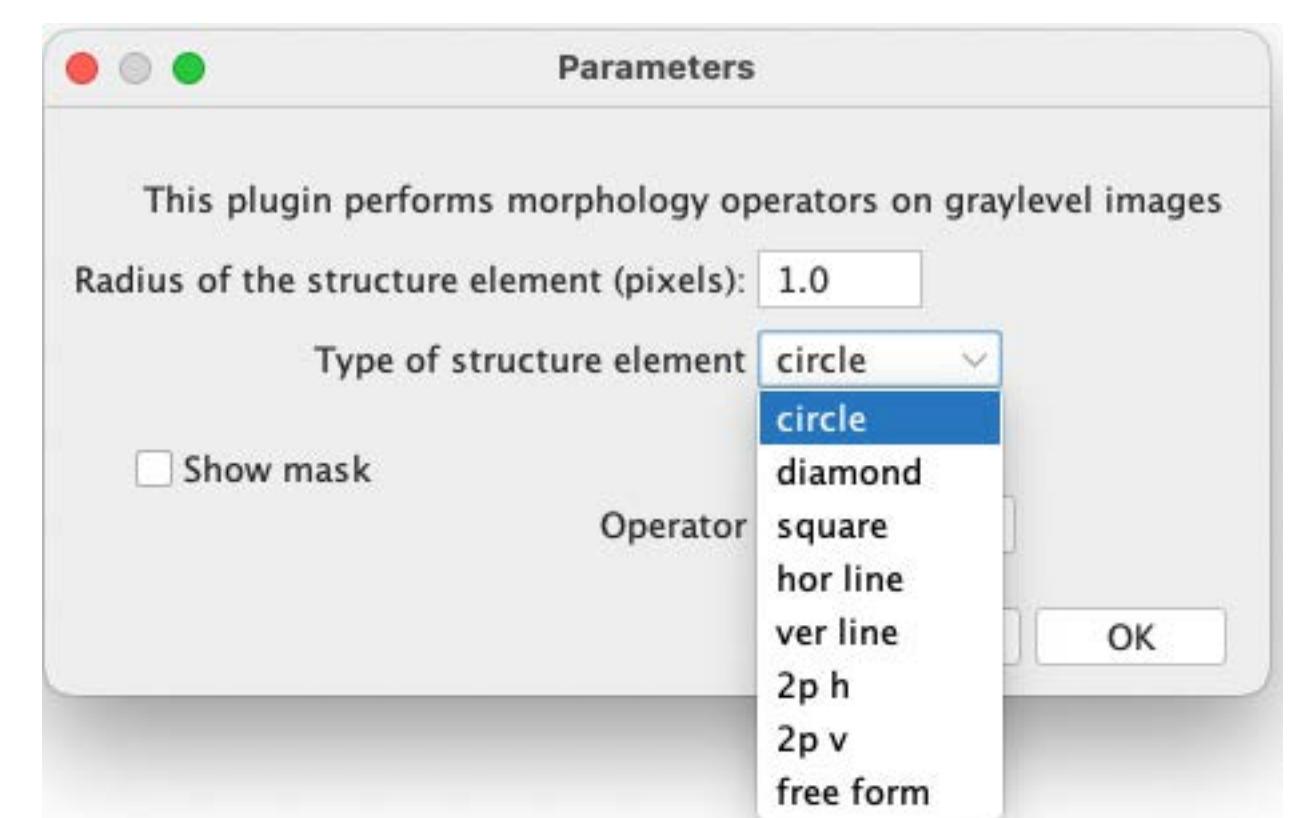
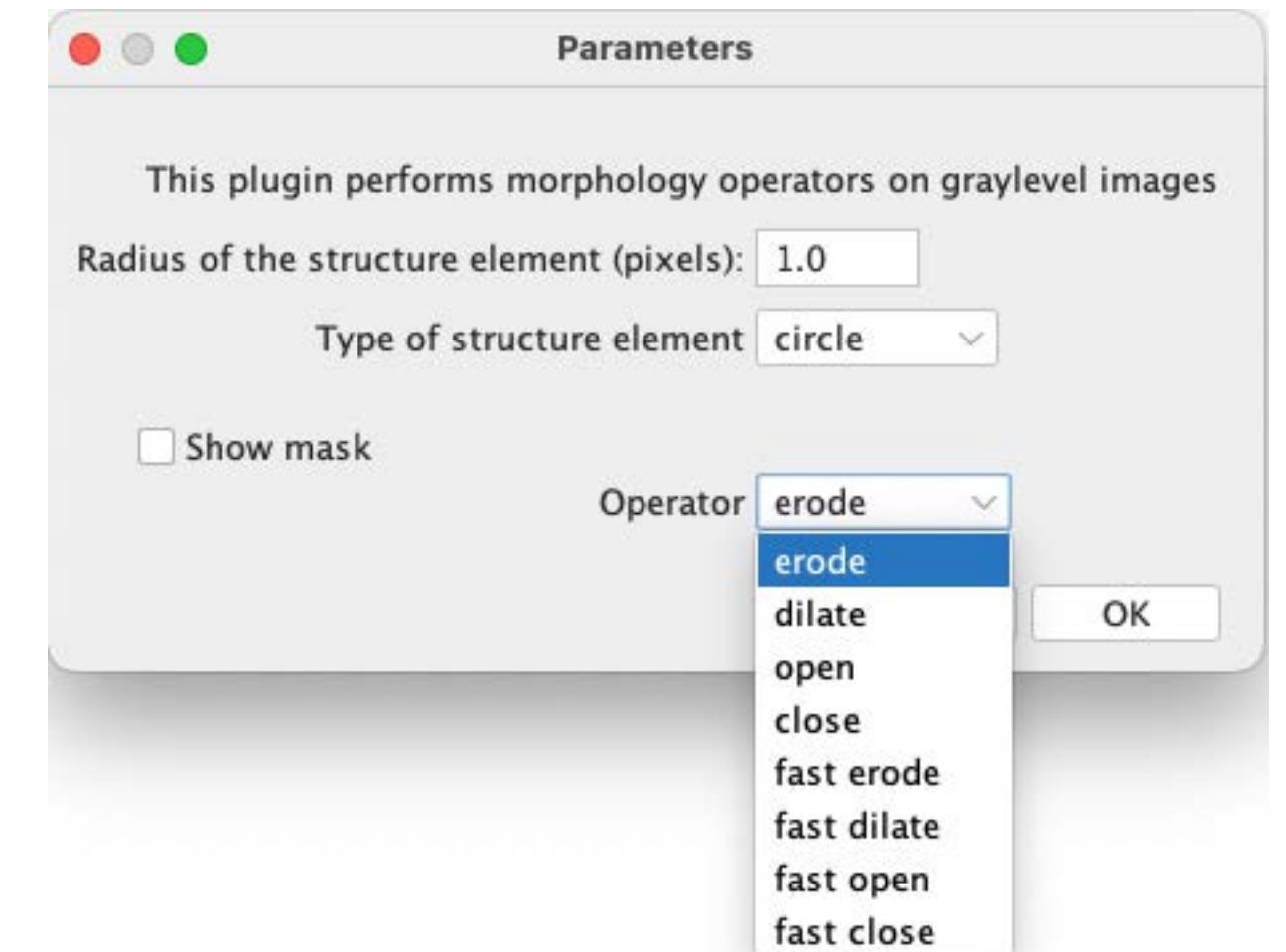
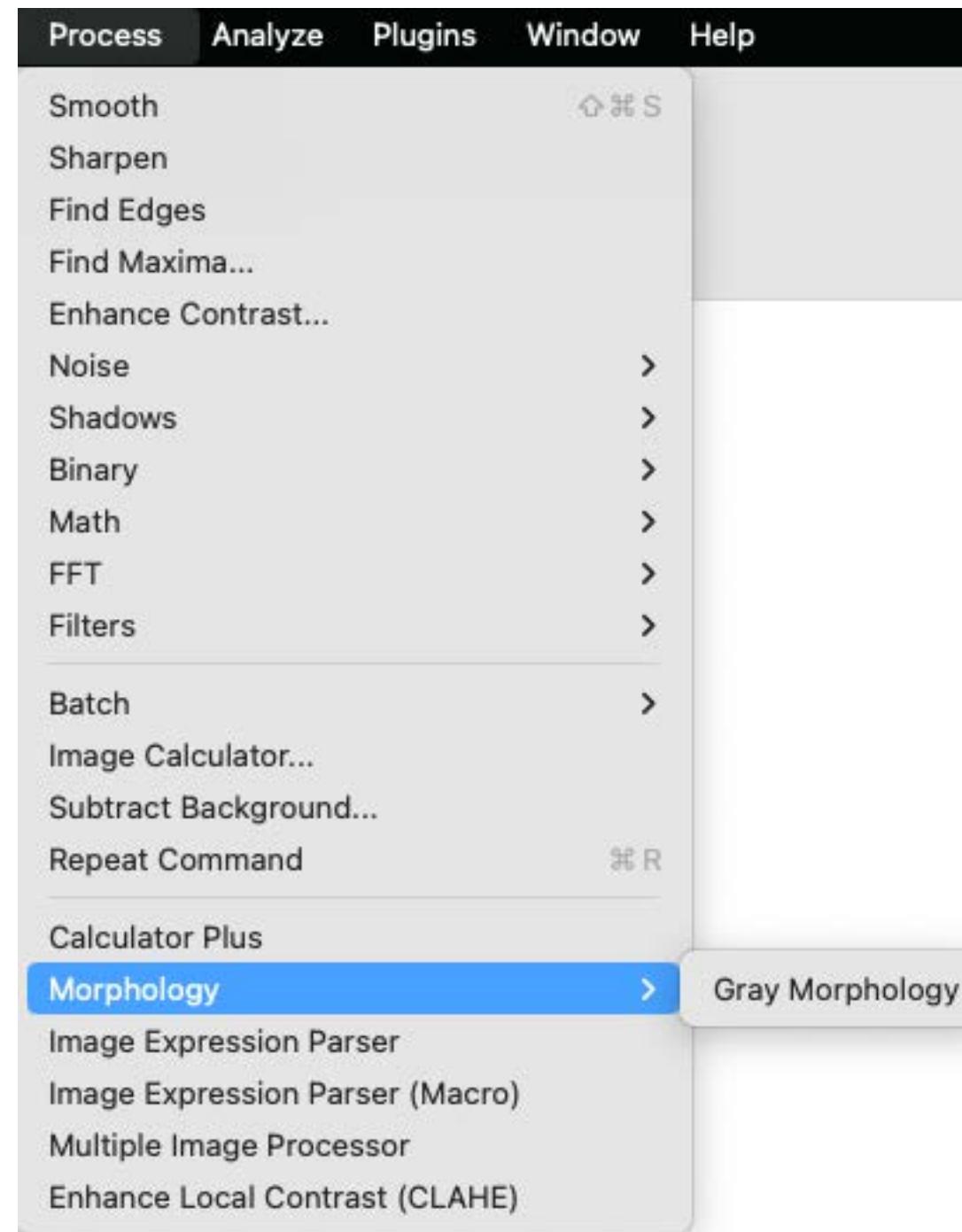
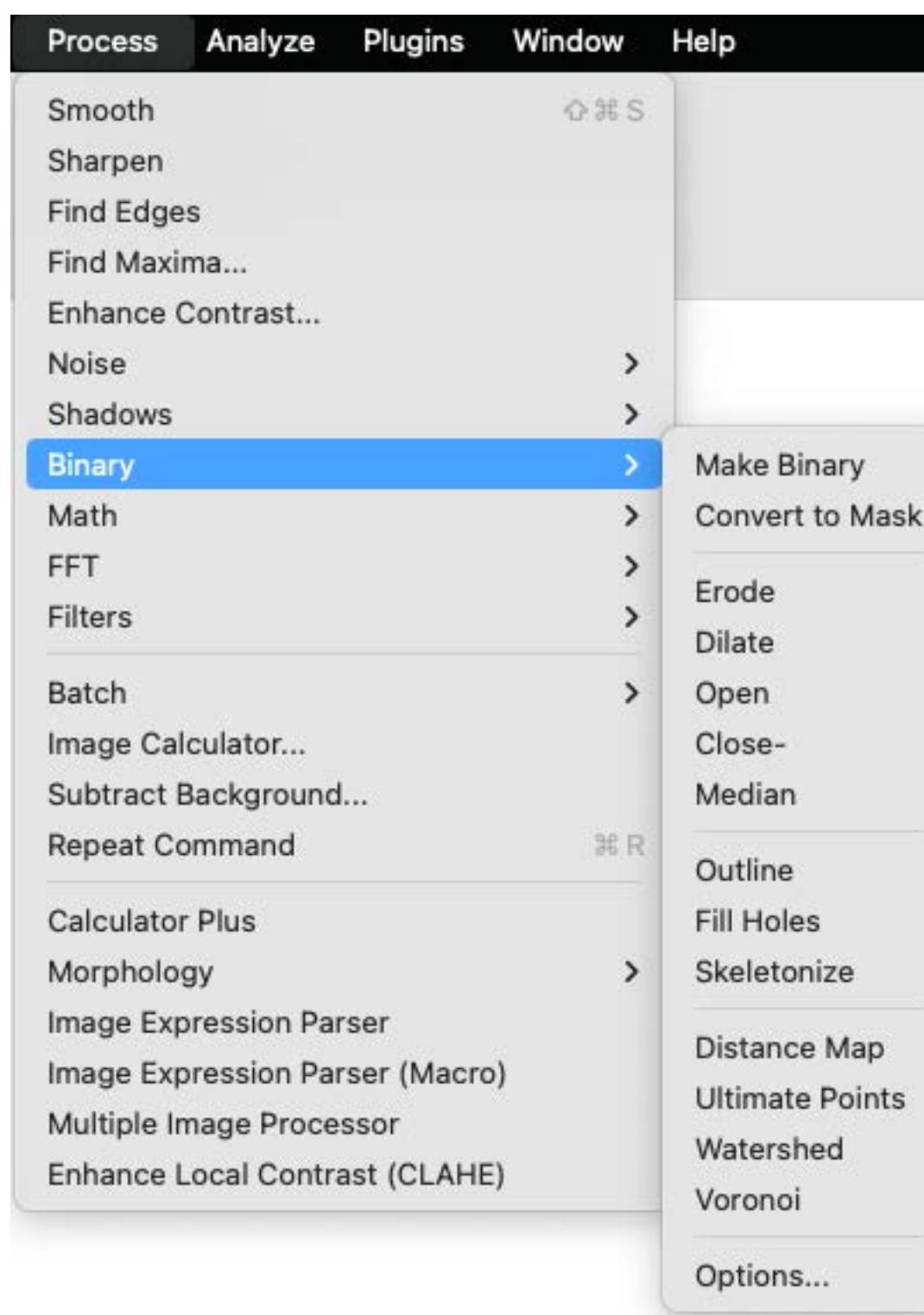


# Structuring Element Shape

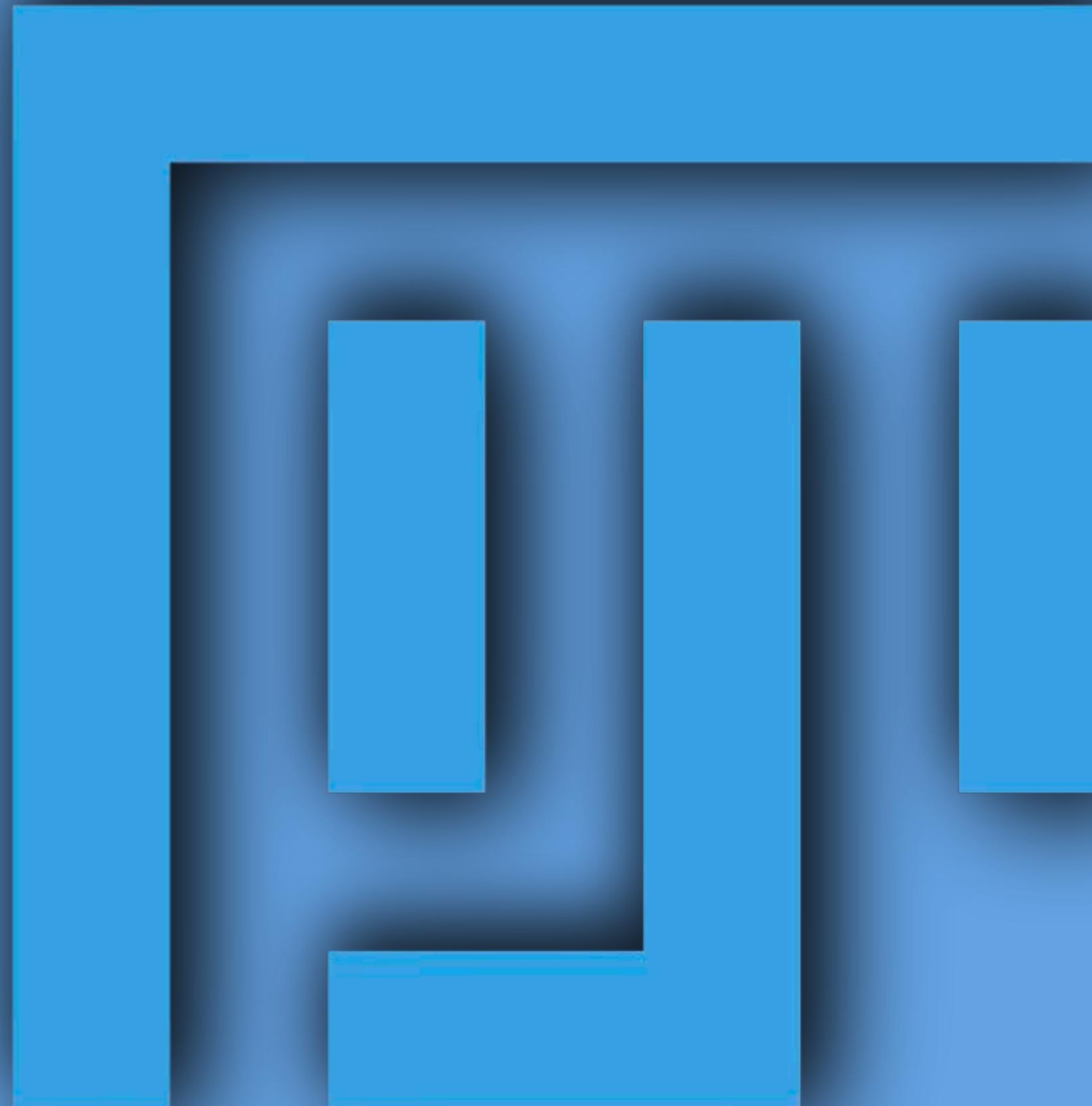


BE CAREFUL!

# Morphology in Fiji



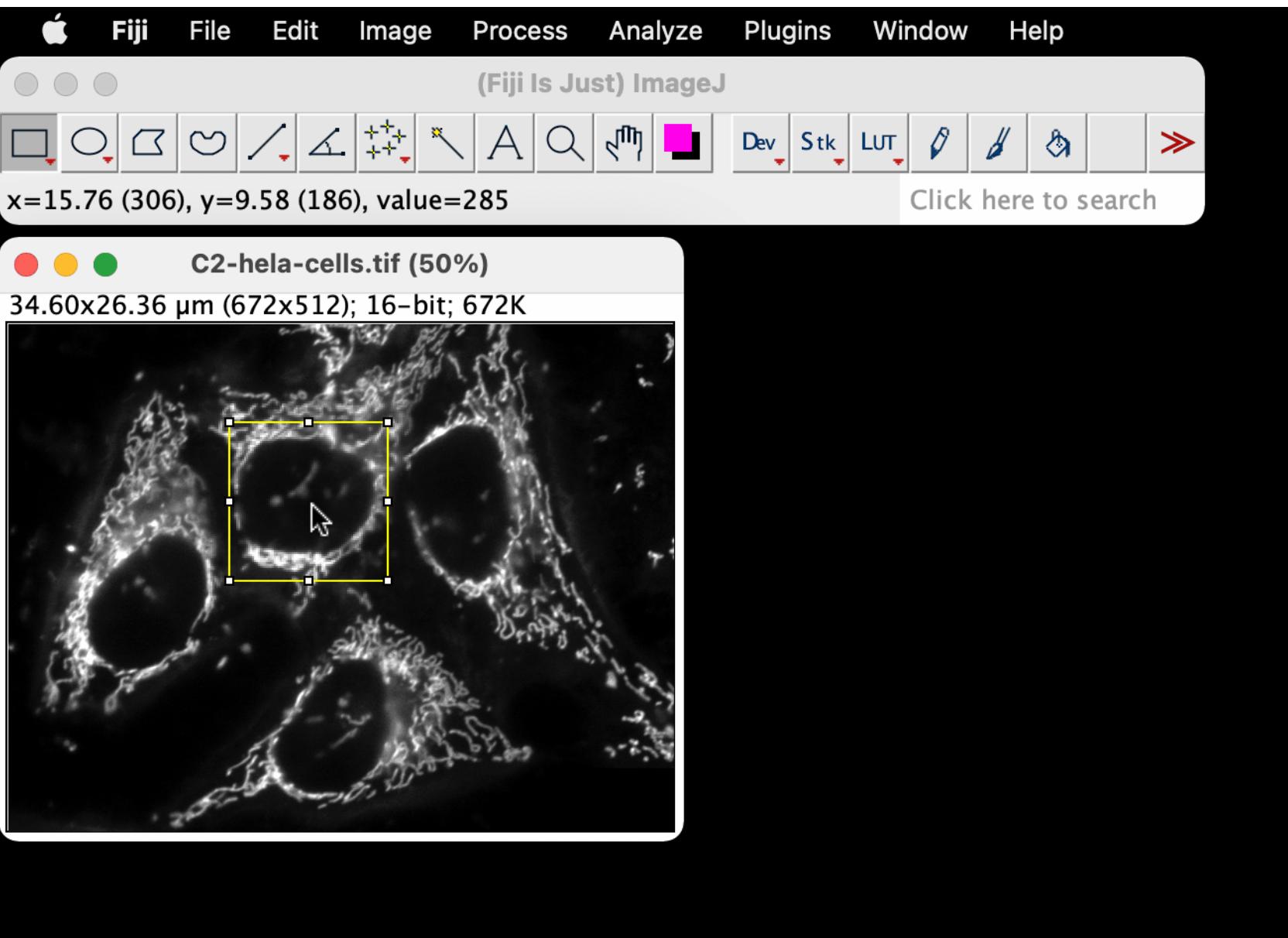
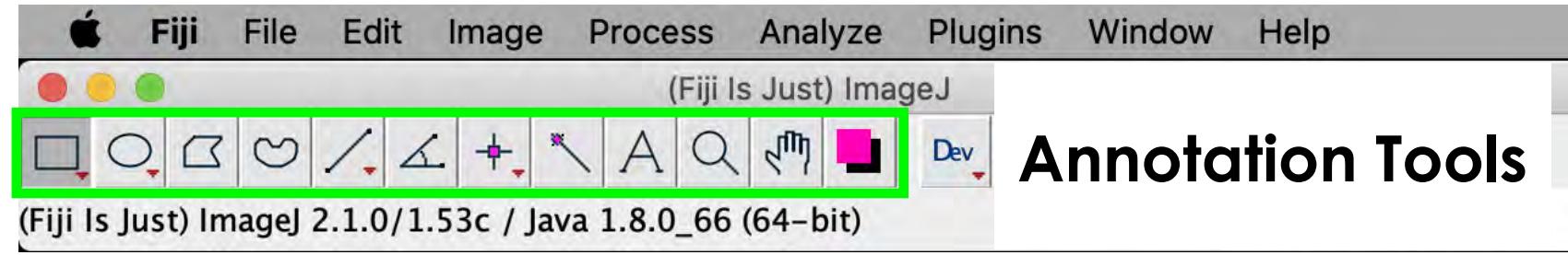
# Morphology — exercises



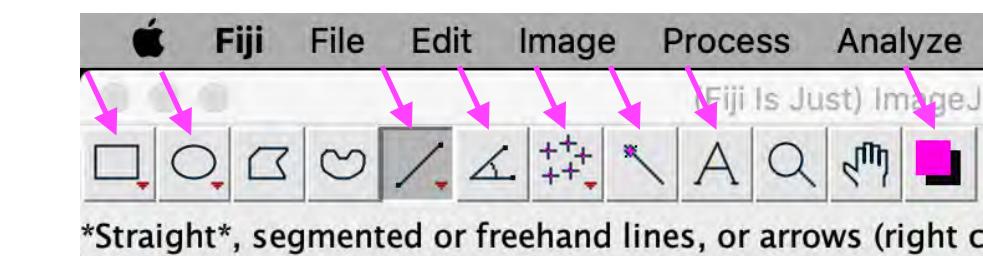
3.4 - Morphological filters - Binary  
3.5 - Morphological filters - Gray

# ROI Manager

# Region Of Interest (ROI)

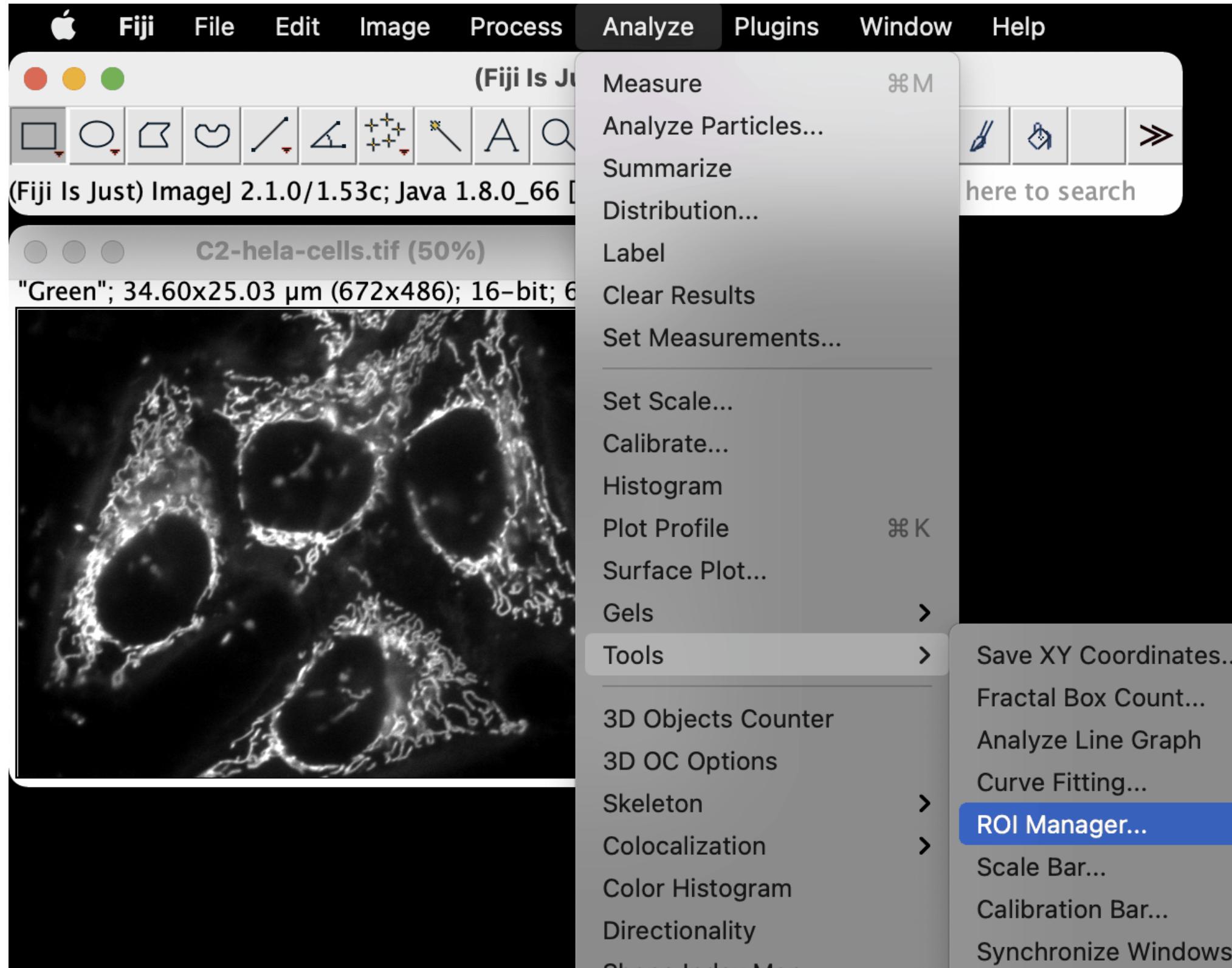


Secondary/Right Click  
for more options



Double Click to  
set/change properties

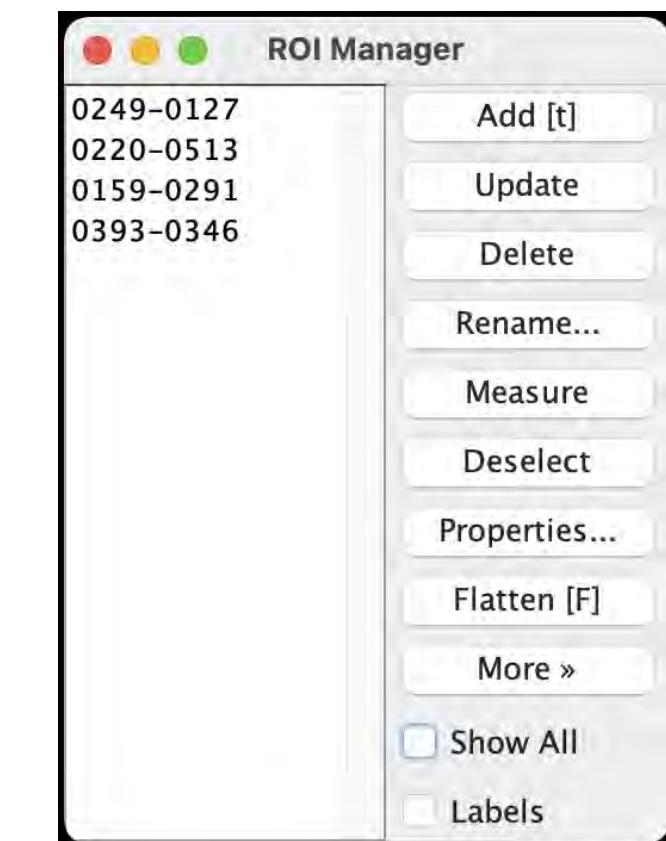
# ROI Manager



Analyze > Tools > ROI Manager...

t \*

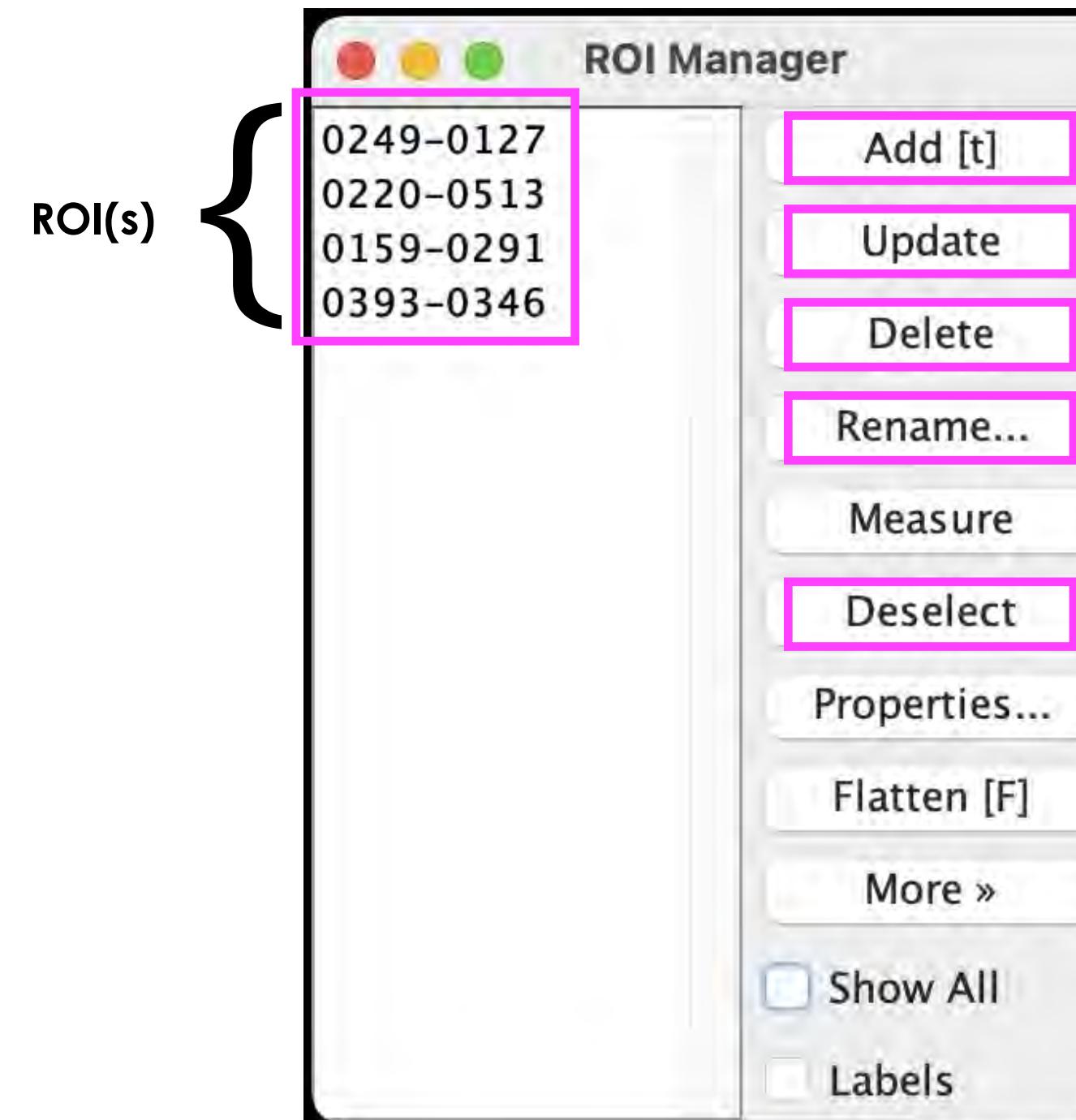
**The ROI Manager is a tool for working with *multiple* selections (ROIs).**



**\*(cmd) + t can be used to both to open the ROI Manager and/or add a new ROI to the Manager.**

**\*\*shift + e can be used to draw the last ROI.**

# ROI Manager



**Add new ROI ([t](#)).**

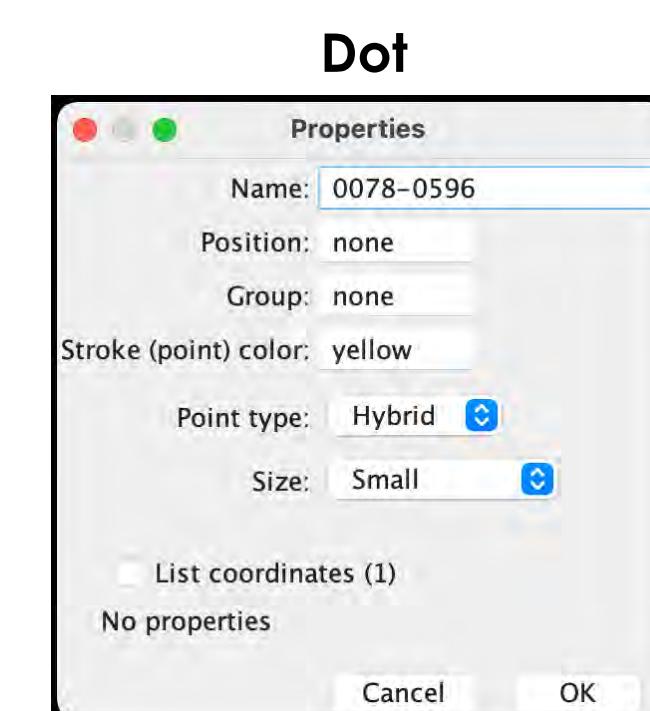
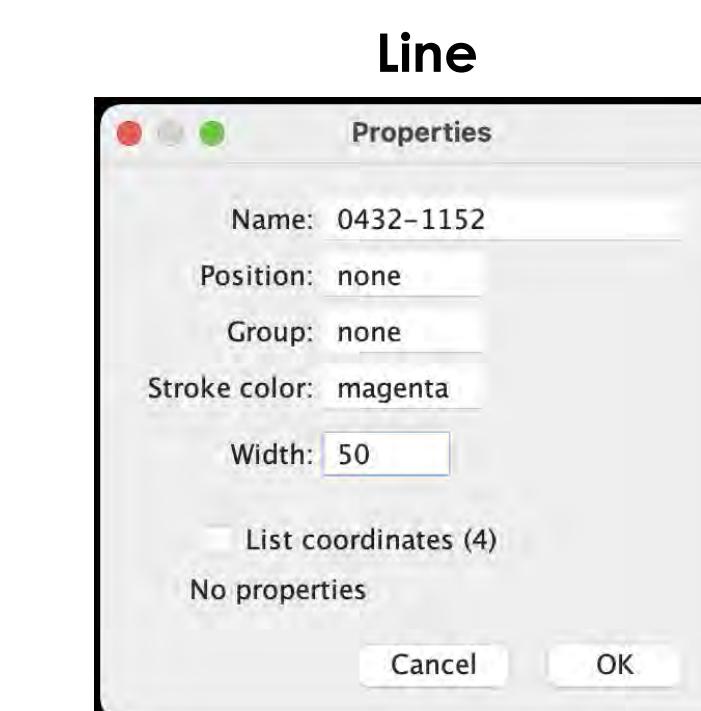
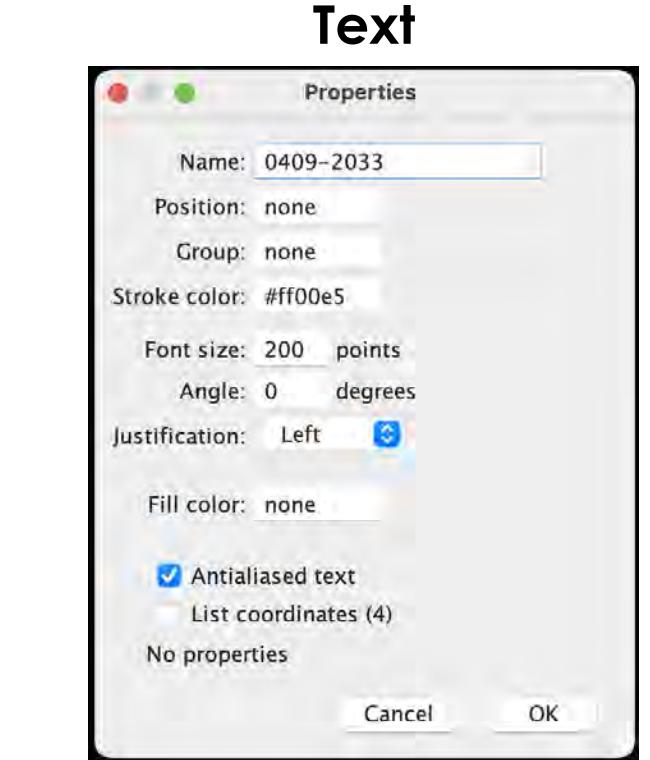
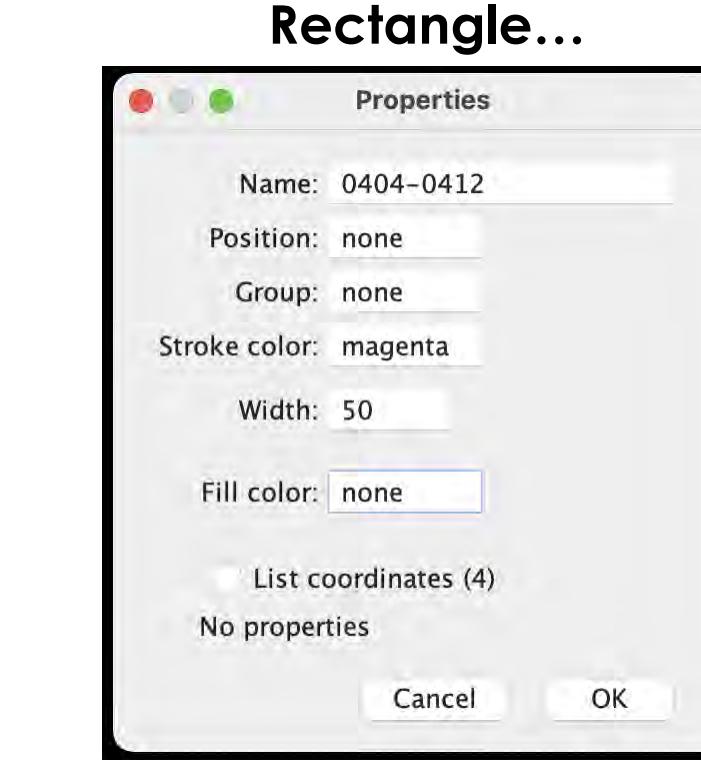
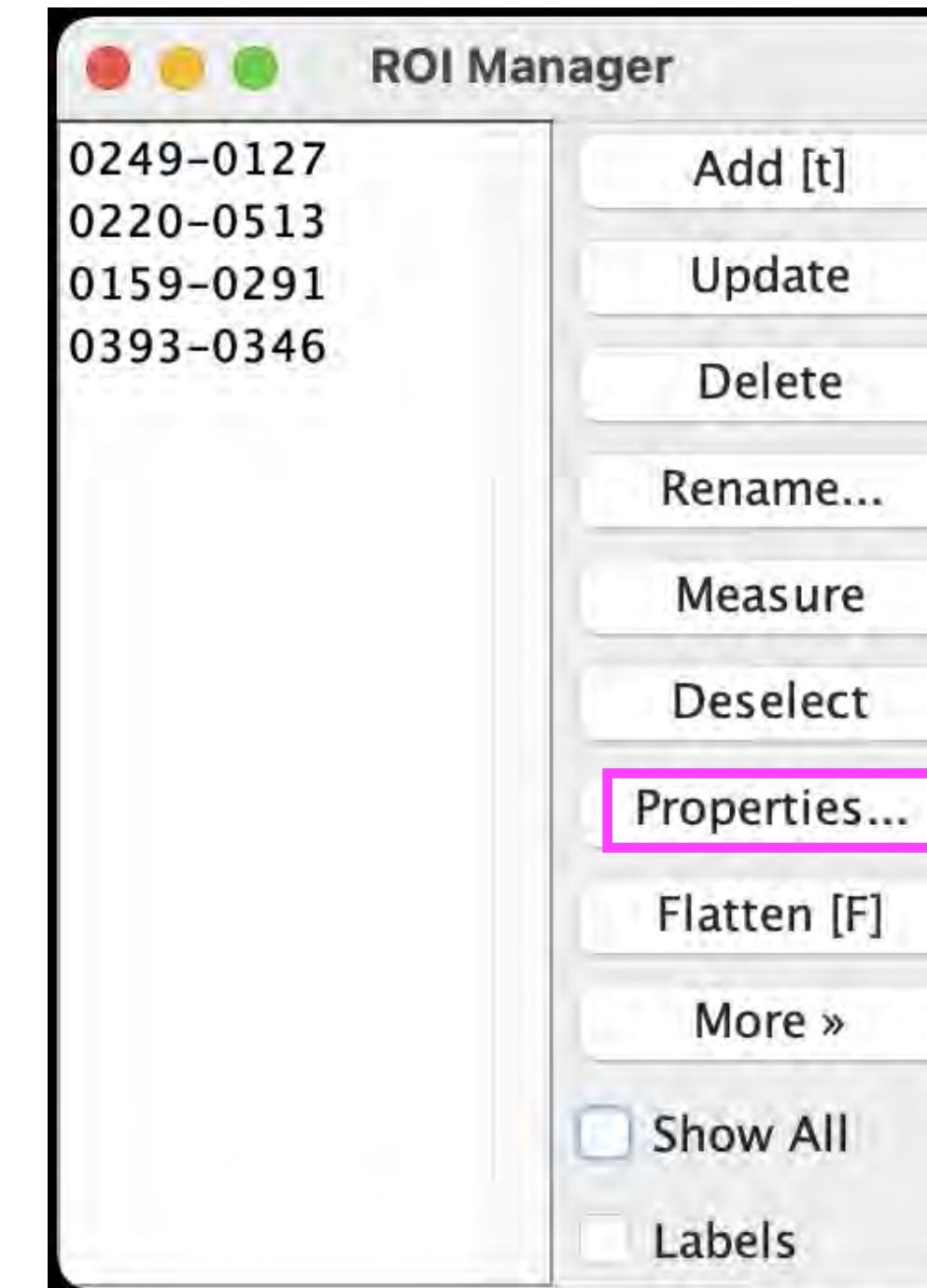
After modifying a ROI, you can use **Update** to save the changes.

**Delete selected ROI.** If none is selected, delete all.

**Rename selected ROI.**

**Deselect one or more selected ROI.**

# ROI Manager

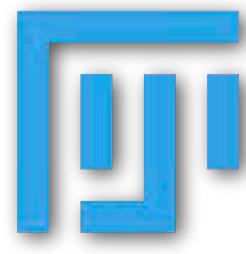


Change the properties of the selected ROI(s)

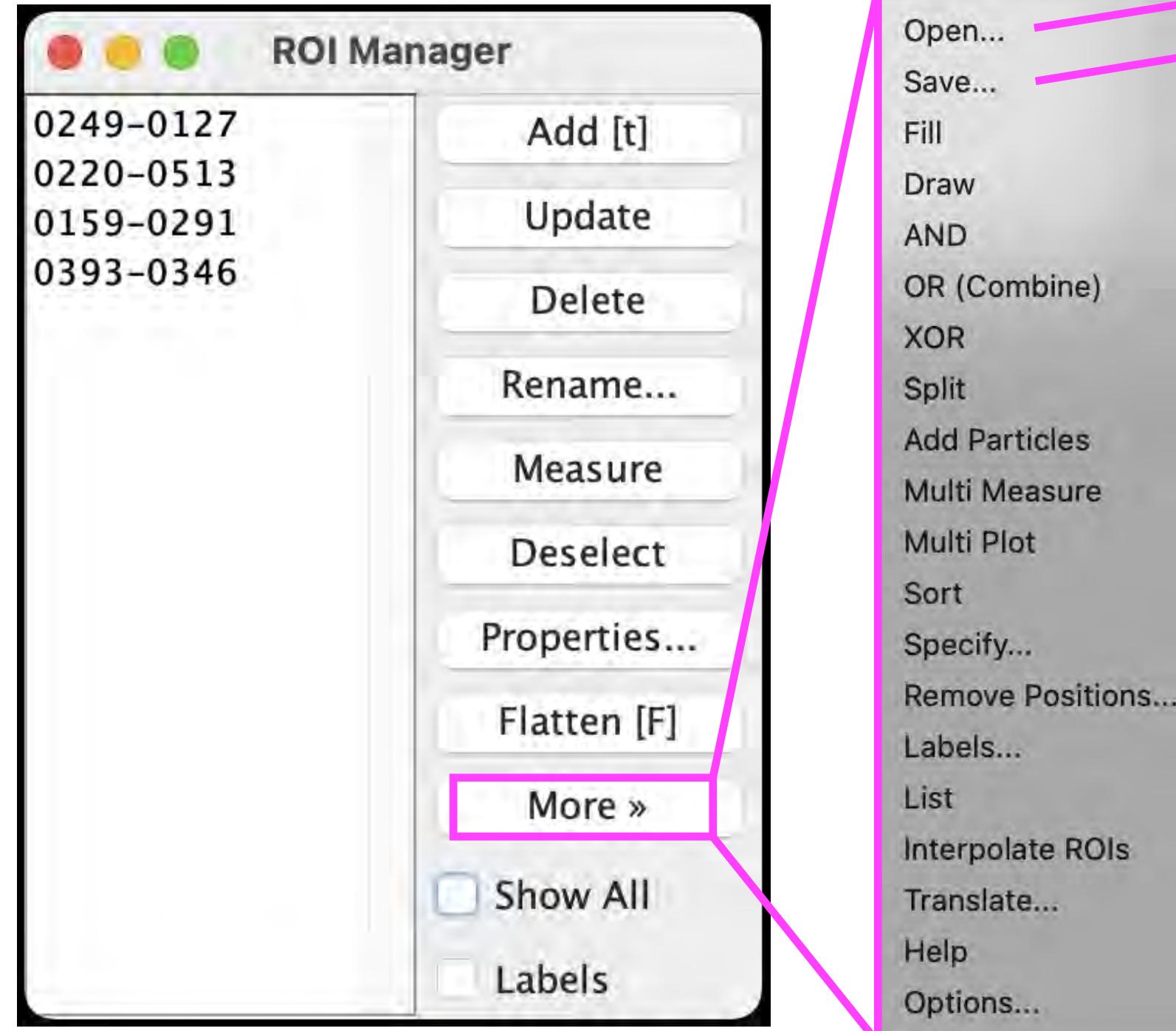
Properties...

“Analyze” menu

“Edit” menu



# ROI Manager



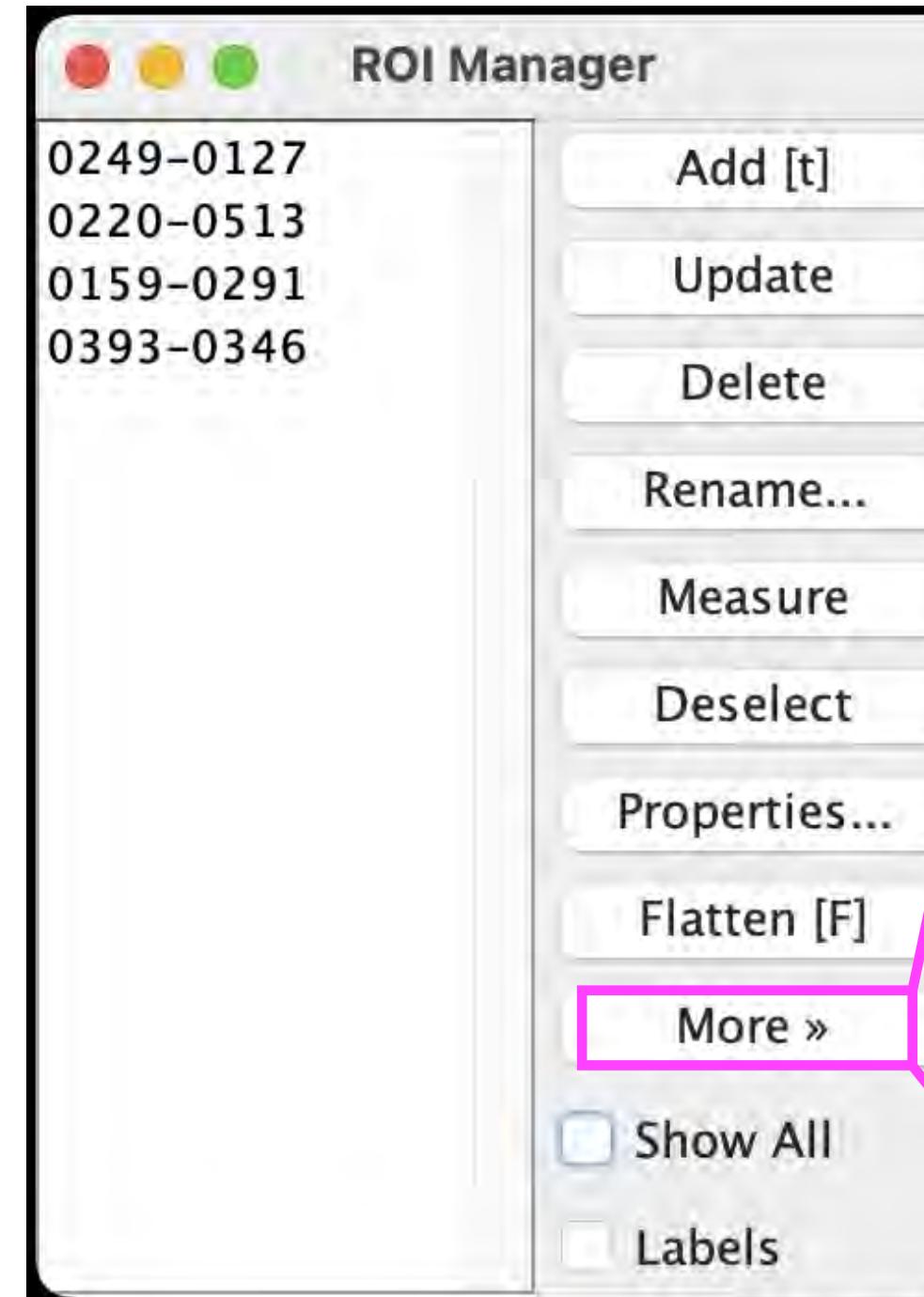
Open a saved ROI.zip file (drag & drop works as well)

Save ROI(s) as a .zip file

“Analyze” menu

“Image” menu

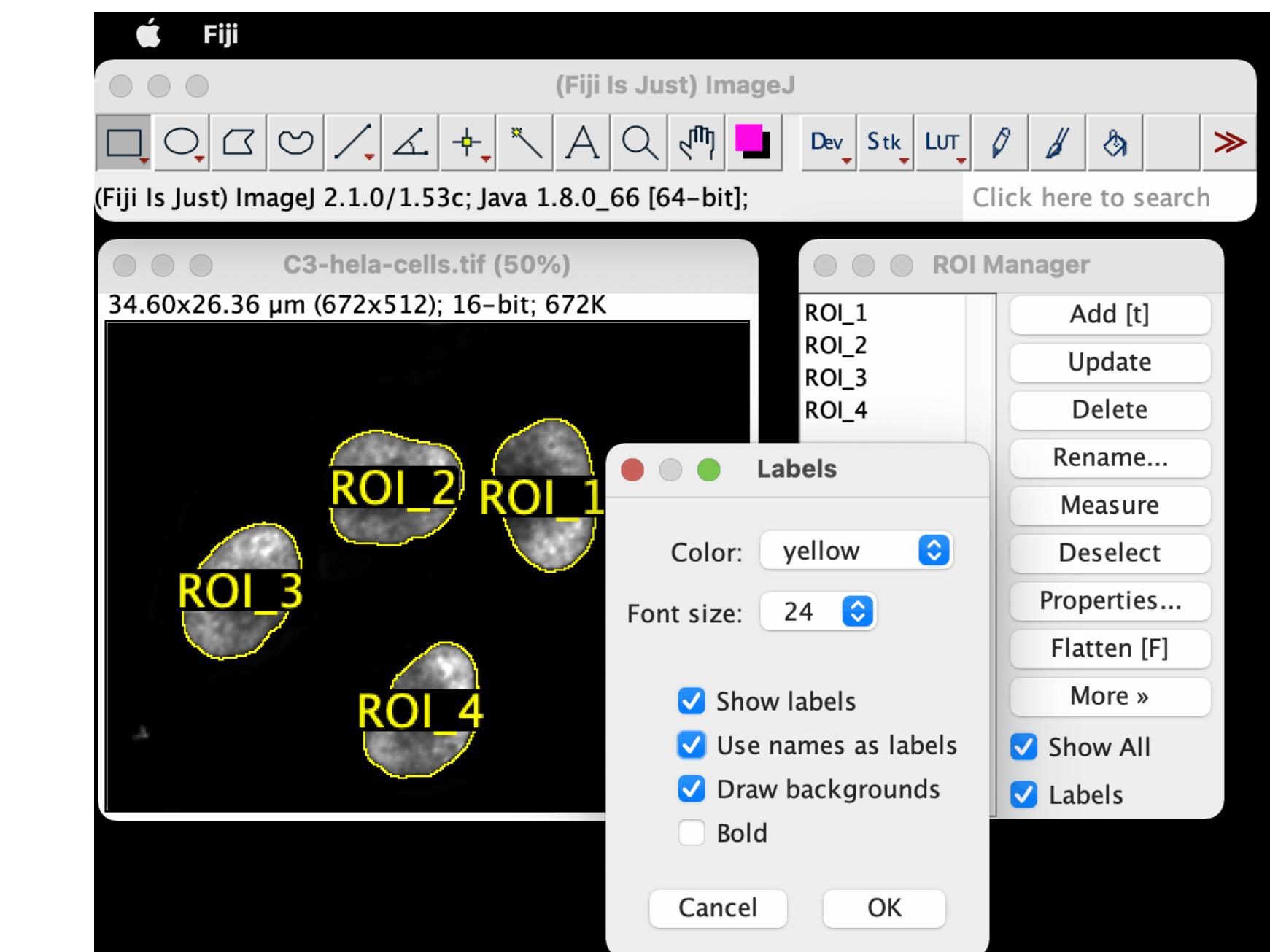
# ROI Manager



- Open...
- Save...
- Fill
- Draw
- AND
- OR (Combine)
- XOR
- Split
- Add Particles
- Multi Measure
- Multi Plot
- Sort
- Specify...
- Remove Positions...
- Labels...
- List
- Interpolate ROIs
- Translate...
- Help
- Options...

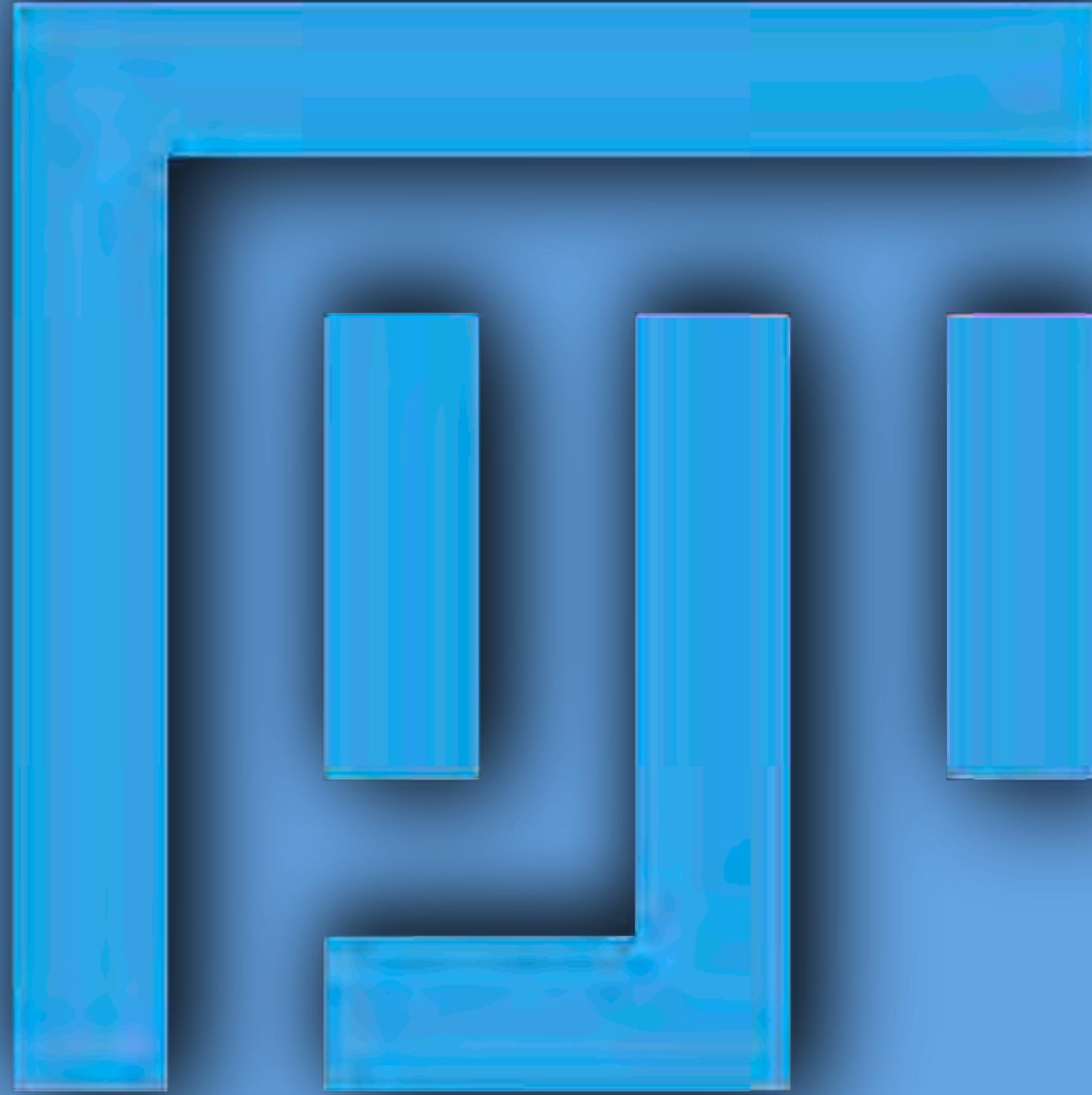
Open a saved ROI.zip file (drag & drop works as well)  
Save ROI(s) as a .zip file

Set ROI(s) label options



\*Labels option can also be found under “Image > Overlay”

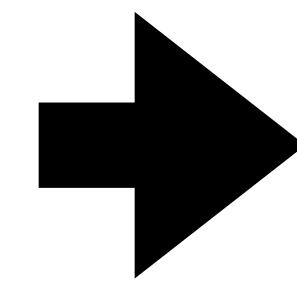
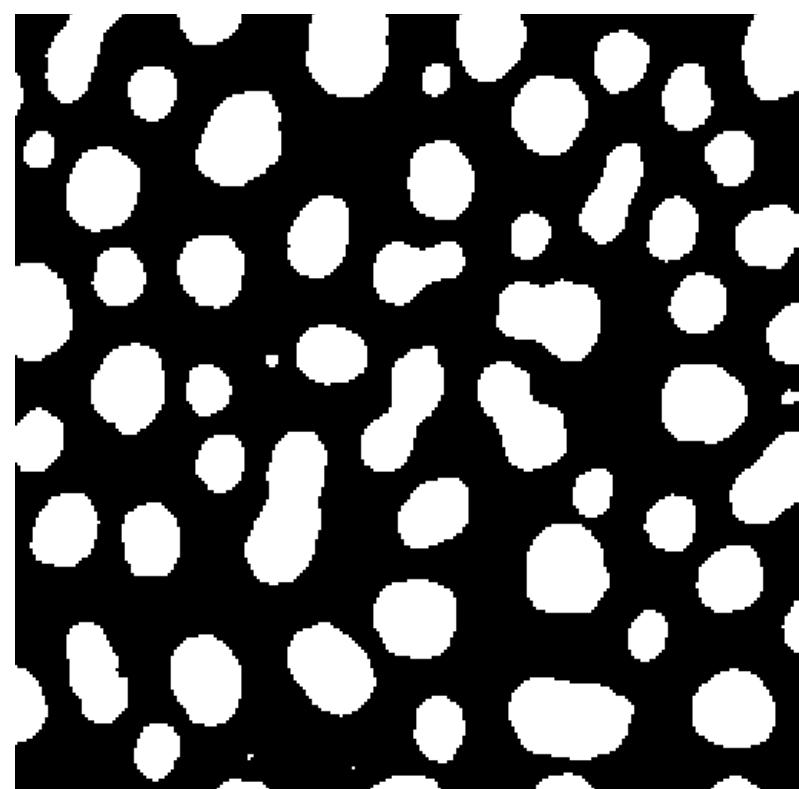
\*\*“Use names as labels” can be also found  
under “More>Options...”



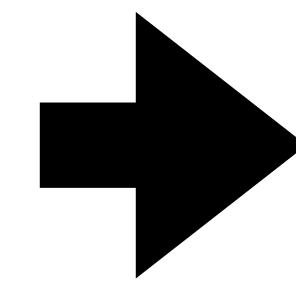
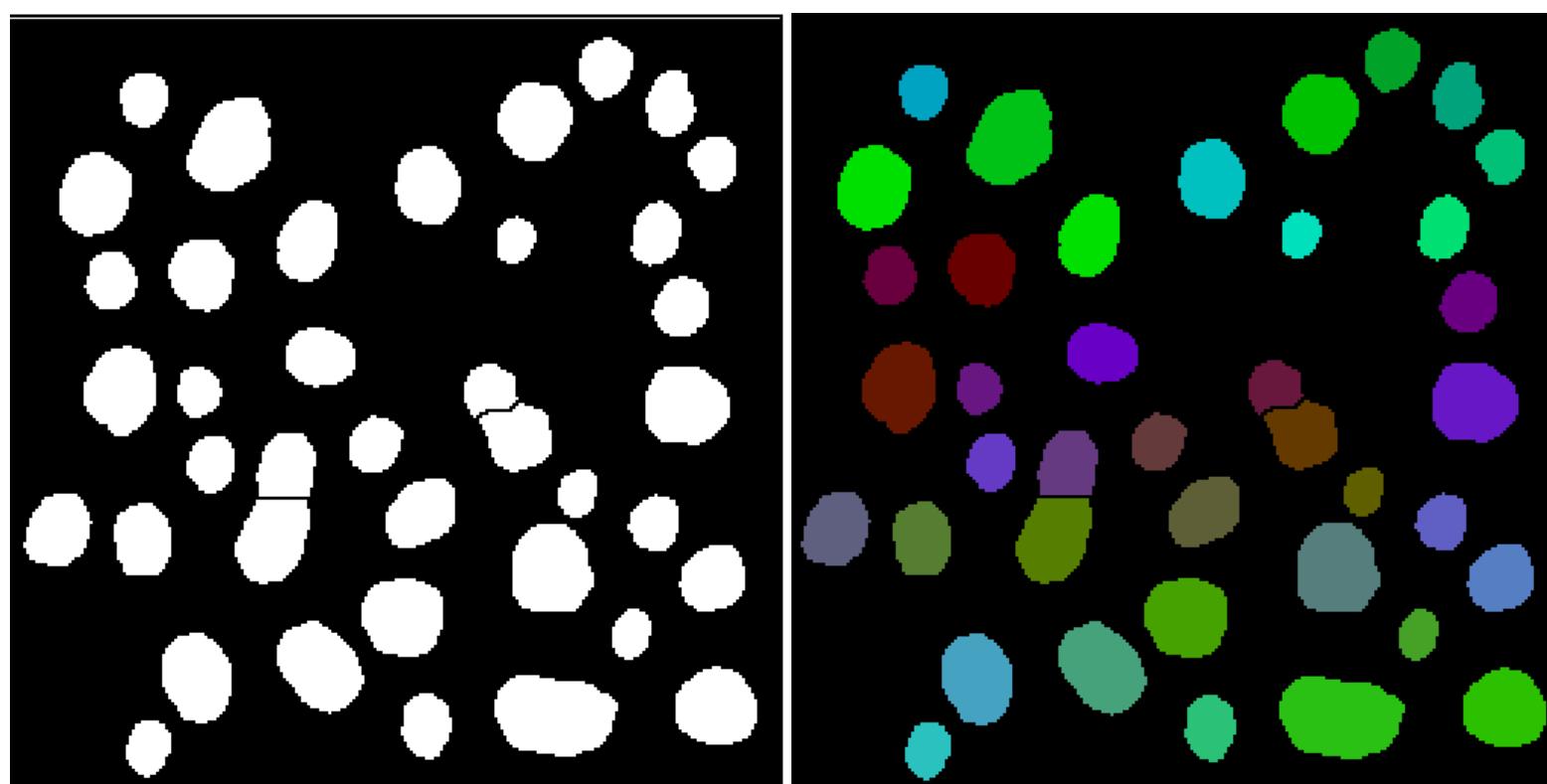
**Live Demo of ROI manager**

# Segmentation And Measurements

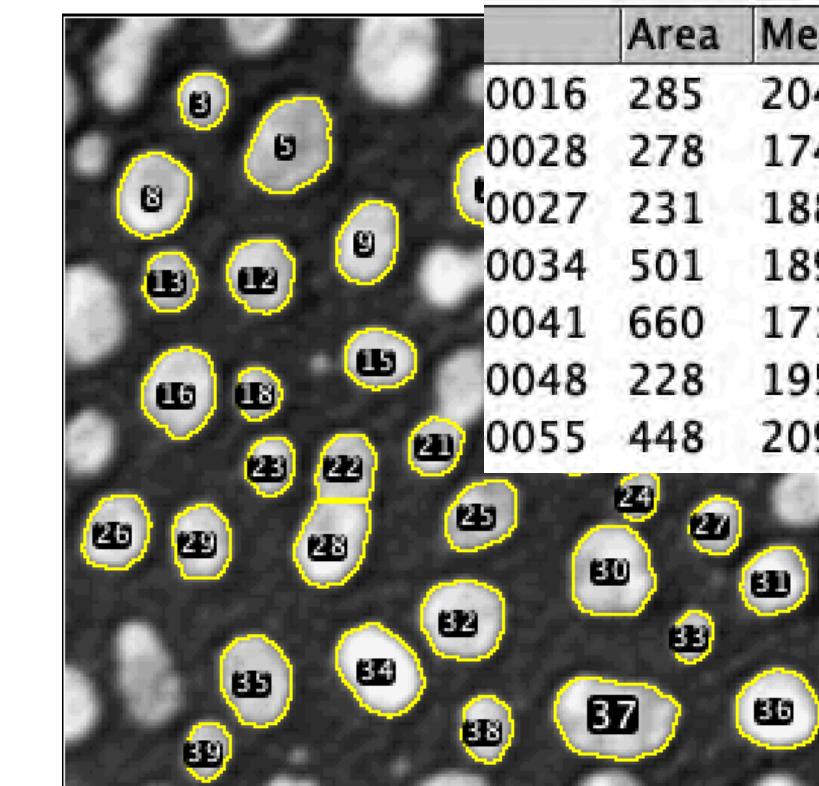
Binary mask



Process and segment

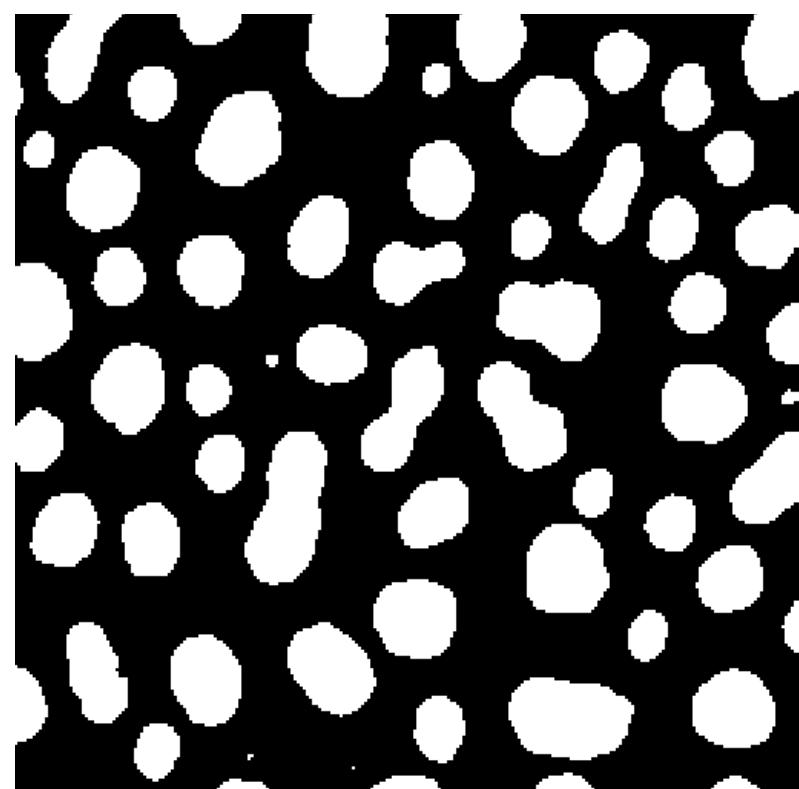


Export and Measure

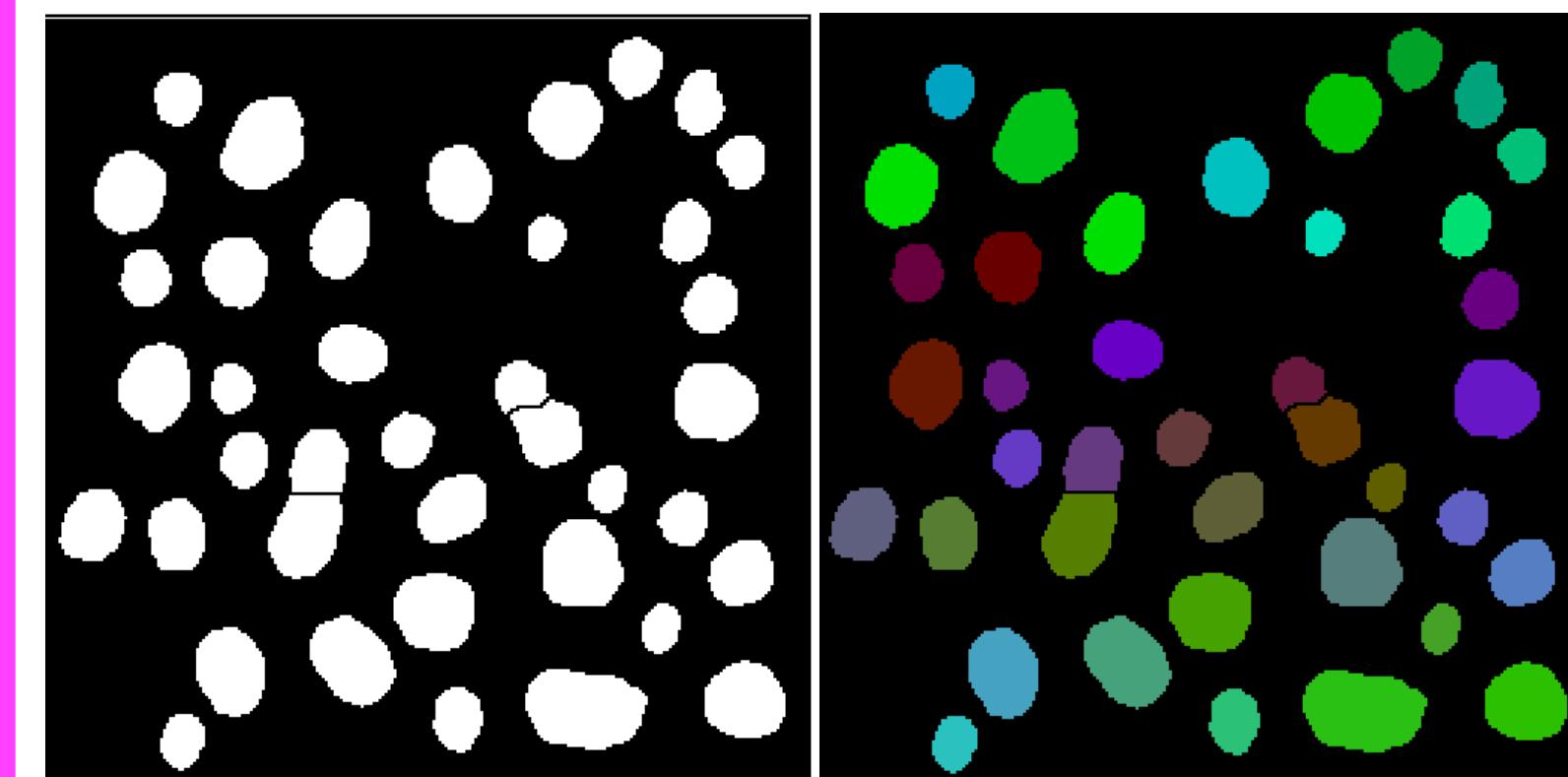


Results			
	Area	Mean	X
0016	285	204.29474	197.
0028	278	174.84892	219.
0027	231	188.46753	45.0
0034	501	189.14172	174.
0041	660	171.69697	73.8
0048	228	195.89474	233.
0055	448	209.03571	138.

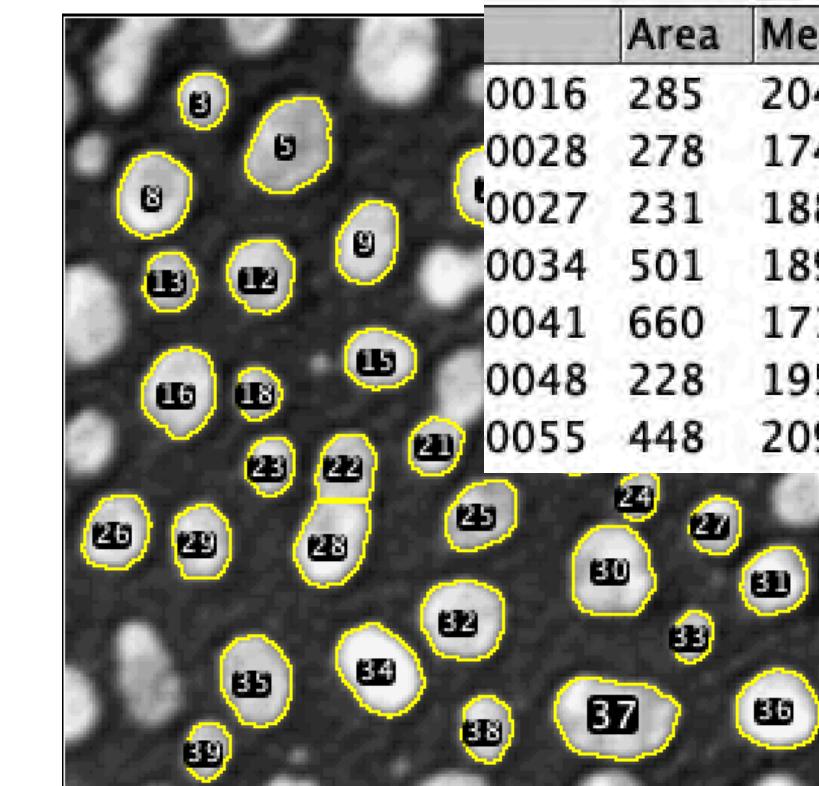
Binary mask



Process and segment



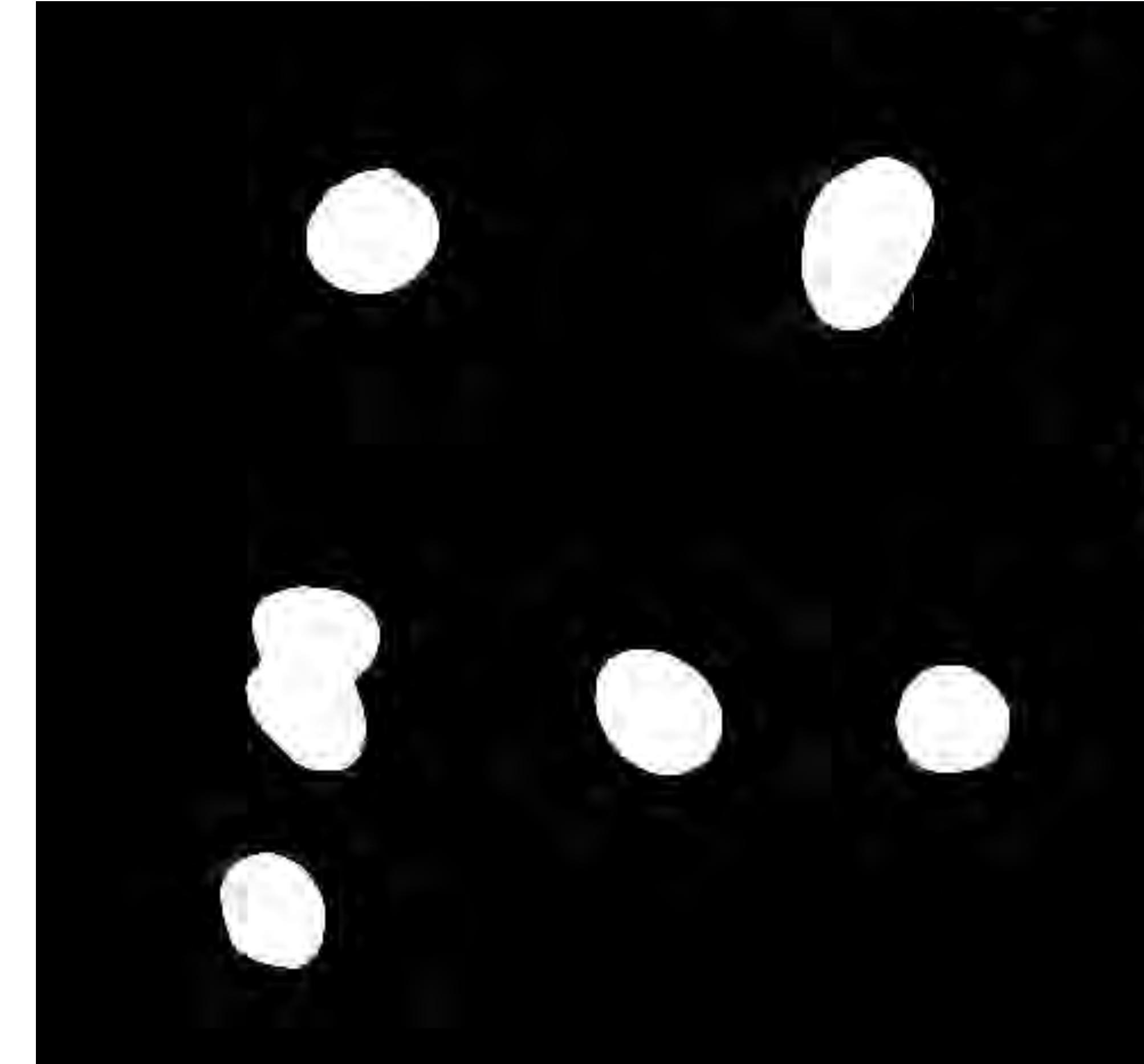
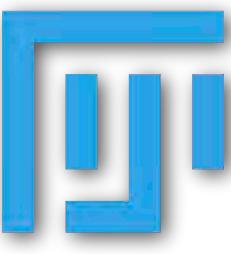
Export and Measure



Results			
	Area	Mean	X
0016	285	204.29474	197.
0028	278	174.84892	219.
0027	231	188.46753	45.0
0034	501	189.14172	174.
0041	660	171.69697	73.8
0048	228	195.89474	233.
0055	448	209.03571	138.

# Do you see an issue here?

Process and segment

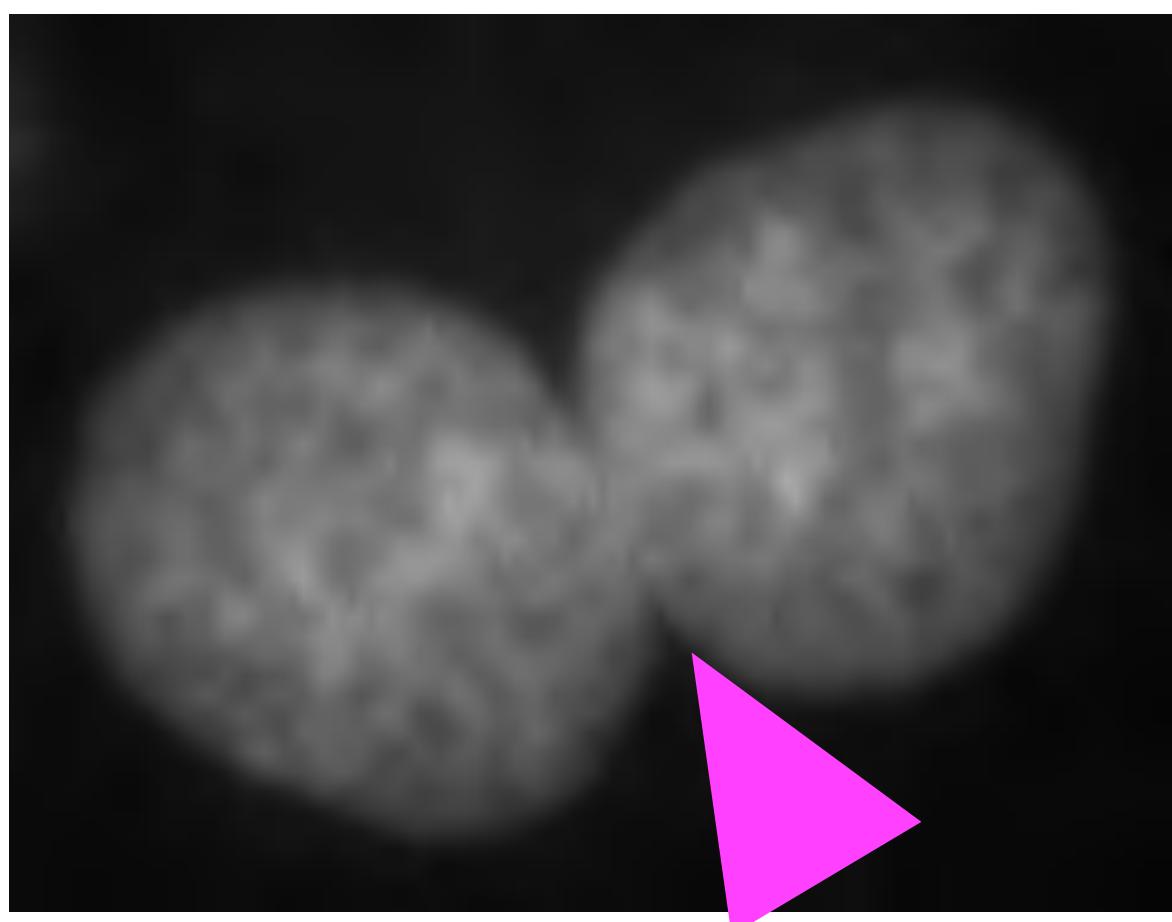


# Solution: Watershed

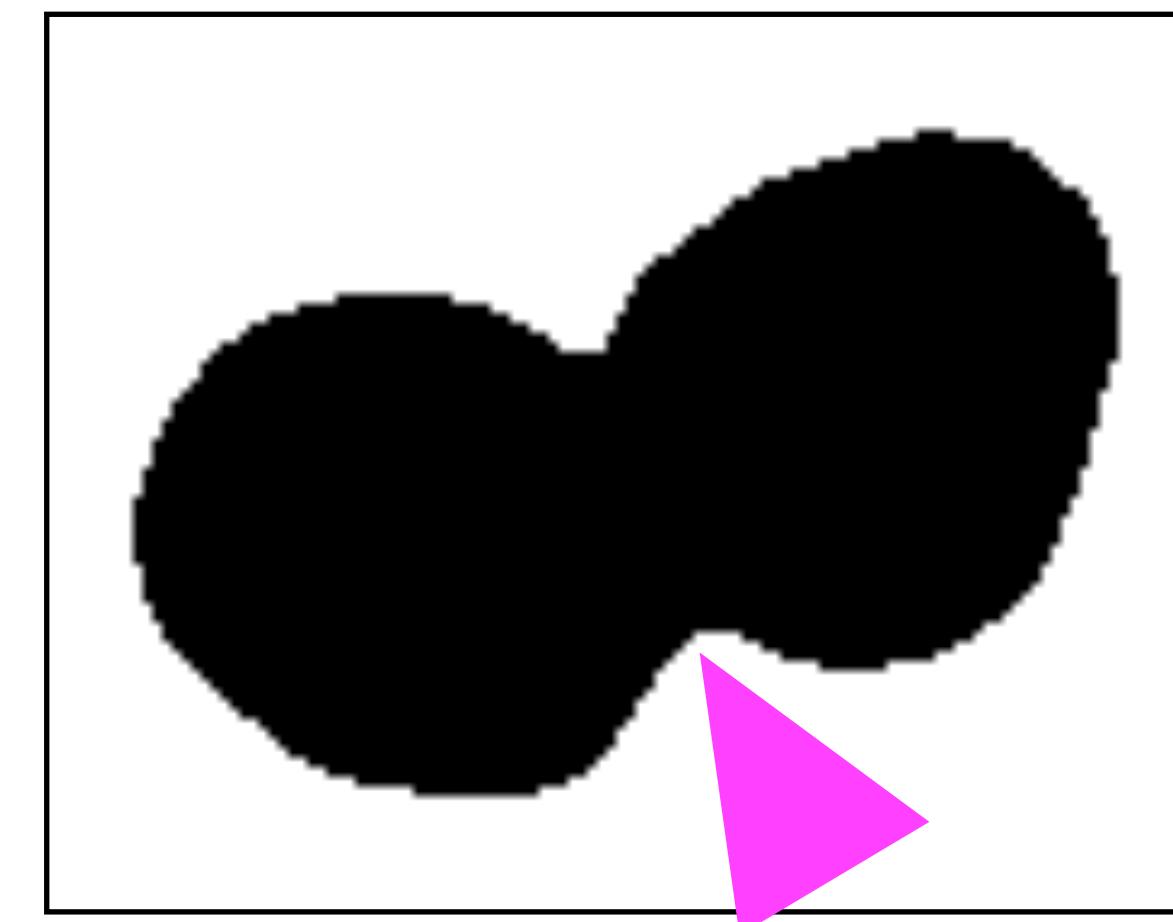
in **Fiji**: Process > Binary > Watershed

**Watershed** is a useful algorithm to try to **separate touching objects**.

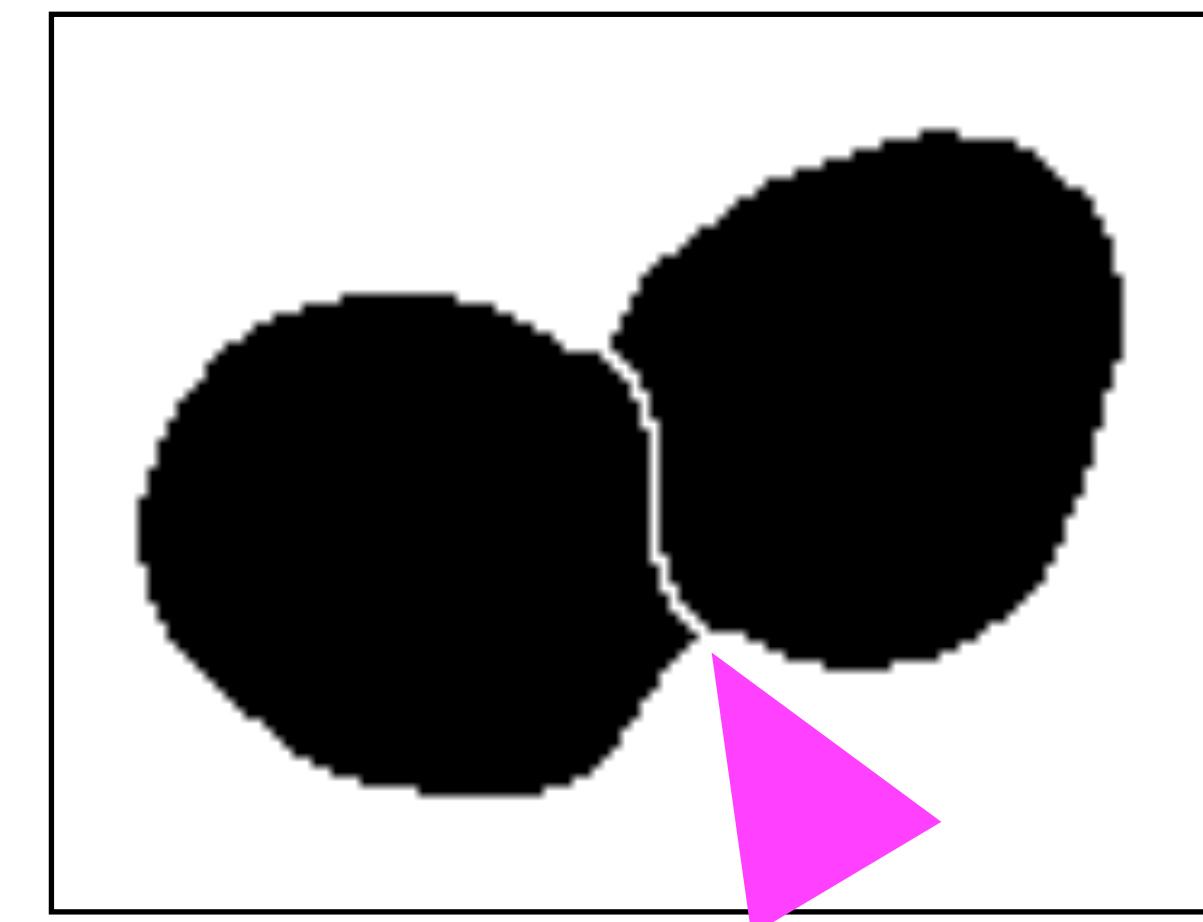
Image



Binary Mask

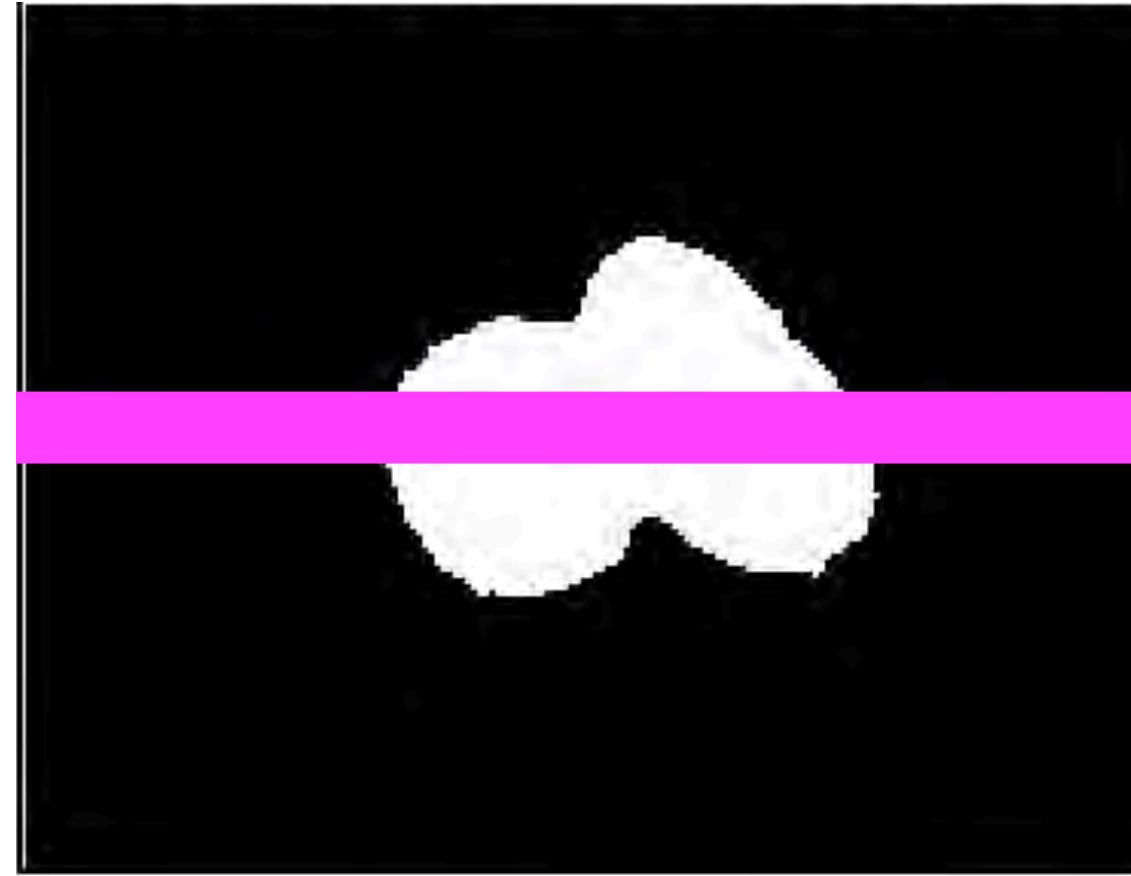


Watershed

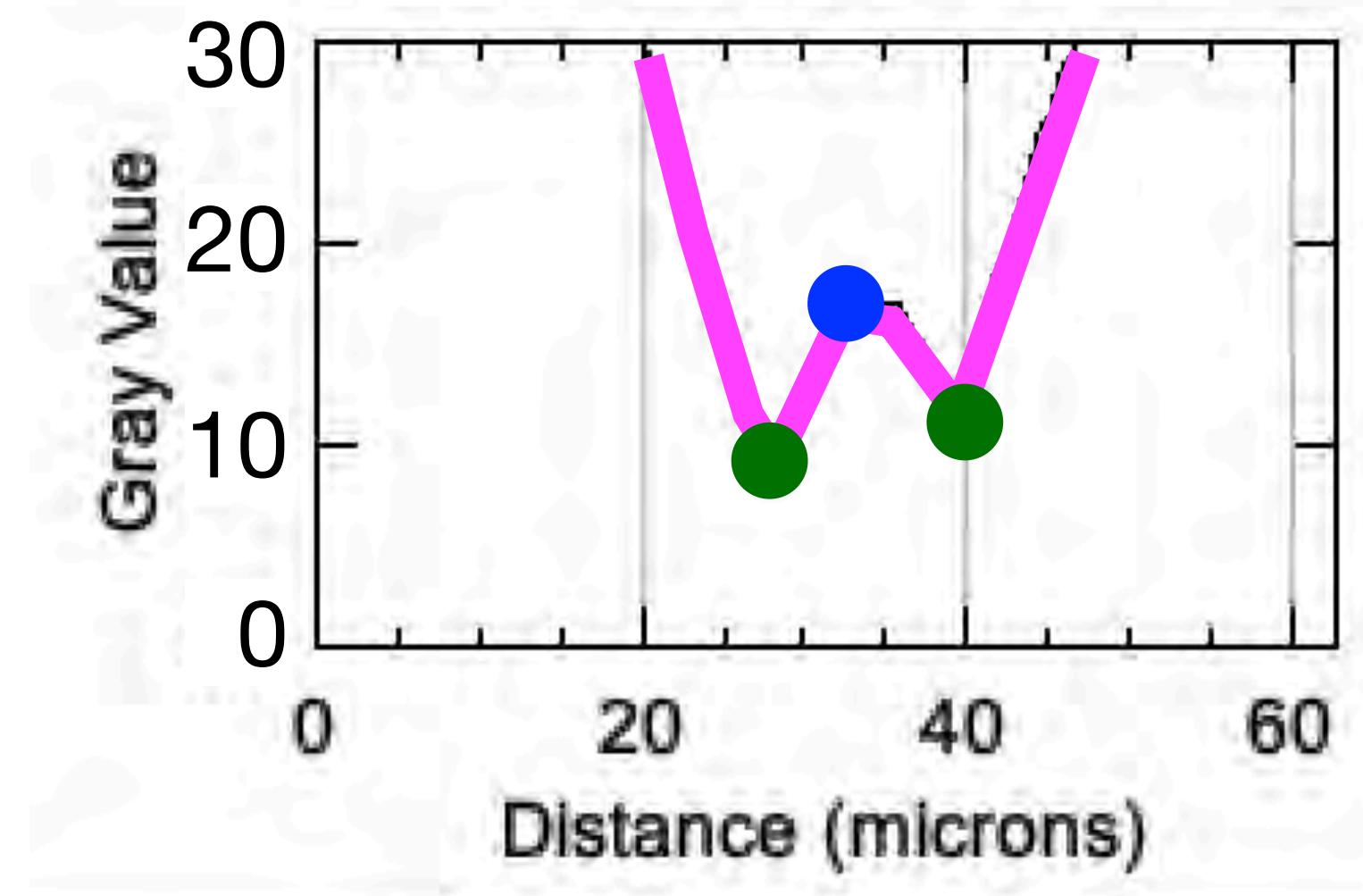
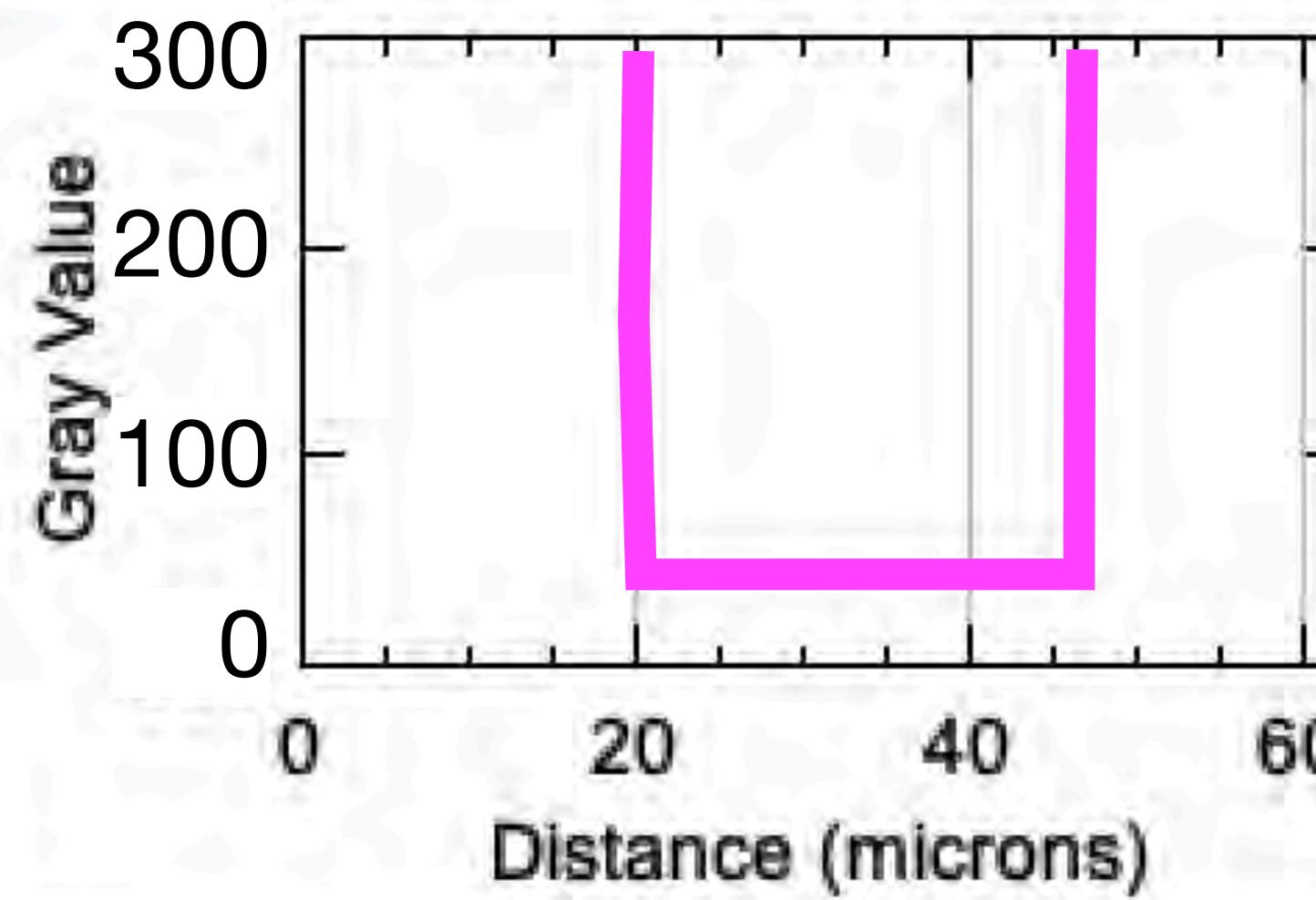
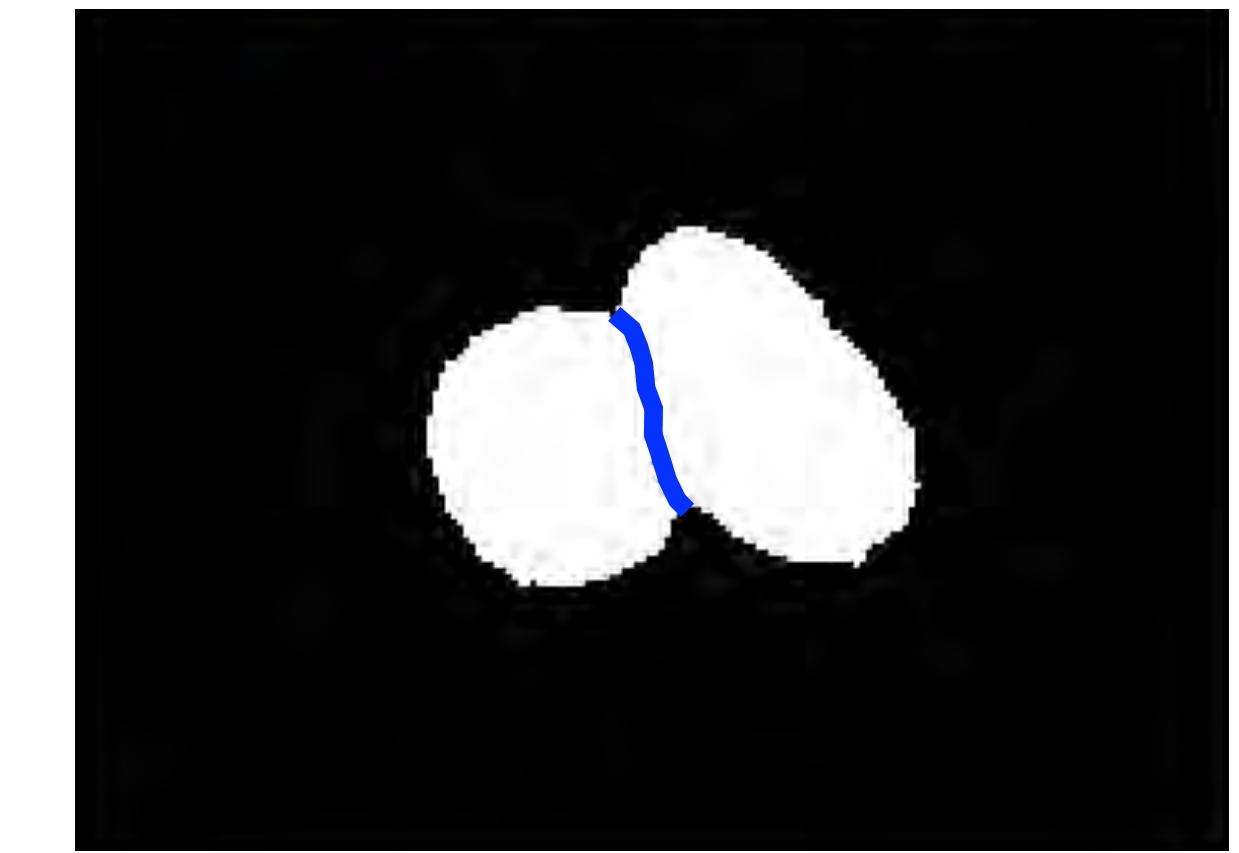
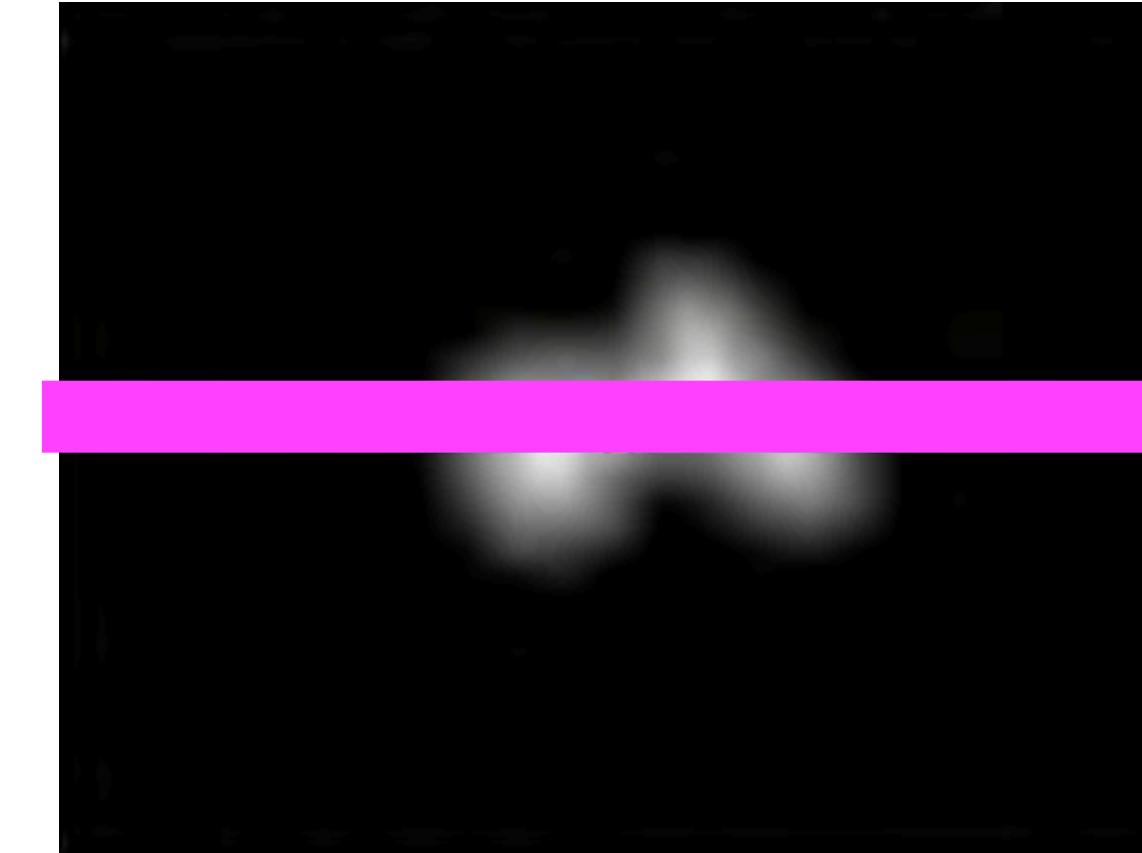


# Solution: Watershed

Binary mask

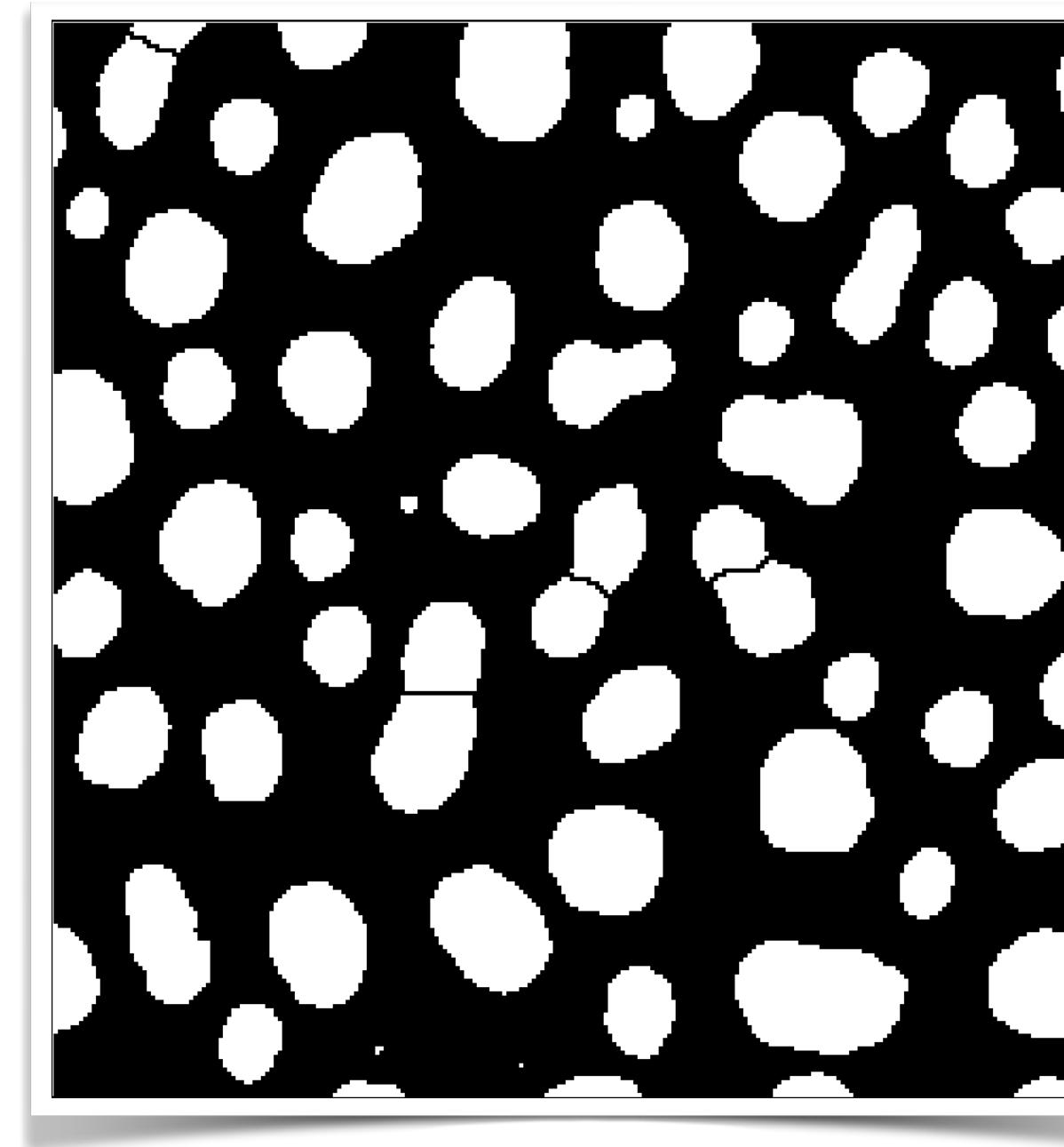


Distance transform

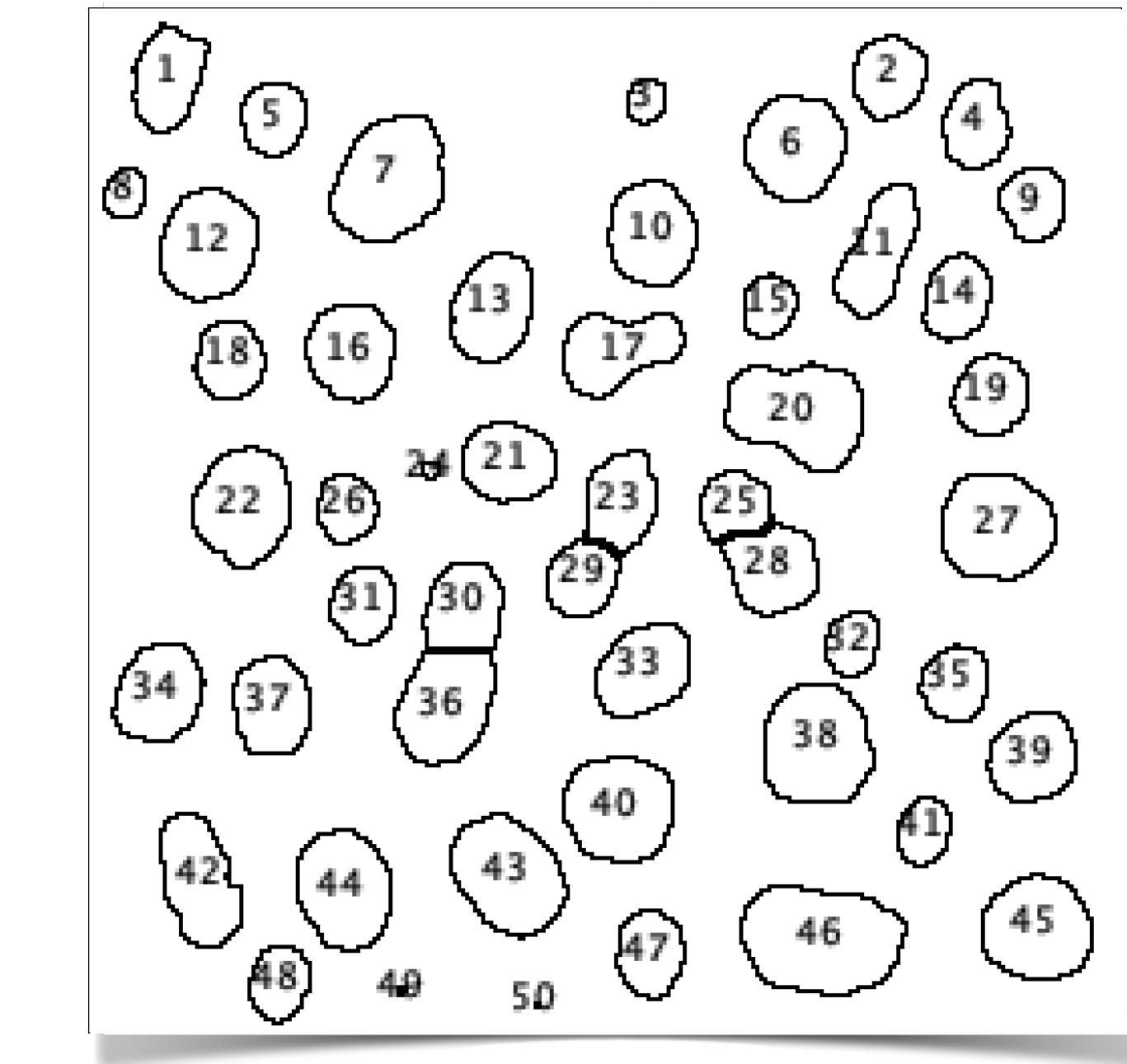


# From binary image to instance segmentation

Process and segment



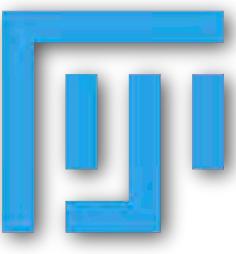
Binary mask



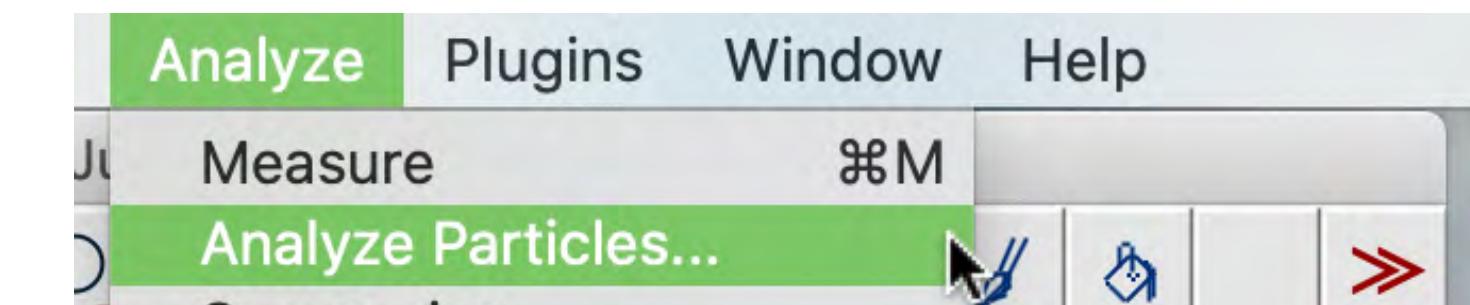
Instance segmentation

# Analyze particles

Process and segment

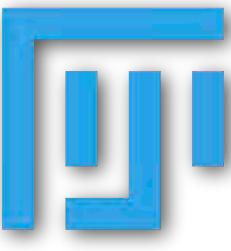


in **Fiji**: **Analyze > Analyze Particles...**

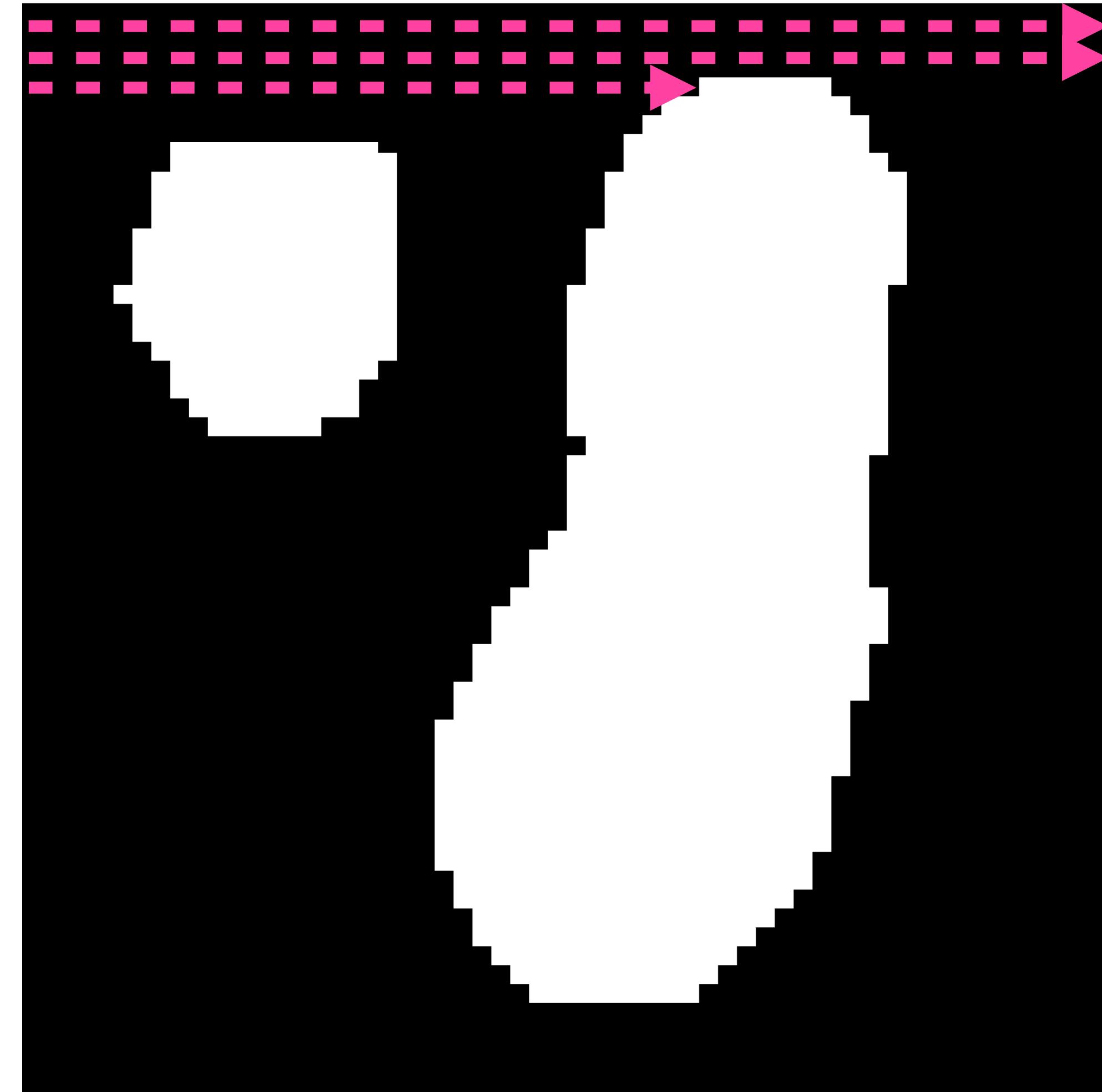


# Analyze particles — underlying principles

Process and segment

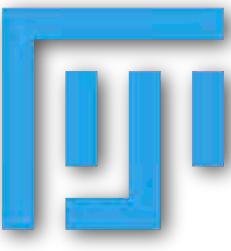


in Fiji:  
**Analyze**  
  > **Analyze Particles...**

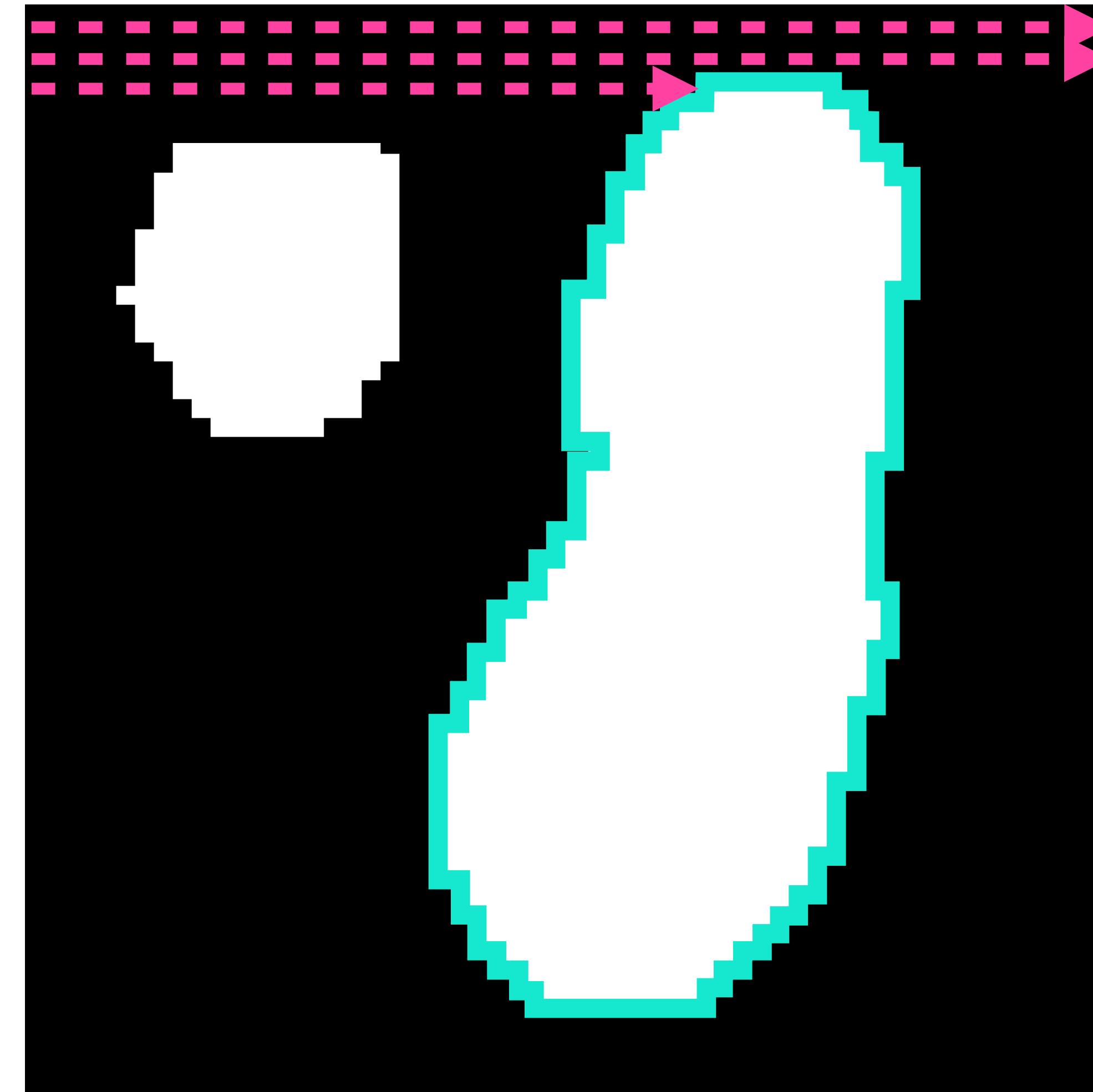


# Analyze particles — underlying principles

Process and segment

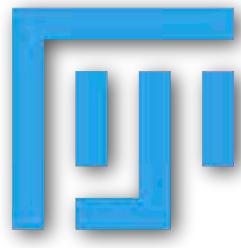


in Fiji:  
**Analyze**  
-> **Analyze Particles...**

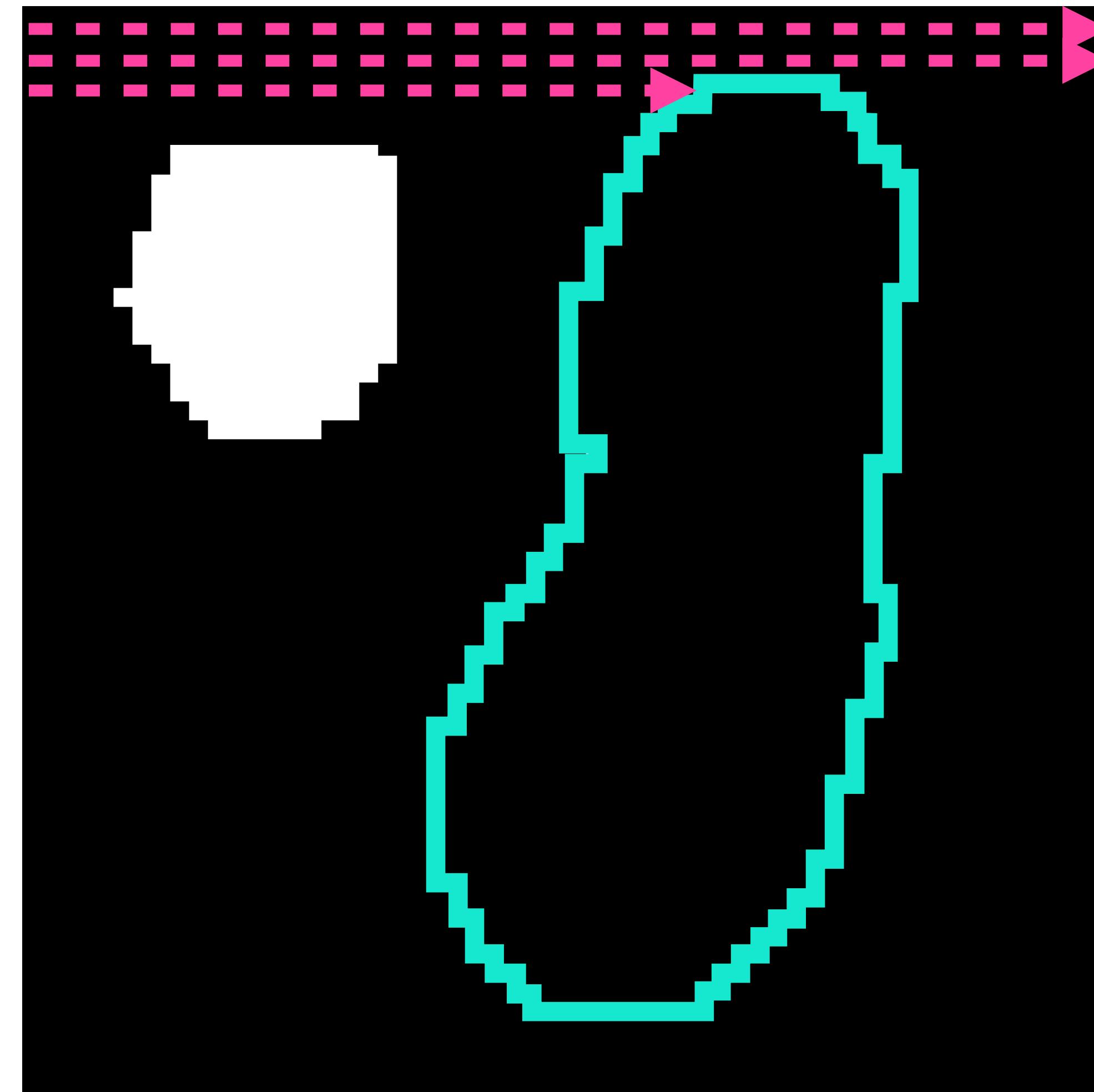


# Analyze particles — underlying principles

Process and segment



in Fiji:  
**Analyze**  
-> **Analyze Particles...**

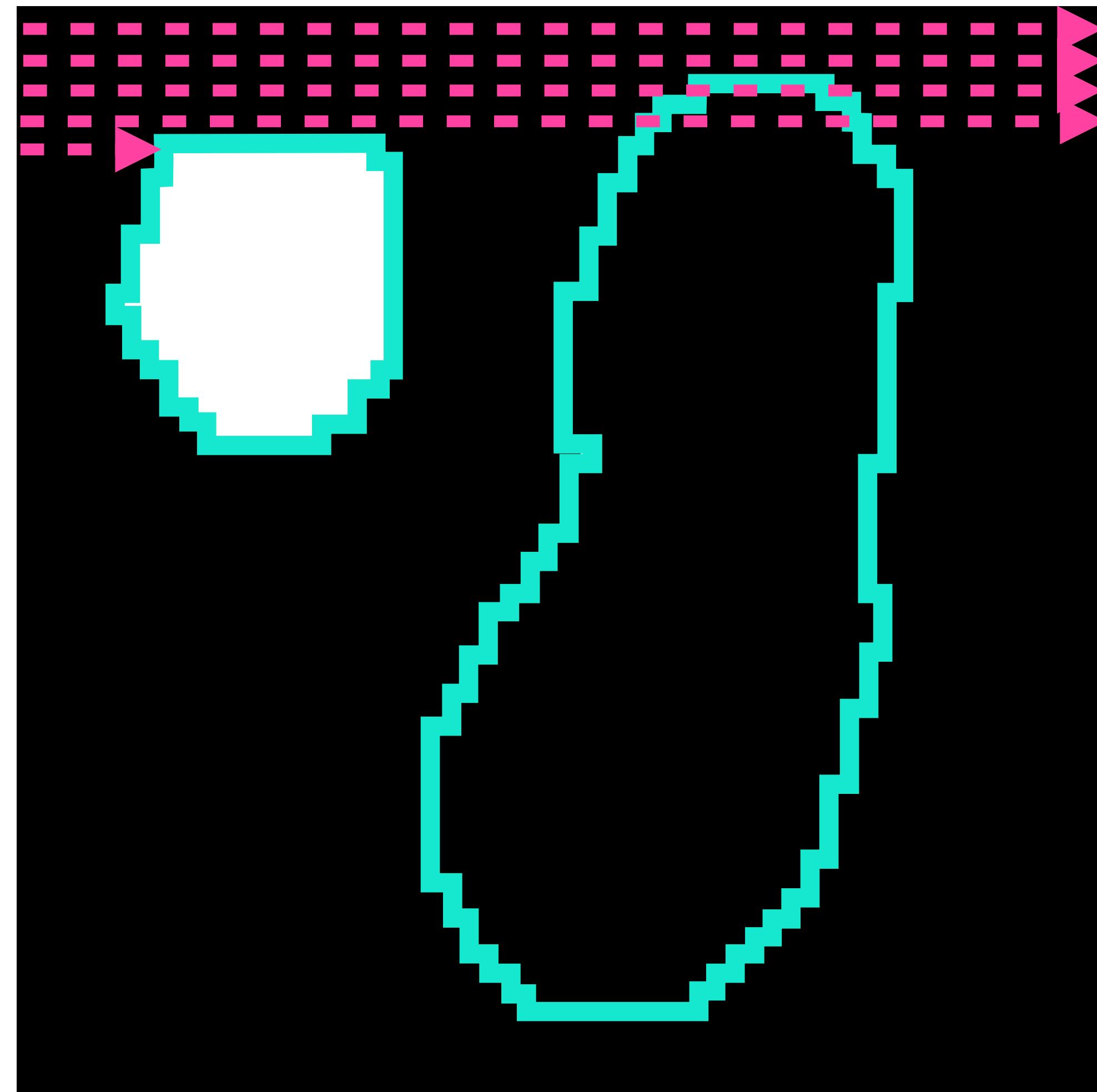


# Analyze particles — underlying principles

Process and segment



in Fiji:  
**Analyze**  
-> **Analyze Particles...**

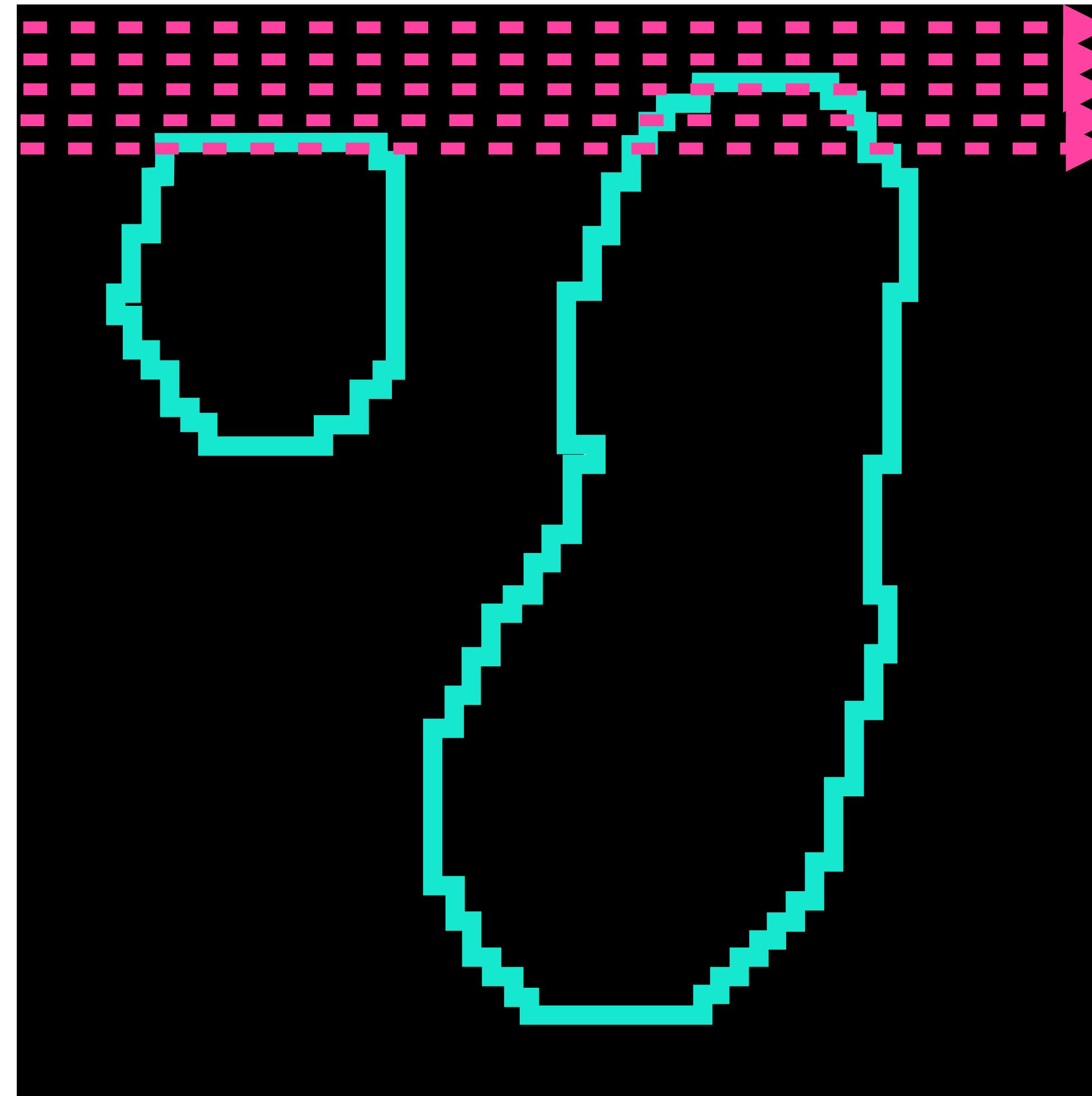


# Analyze particles — underlying principles

Process and segment

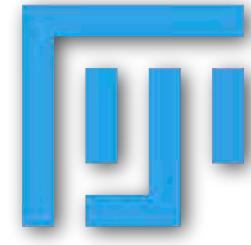


in Fiji:  
**Analyze**  
-> **Analyze Particles...**

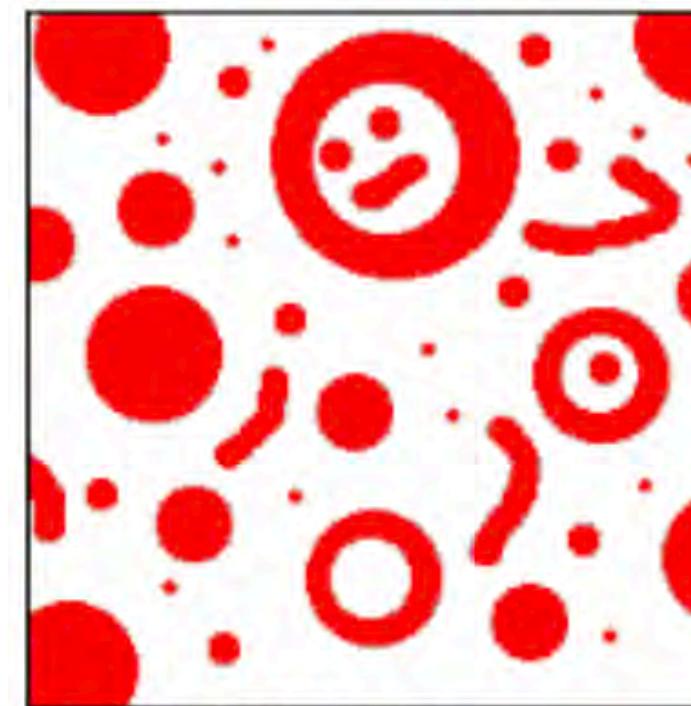


# Analyze particles: Select for morphology

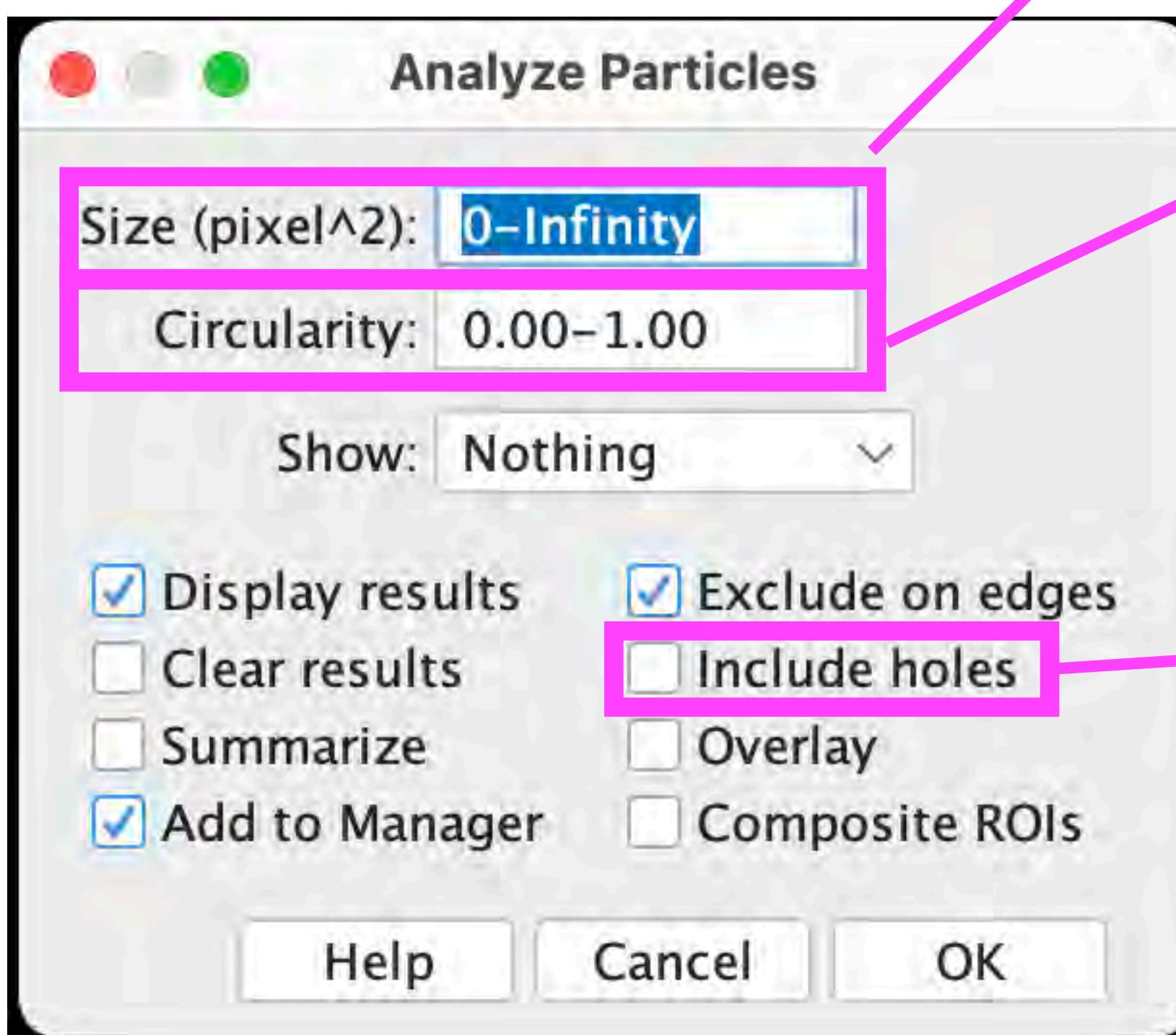
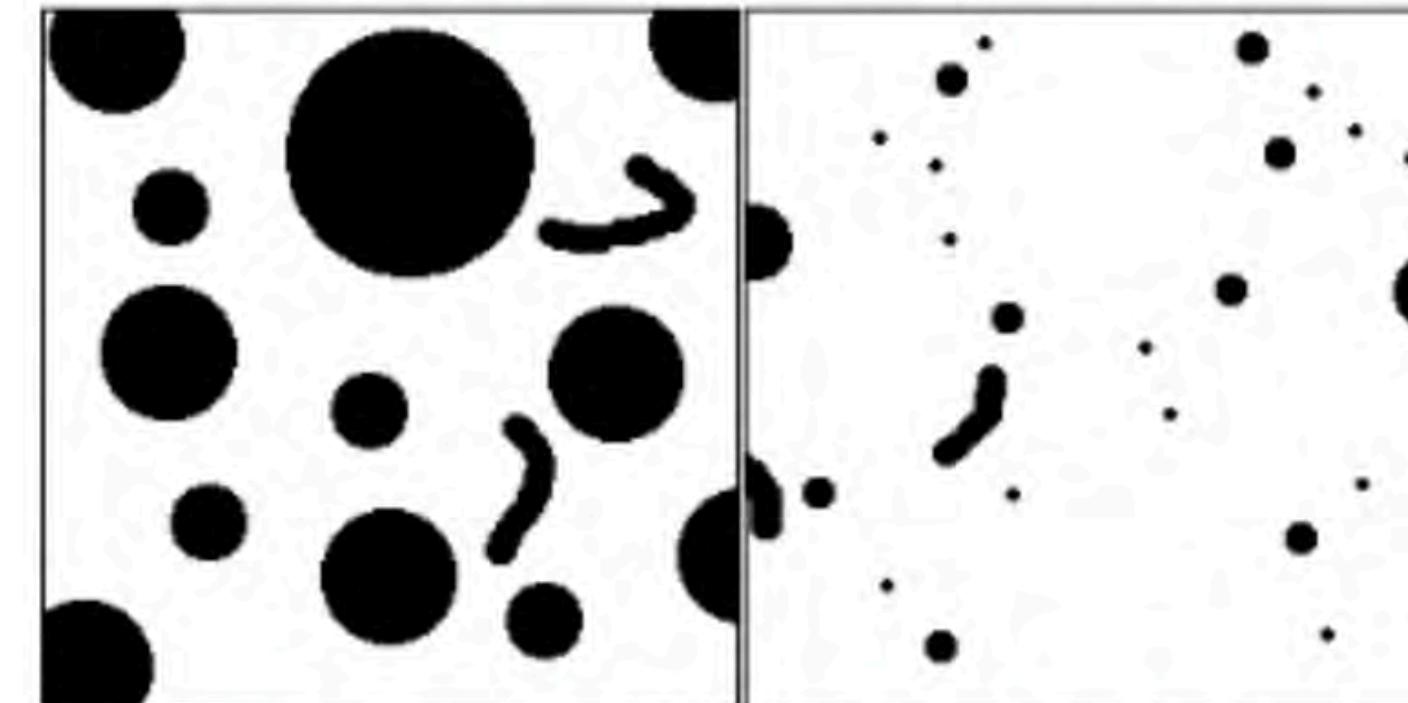
Process and segment



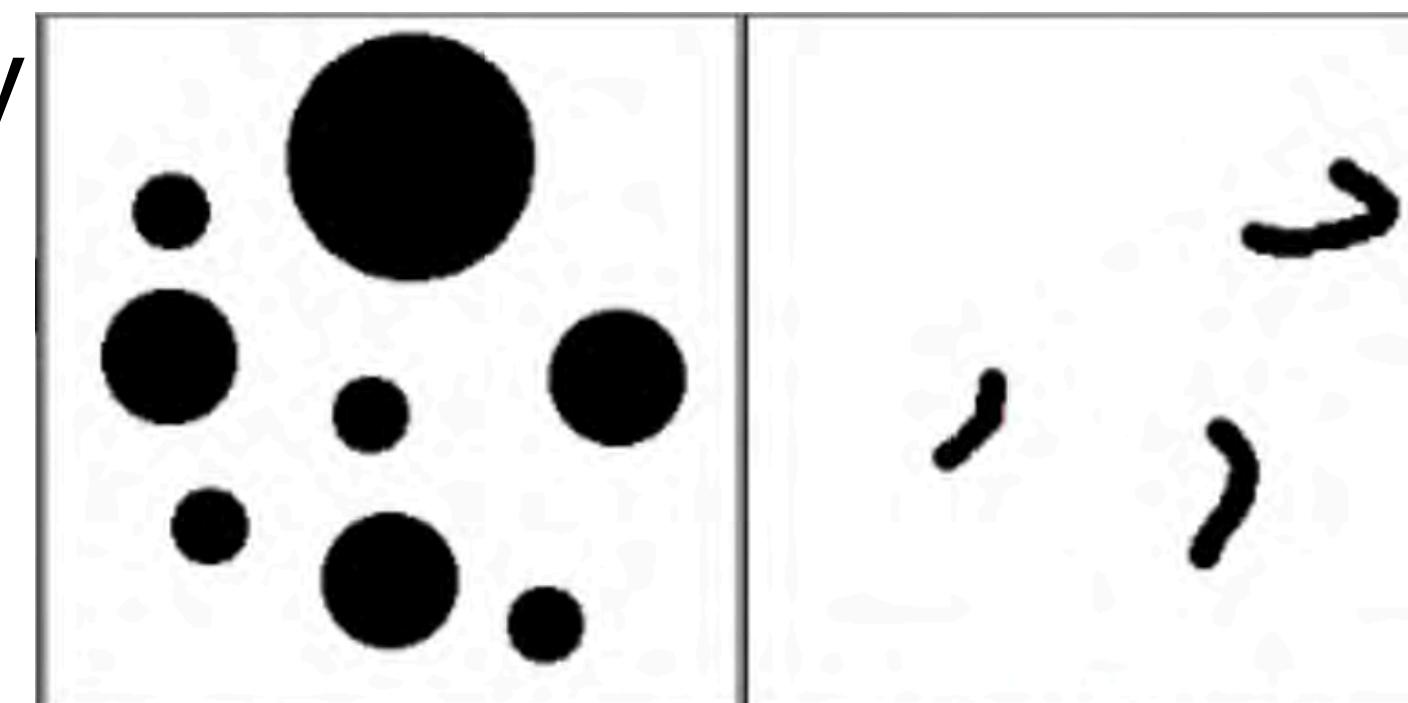
in Fiji:  
Analyze  
> Analyze Particles...



Size



Circularity

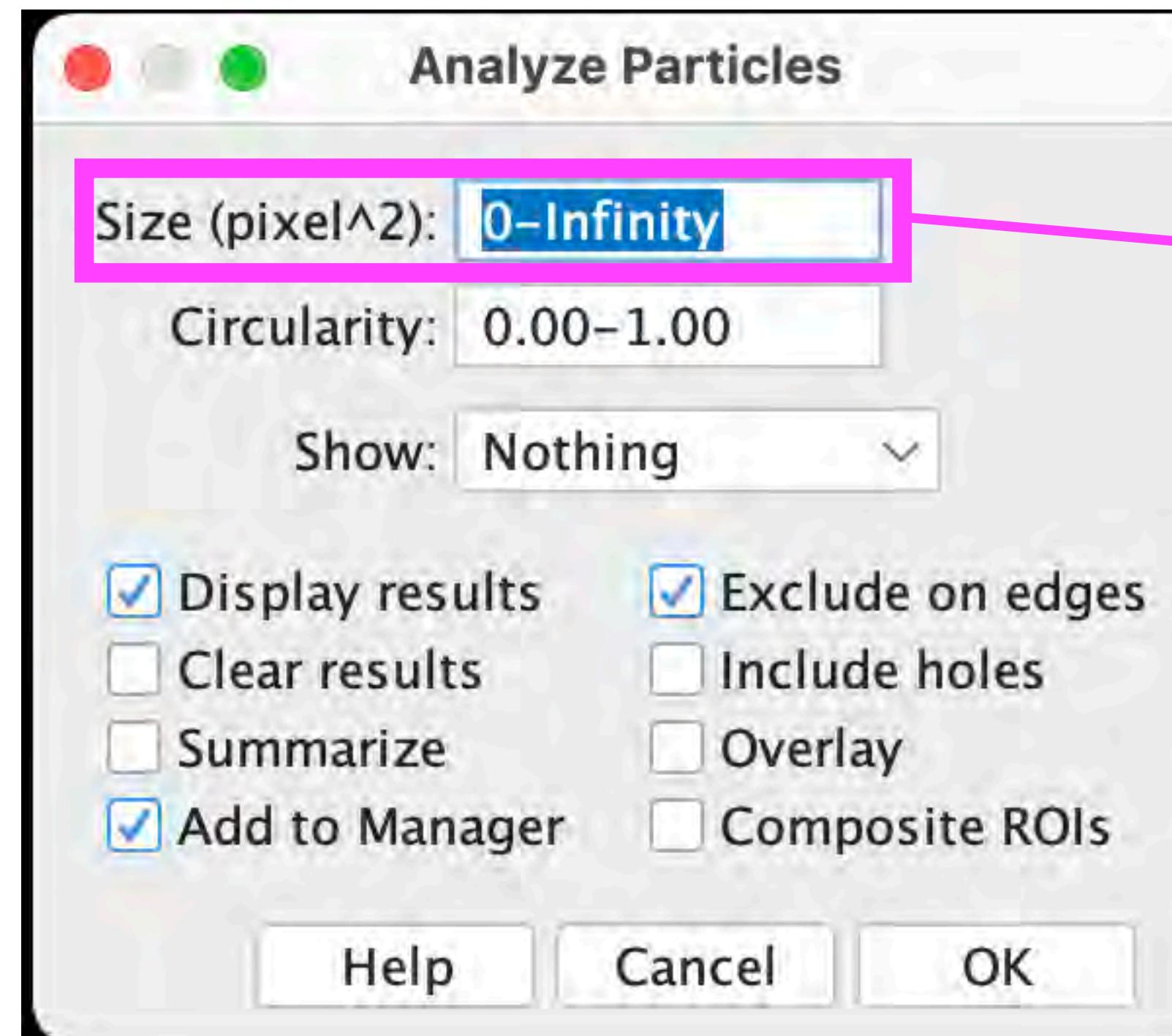


Fill

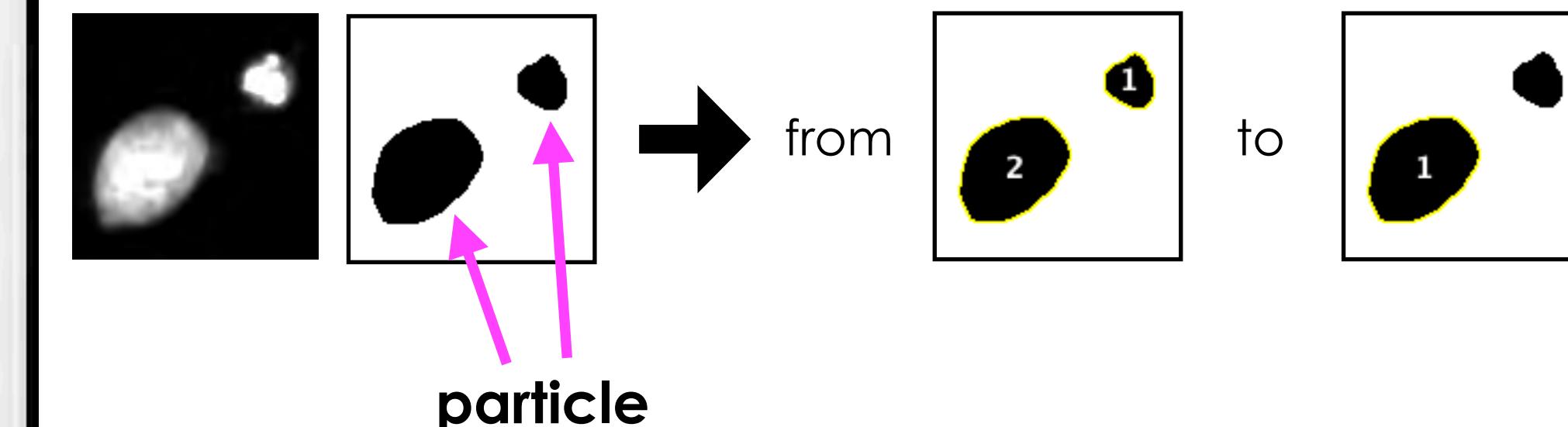


# Analyze particles: Size

in Fiji: Analyze > Analyze Particles...

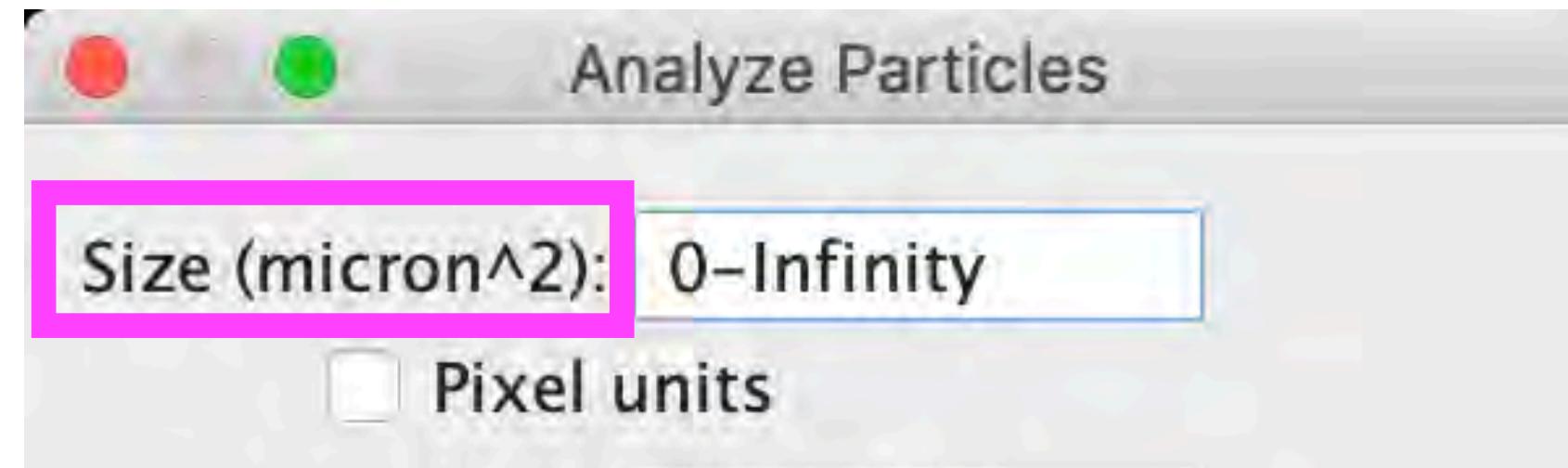


Size range of the particles that you want to detect.

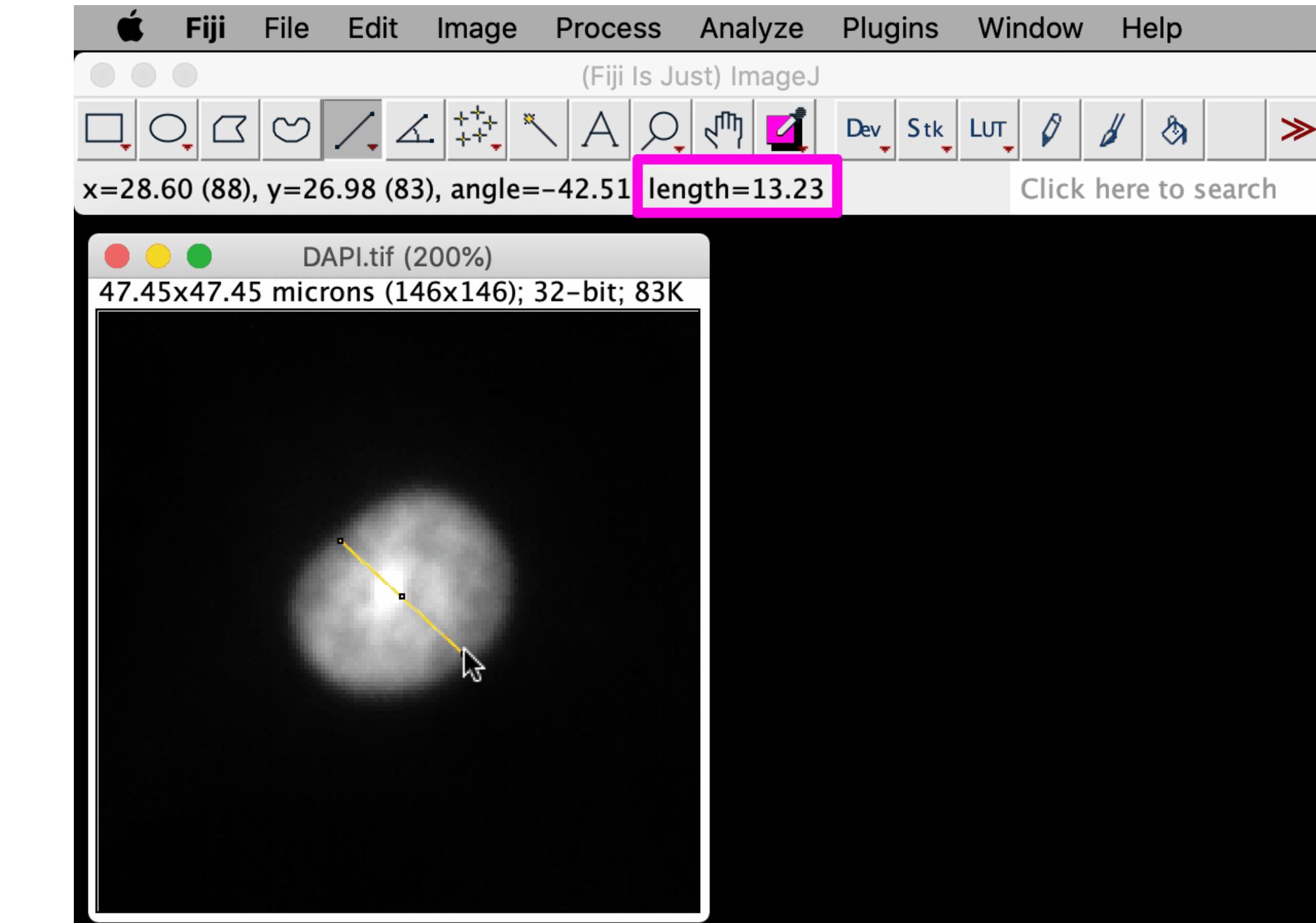


# Analyze particles: Size

Process and segment

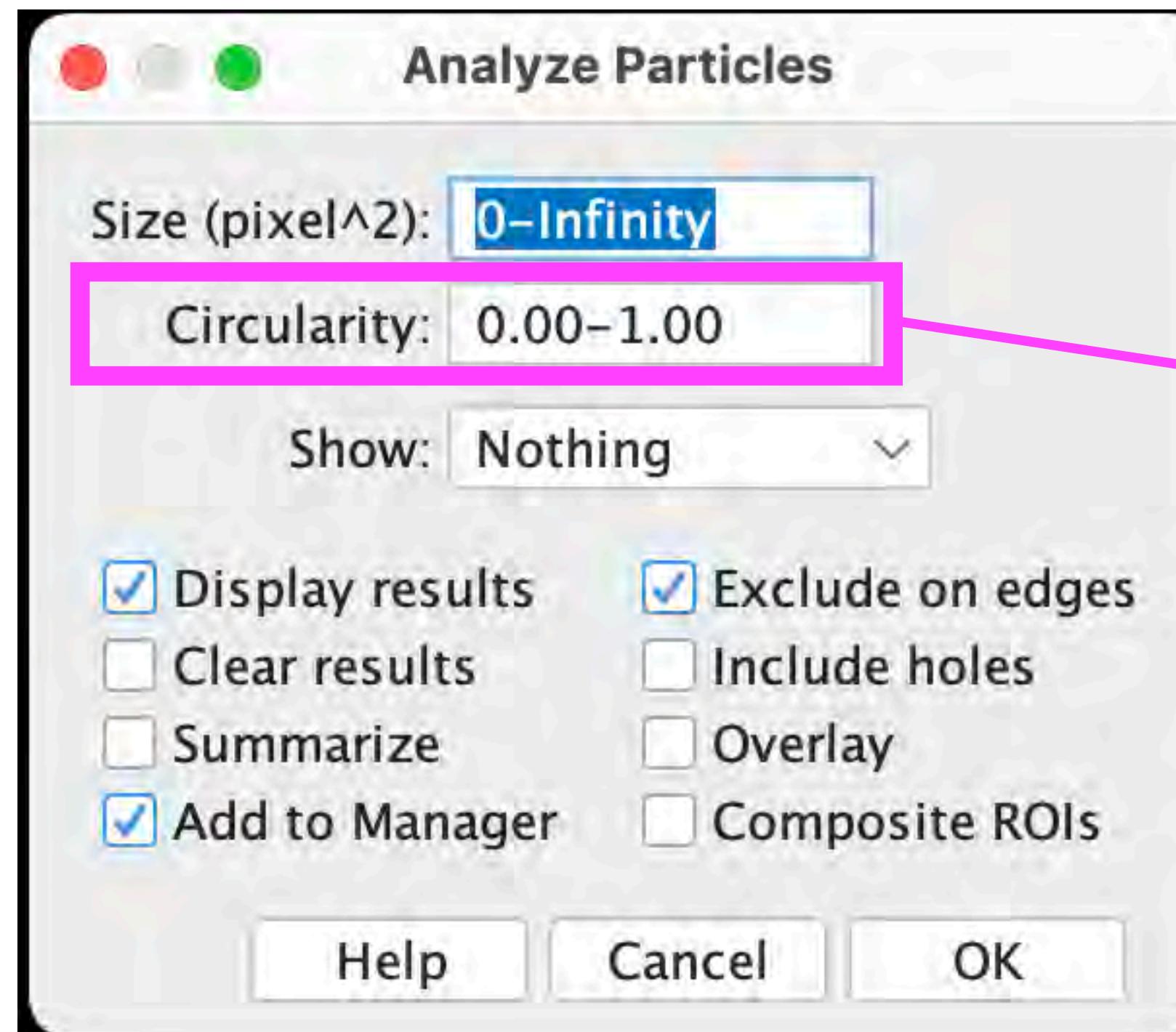


the **size range** of the **particles** that you want to detect.



# Analyze particles: Circularity

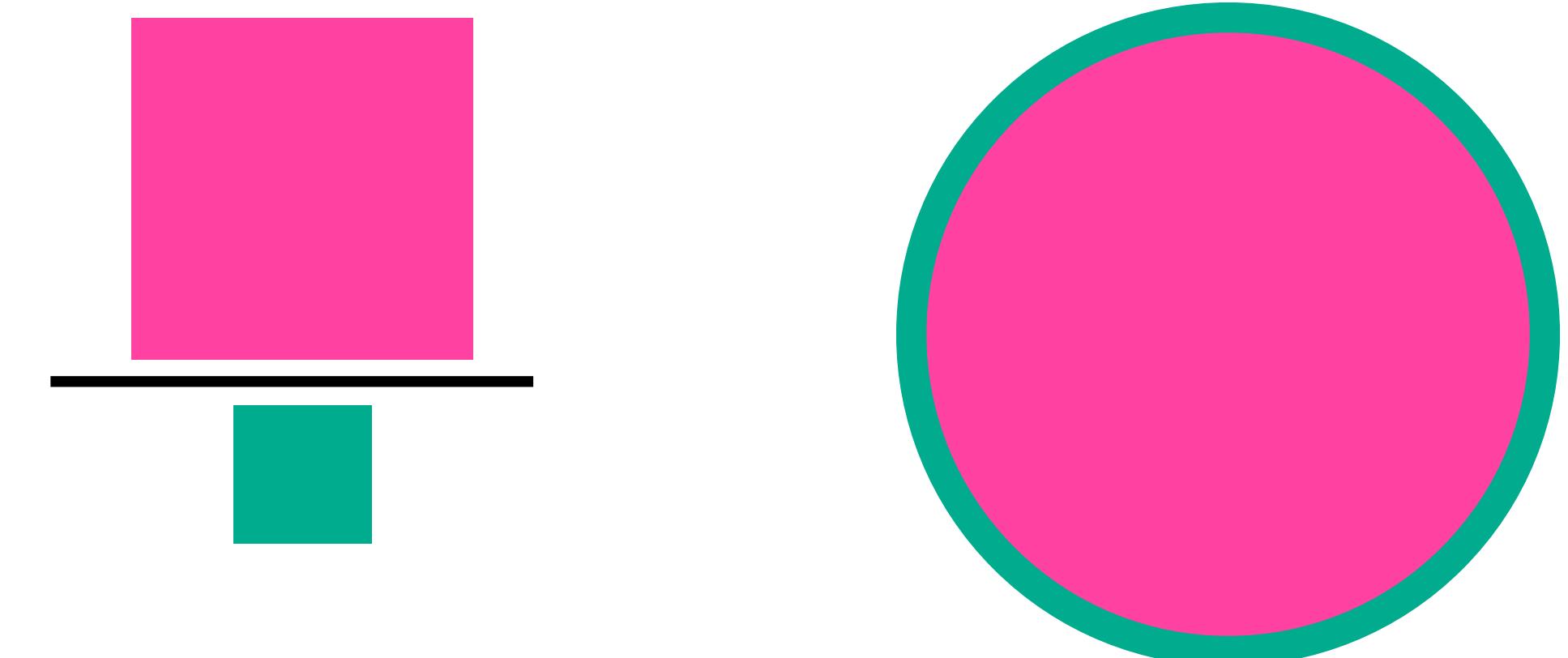
in Fiji: Analyze > Analyze Particles...



Circularity of the **particles** that you want to detect.

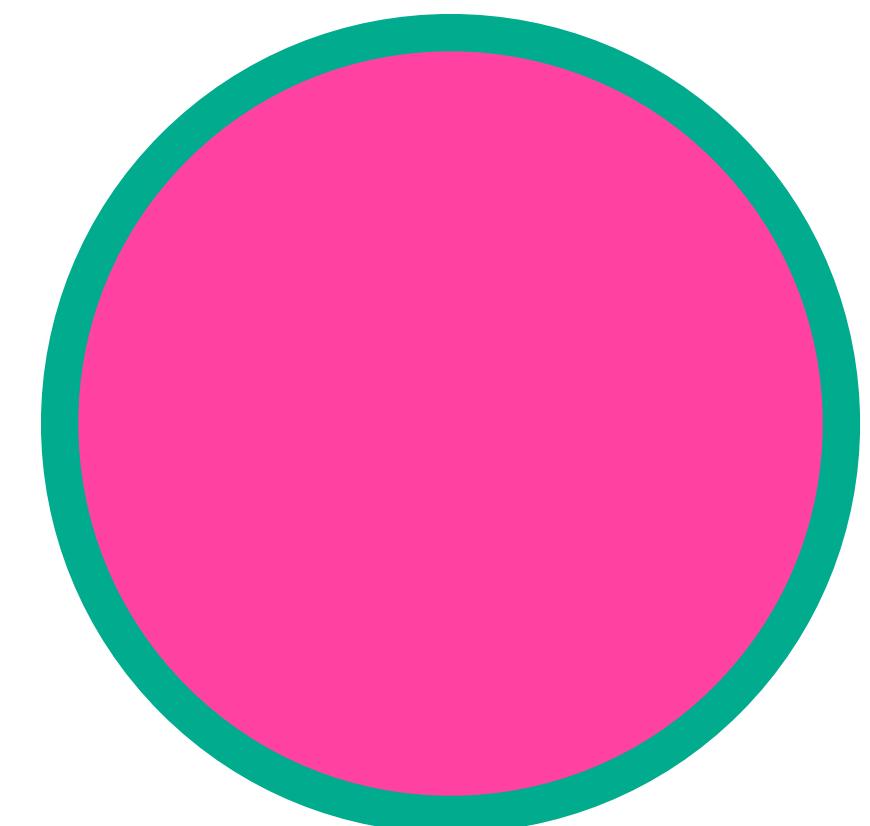
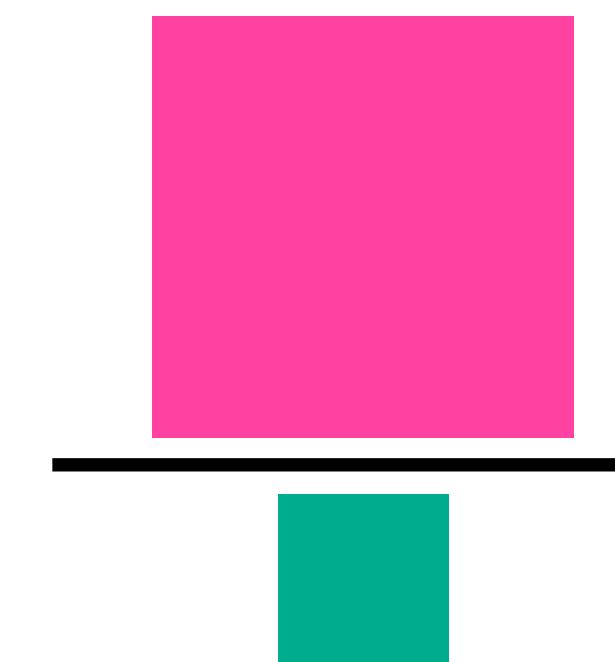
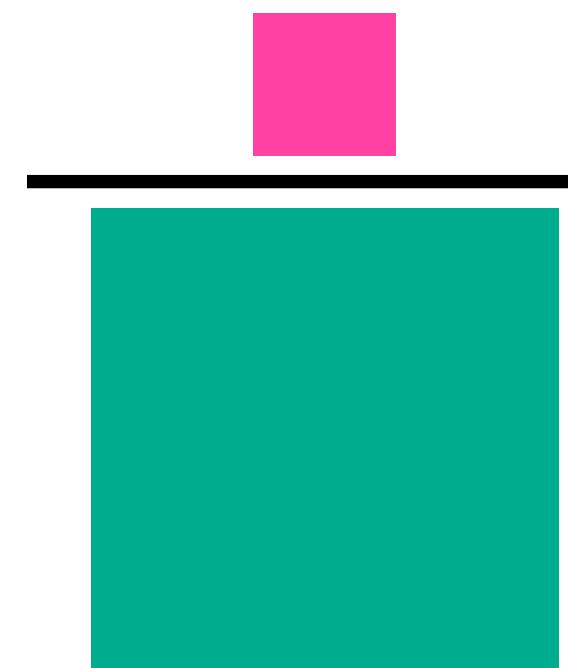
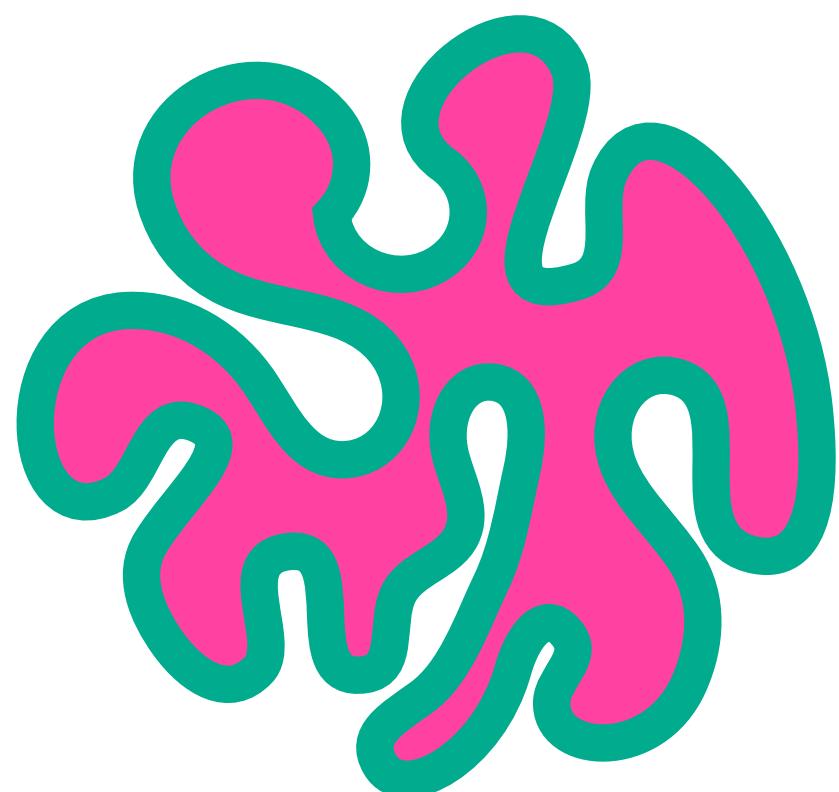
# Analyze particles: Circularity

$$4\pi \times \frac{\text{area}}{\text{perimeter}^2} = 1$$



# Analyze particles: Circularity

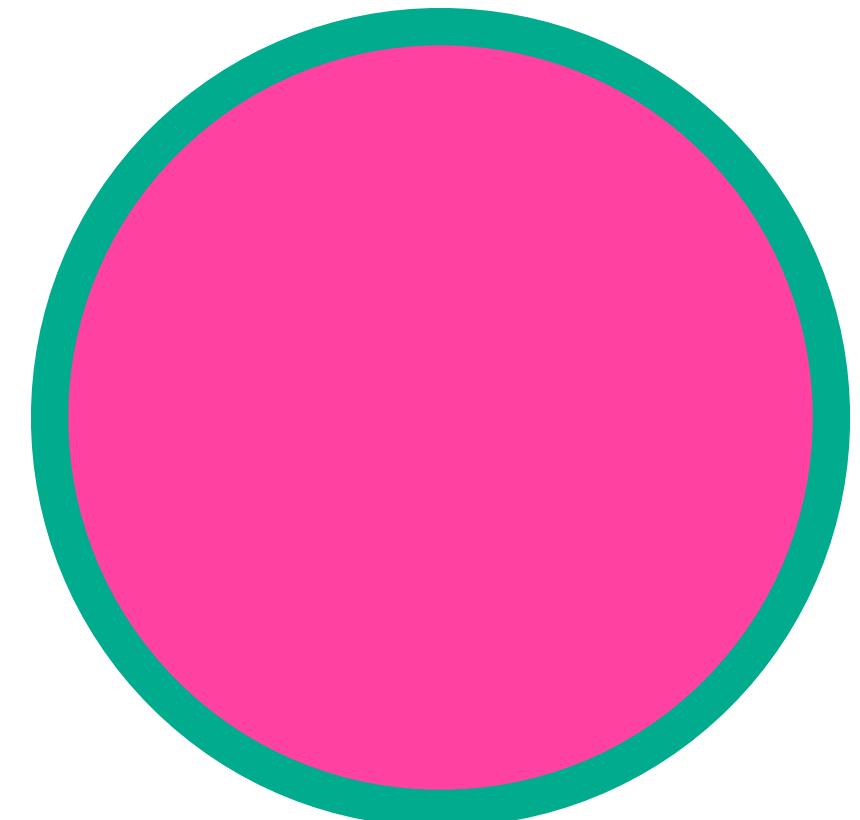
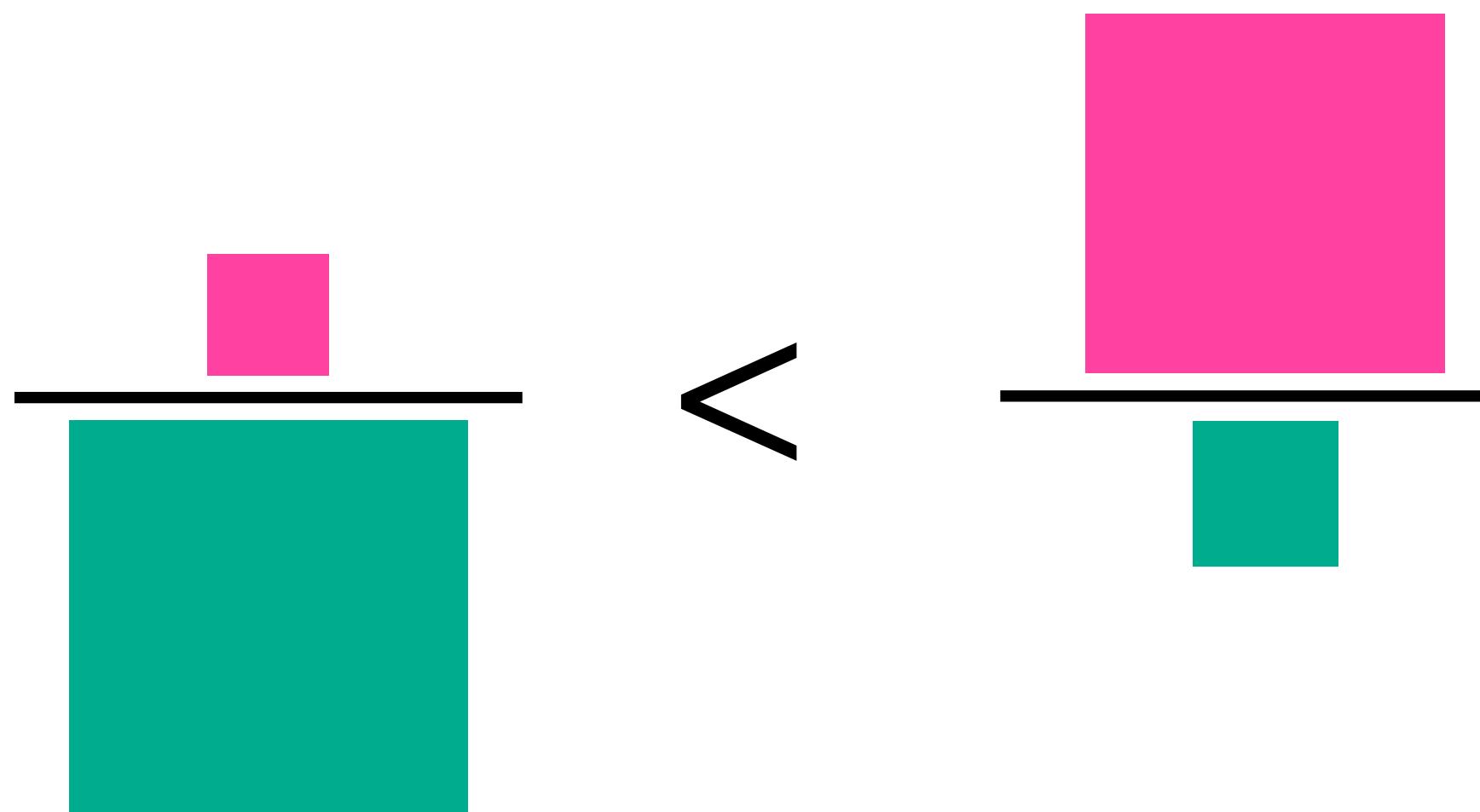
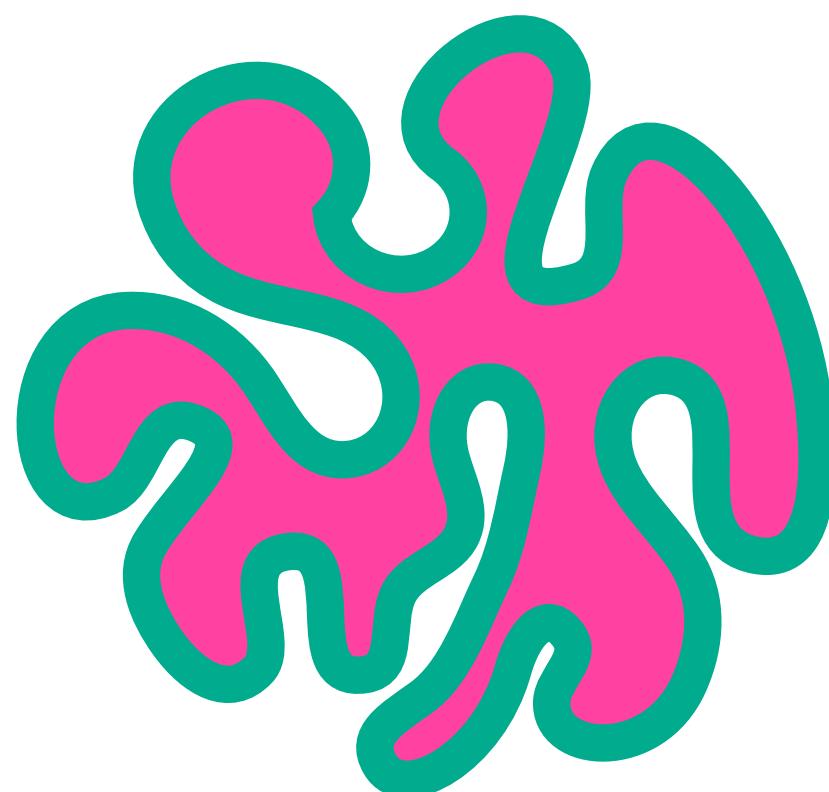
$$4\pi \times \frac{\text{area}}{\text{perimeter}^2} = 1$$



# Analyze particles: Circularity

$$4\pi \times \frac{\text{area}}{\text{perimeter}^2} < 1$$

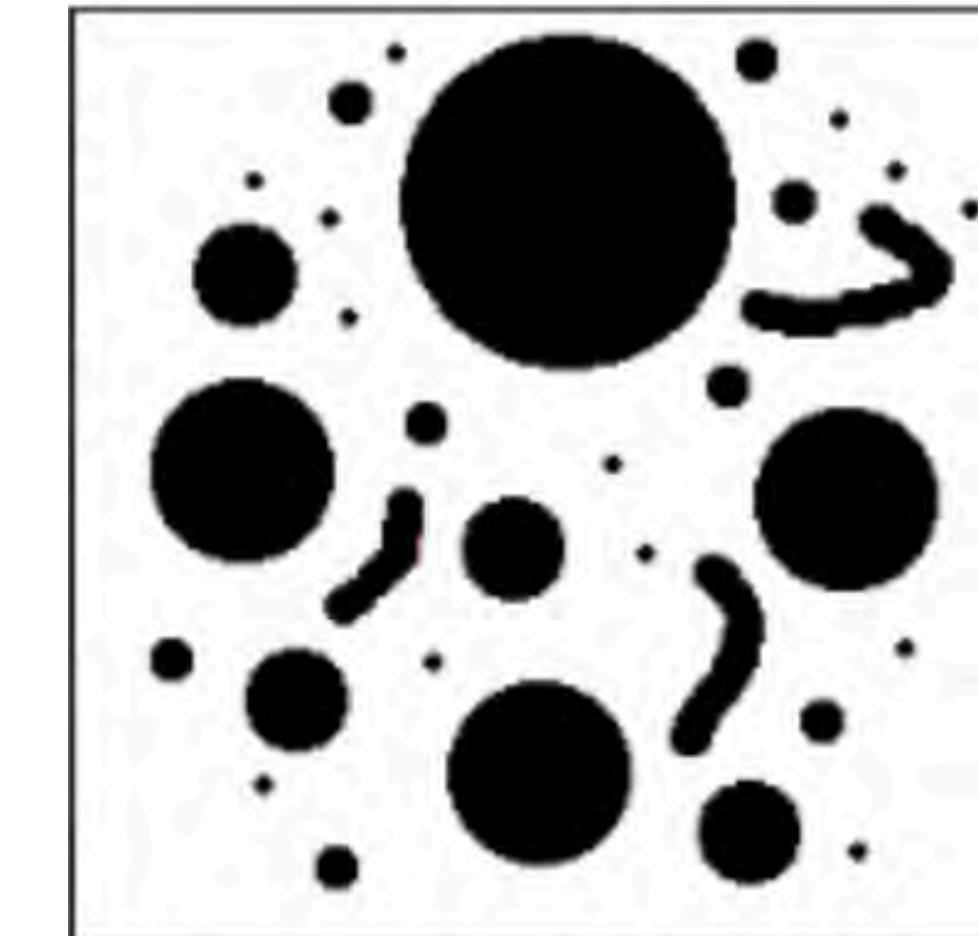
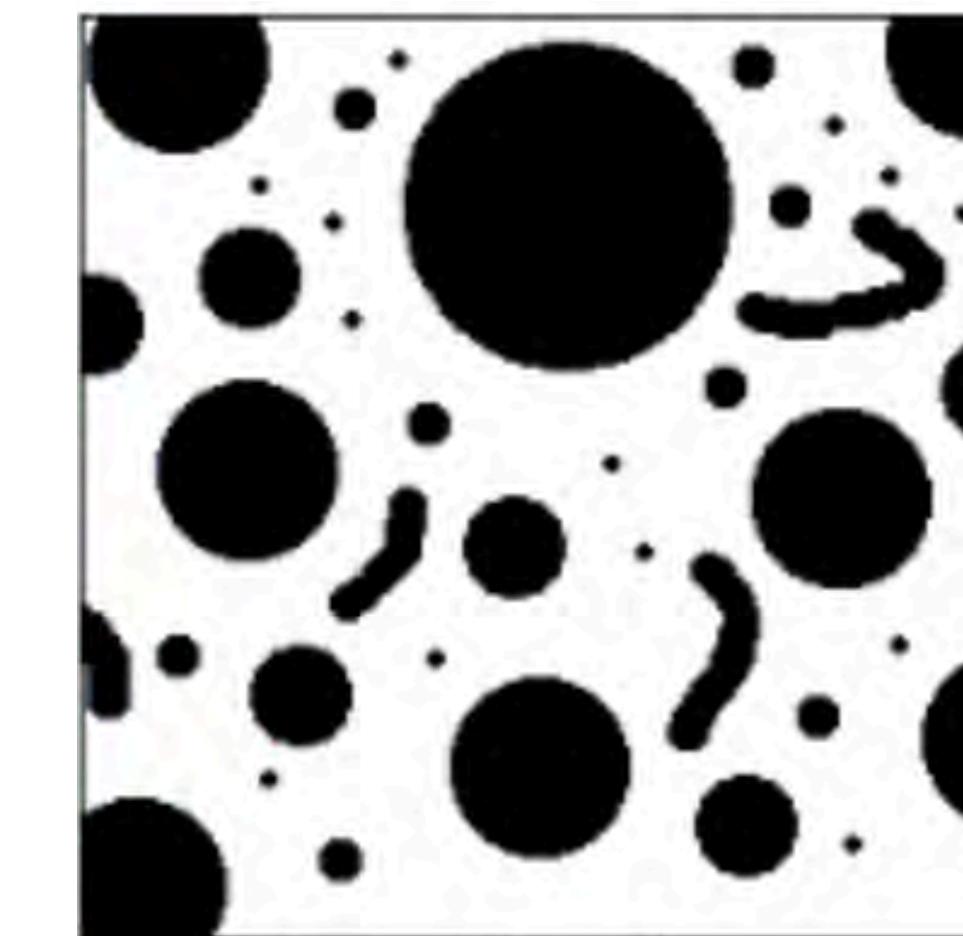
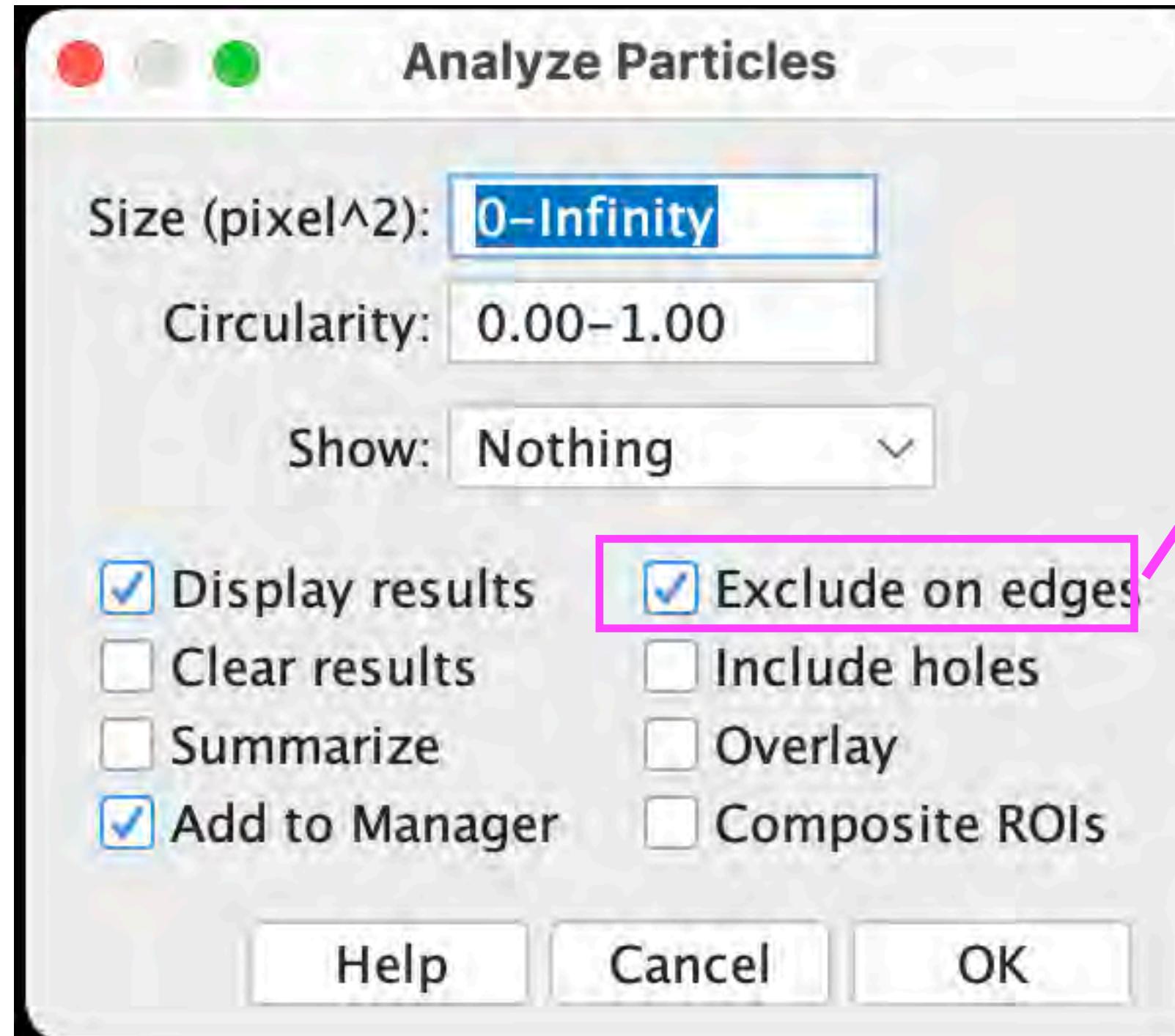
$$4\pi \times \frac{\text{area}}{\text{perimeter}^2} = 1$$



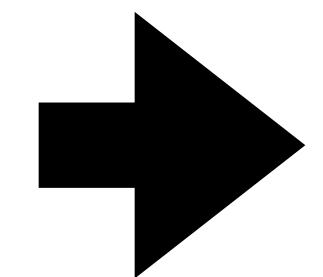
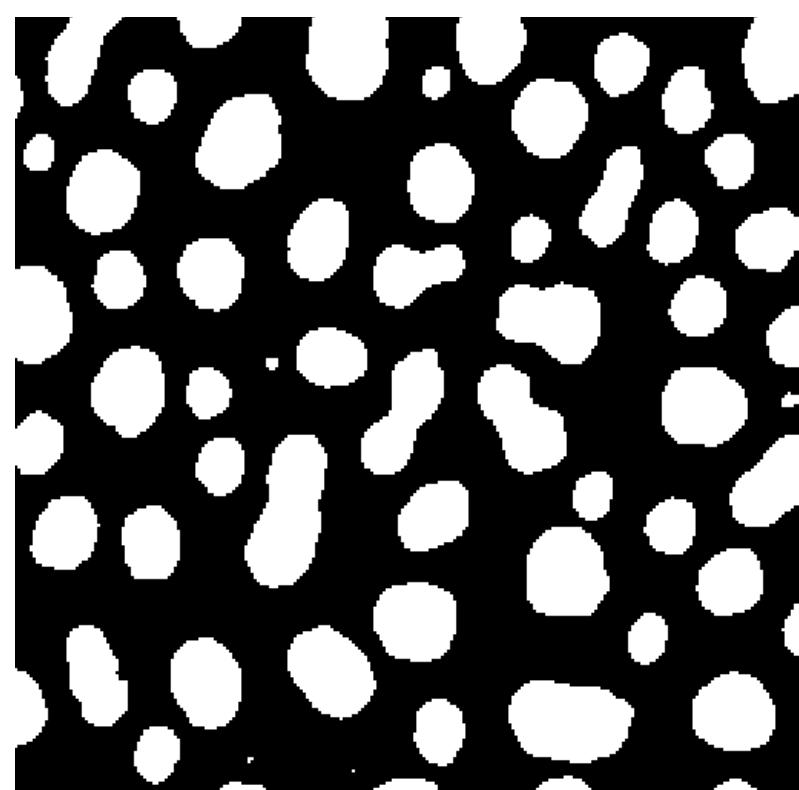
# Analyze particles: Exclude on Edges

in Fiji: Analyze > Analyze Particles...

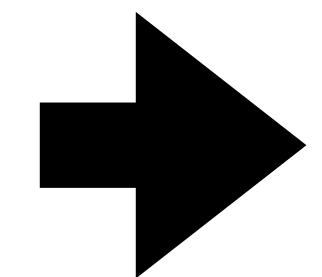
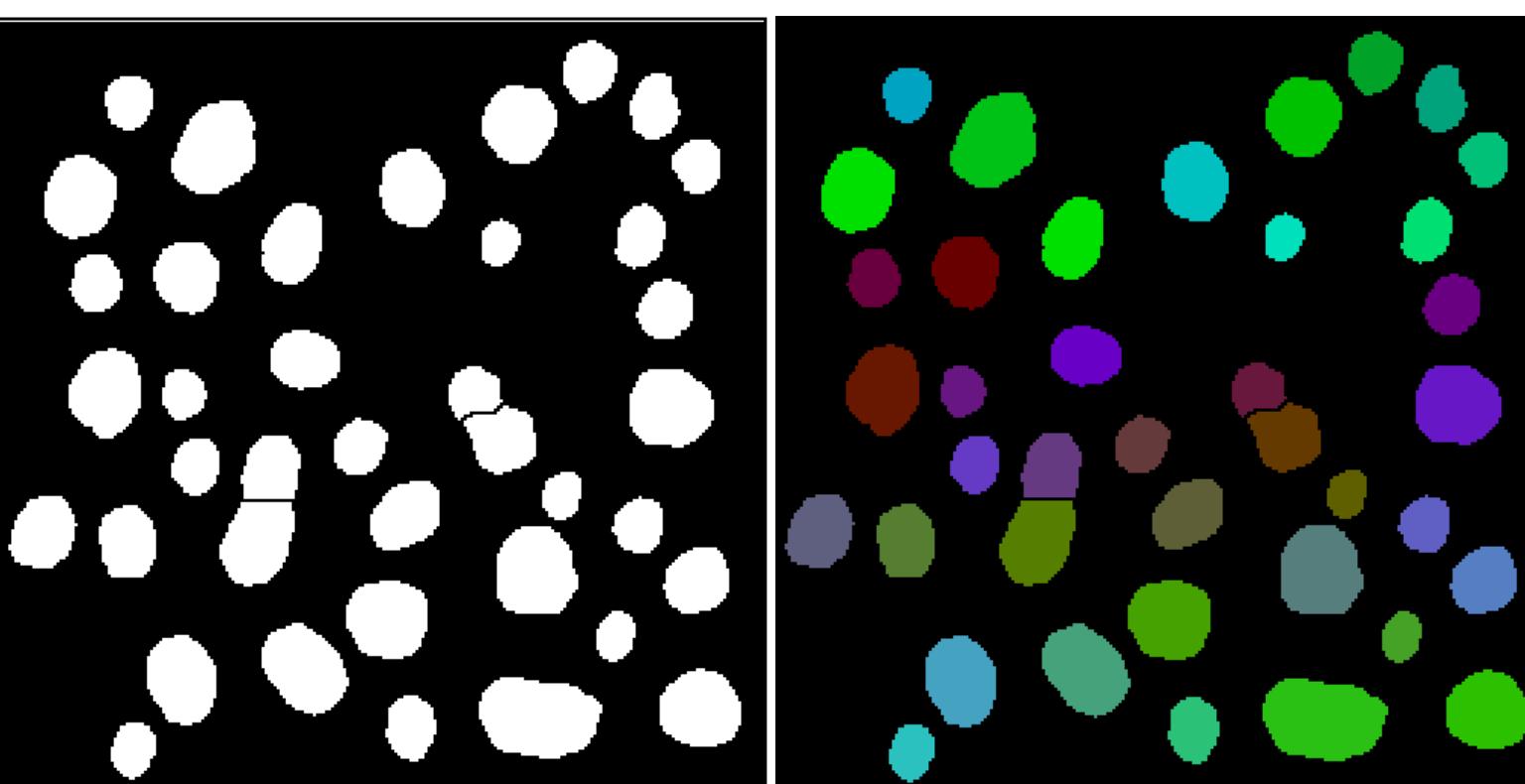
**"Exclude on edges"** excludes objects that are touching the borders of the image.



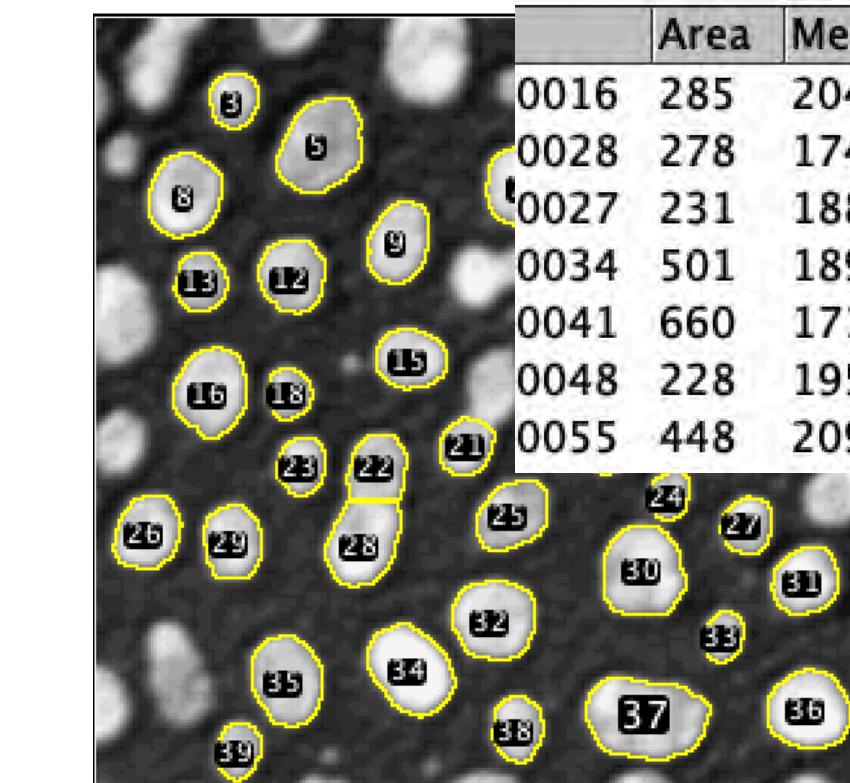
Binary mask



Process and segment



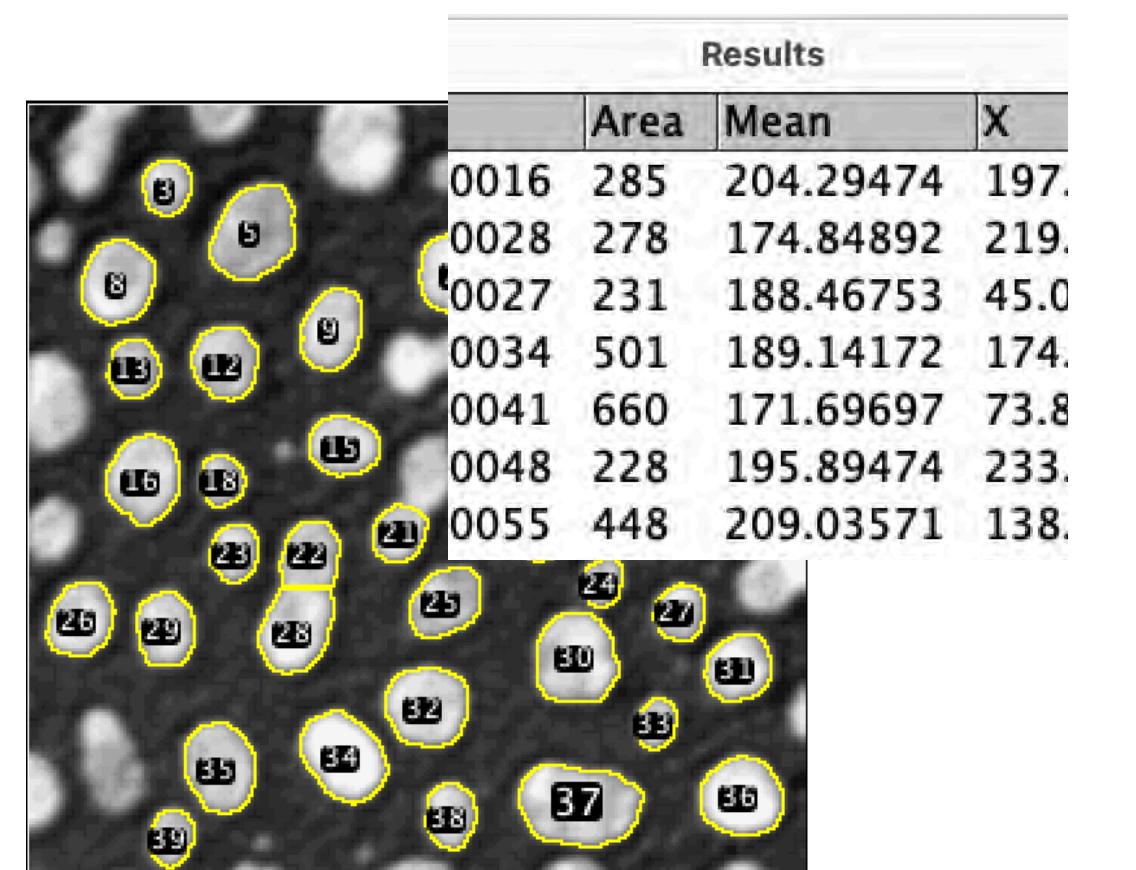
Export and Measure



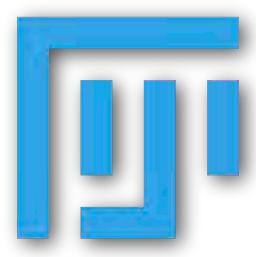
Results			
	Area	Mean	X
0016	285	204.29474	197.
0028	278	174.84892	219.
0027	231	188.46753	45.0
0034	501	189.14172	174.
0041	660	171.69697	73.8
0048	228	195.89474	233.
0055	448	209.03571	138.

- Save the segmented image
- Add to ROI manager
- Generate and export measurements

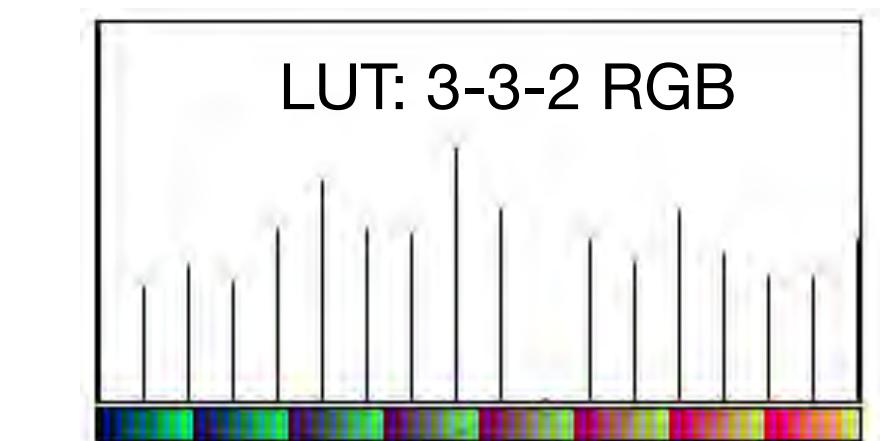
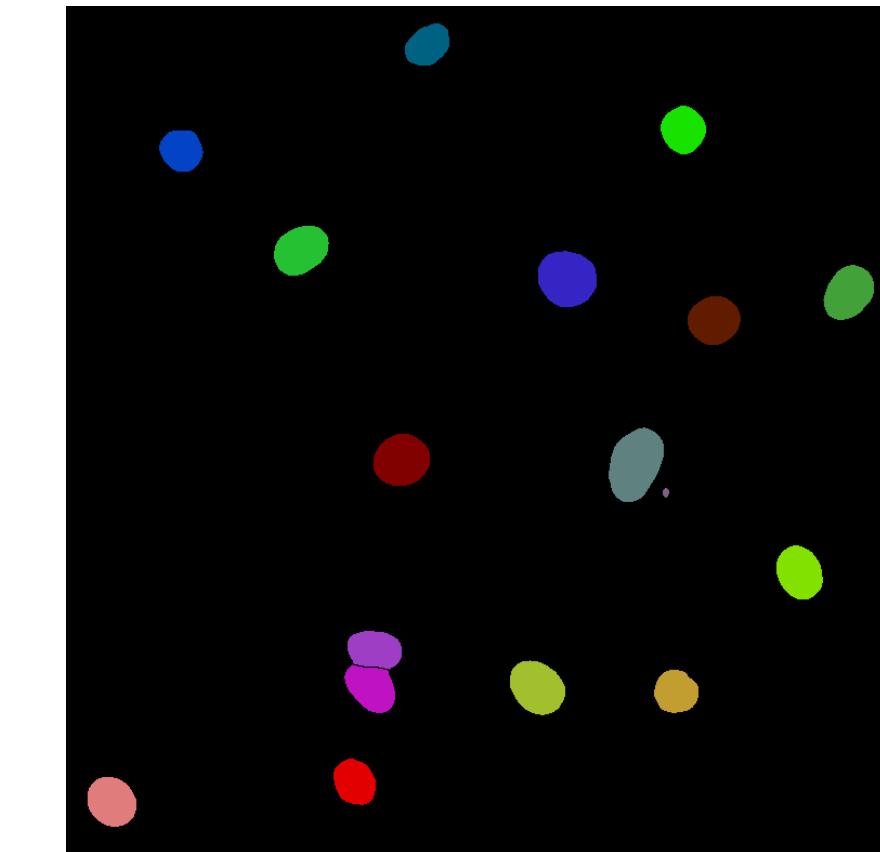
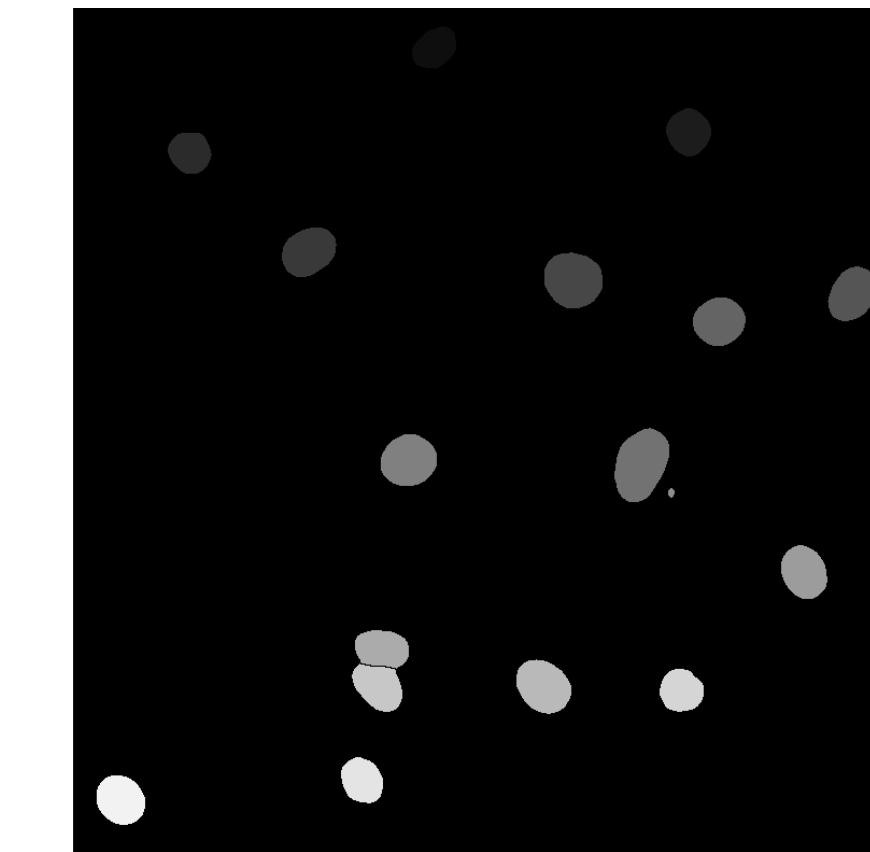
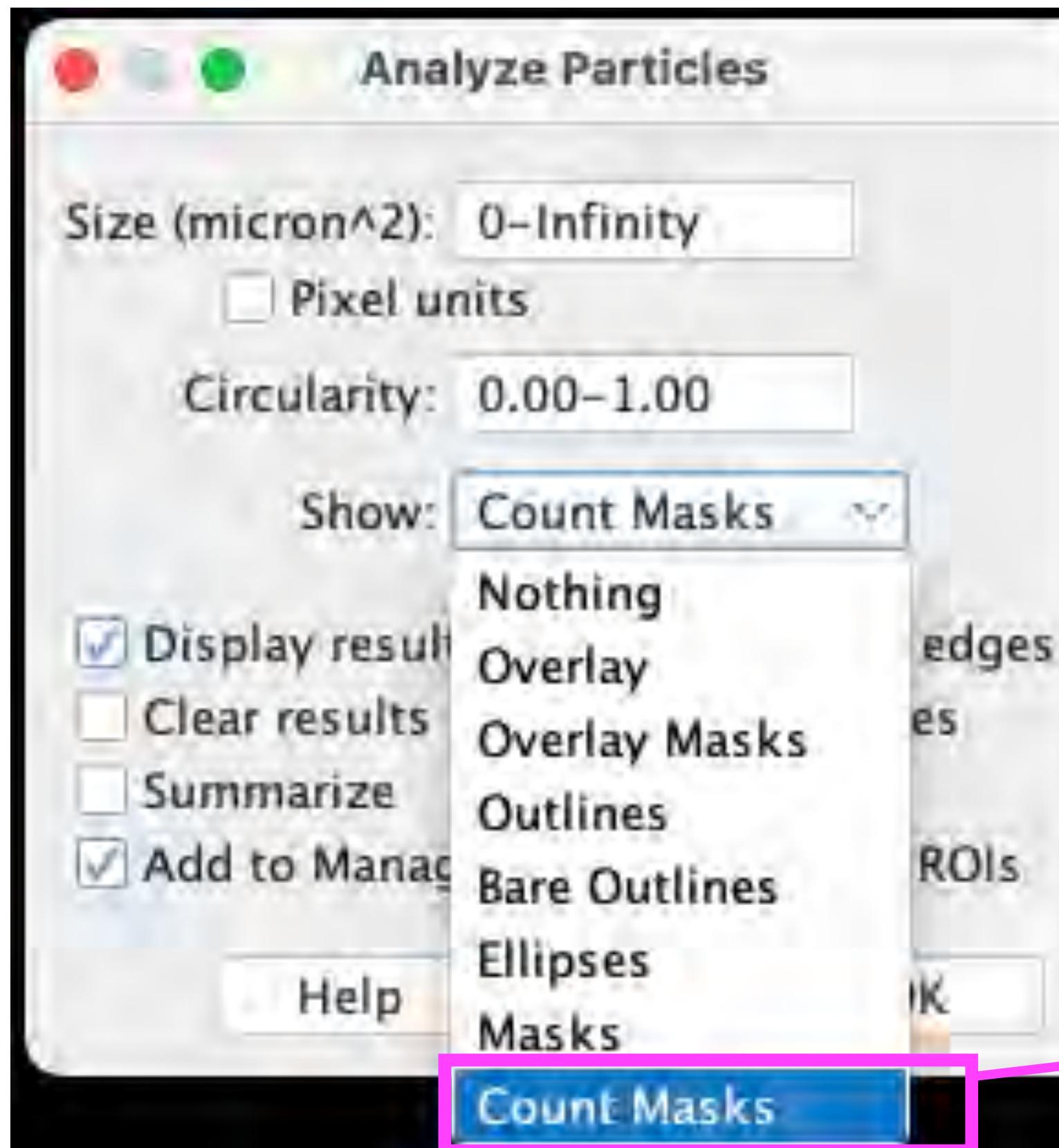
## Export and Measure



# Analyze particles: Save the segmented image

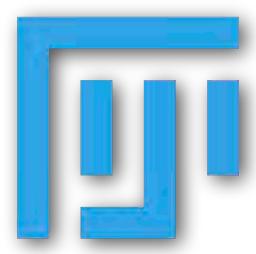
[Export](#)

in **Fiji**: **Analyze > Analyze Particles...**

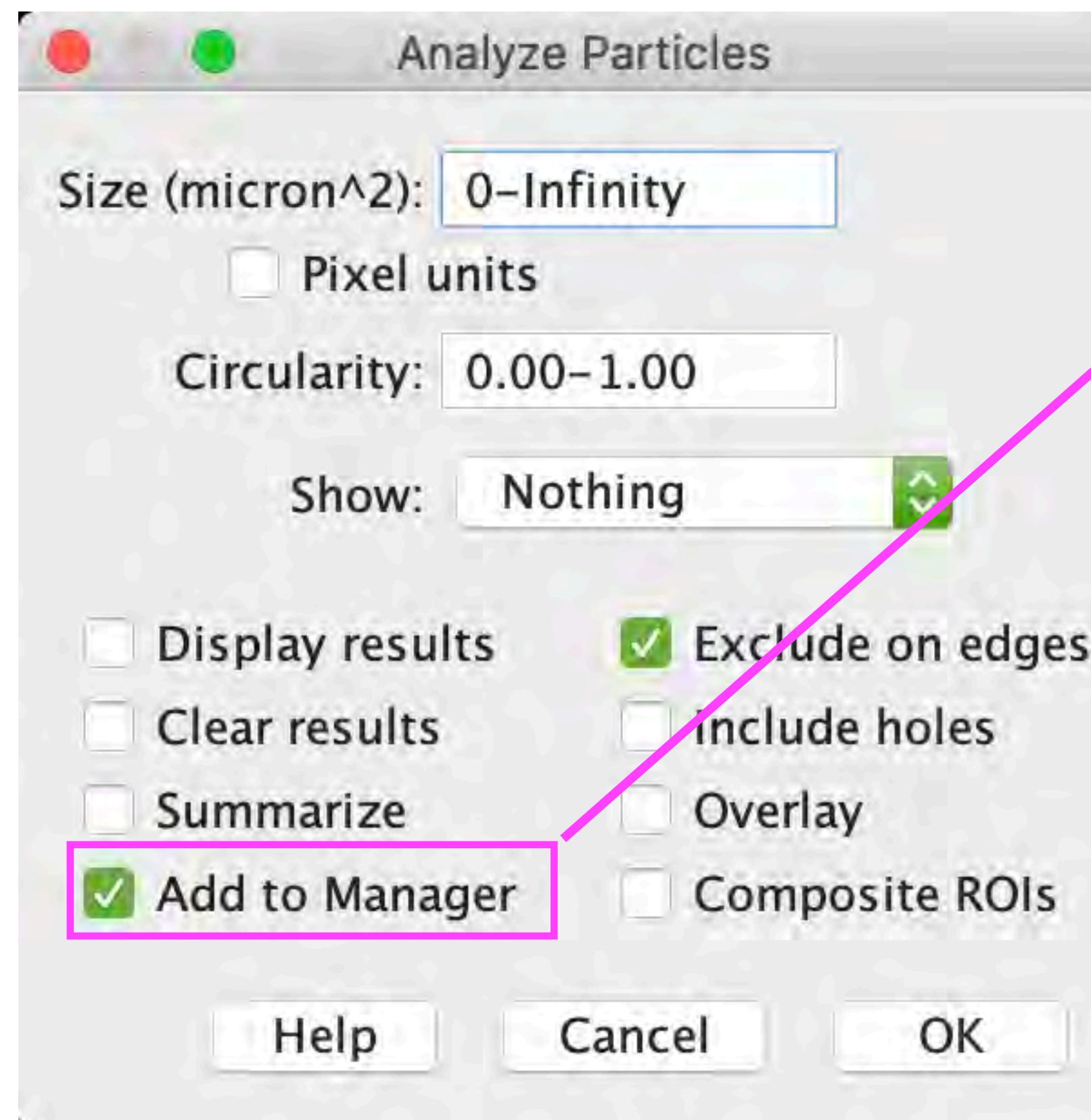


**"Count Masks"** encodes object identity as gray value.

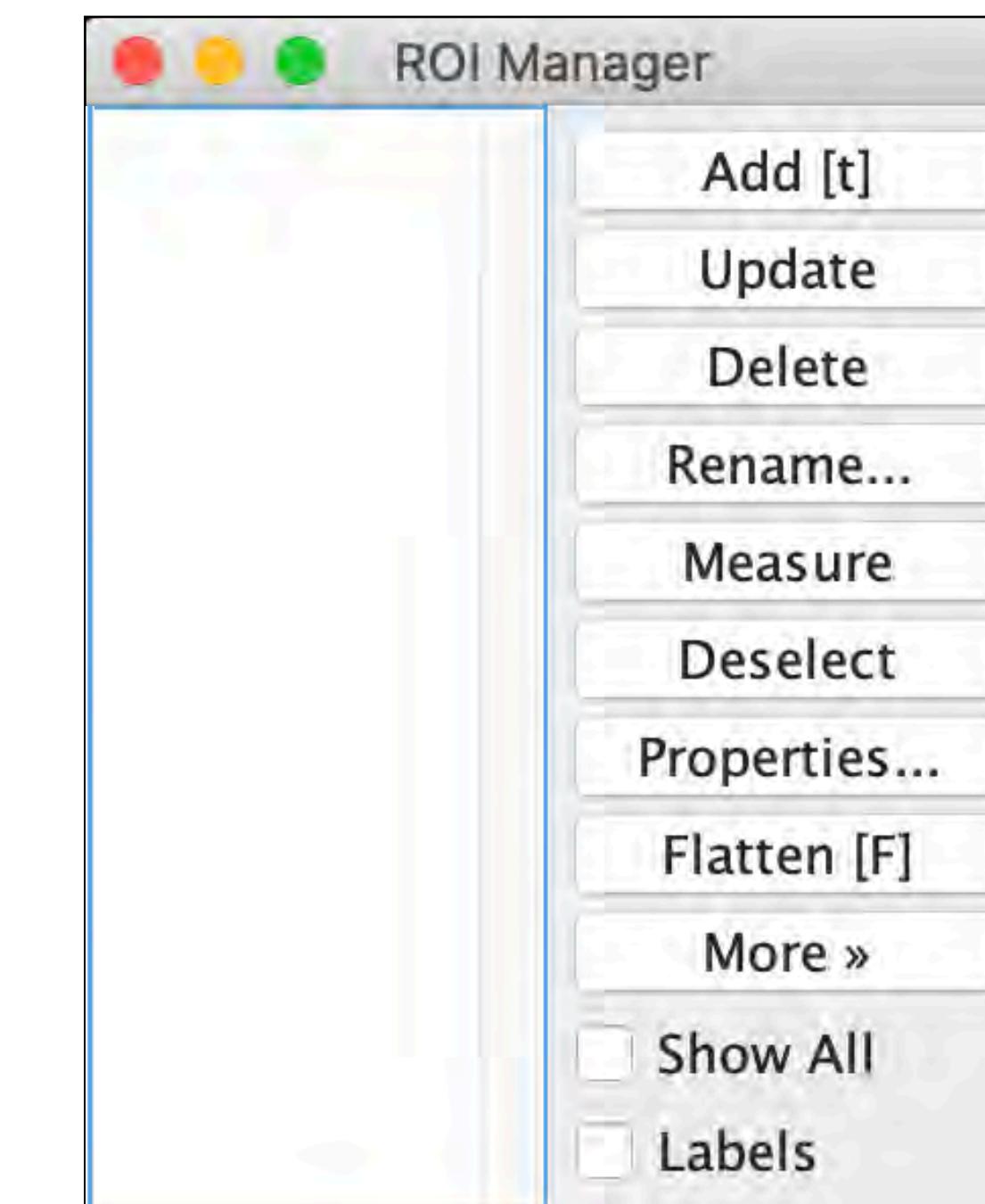
# Analyze particles: Add to ROI Manager

[Export](#)

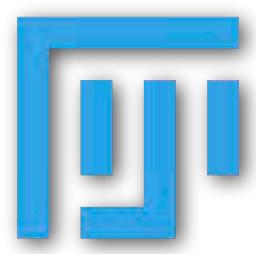
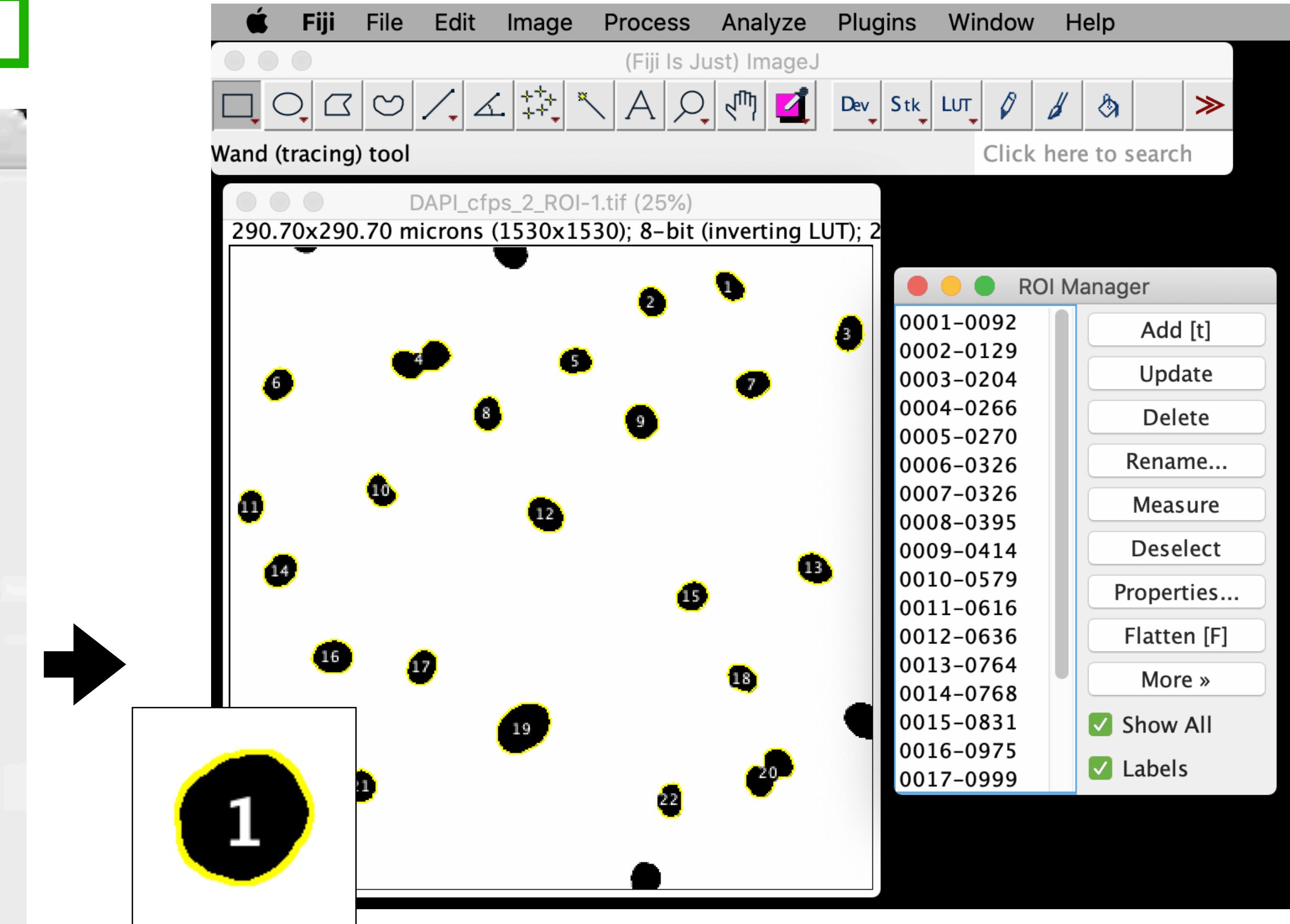
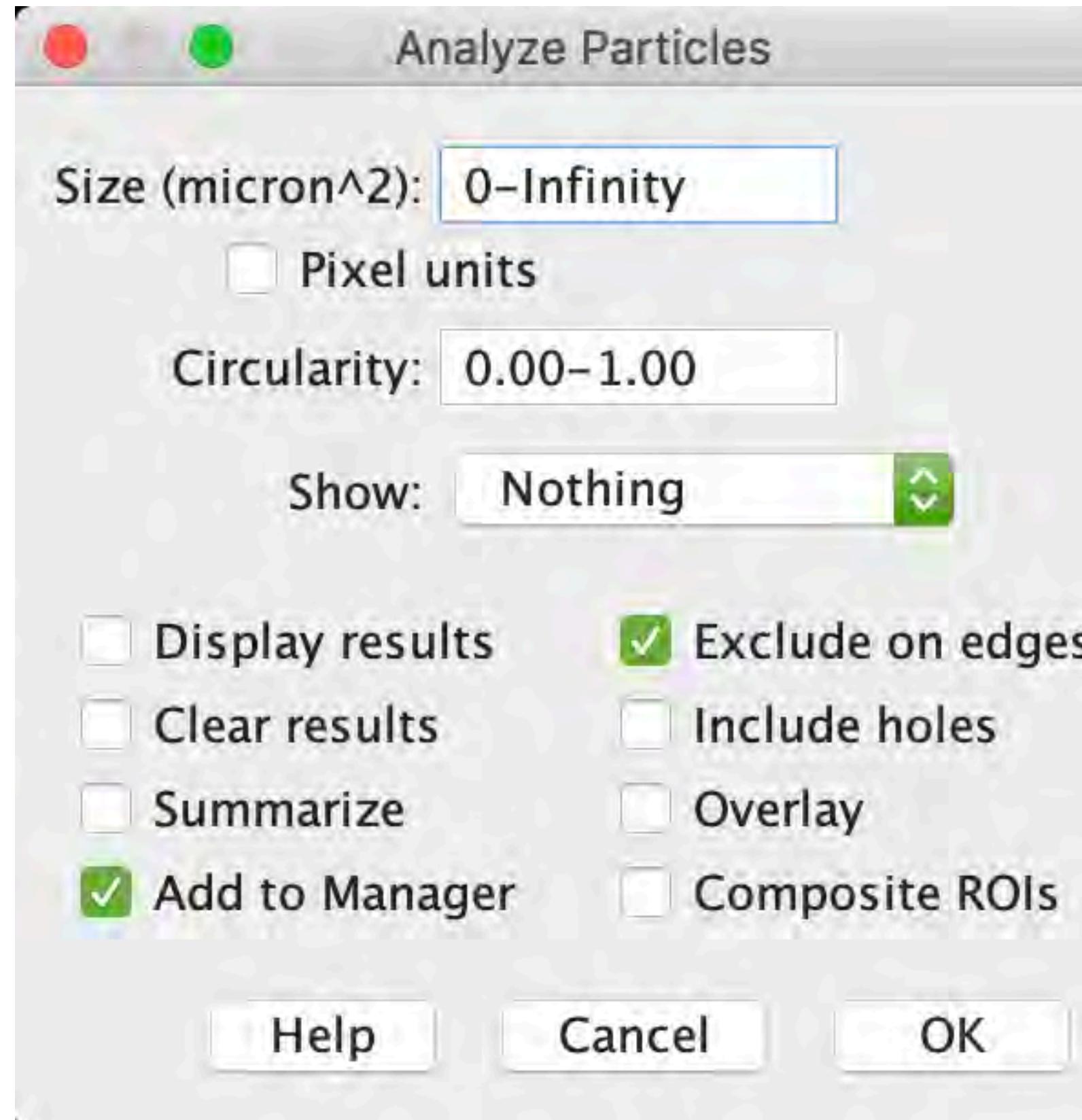
in **Fiji**: **Analyze > Analyze Particles...**



**"Add to Manager"** stores all found objects in the **ROI Manager**.

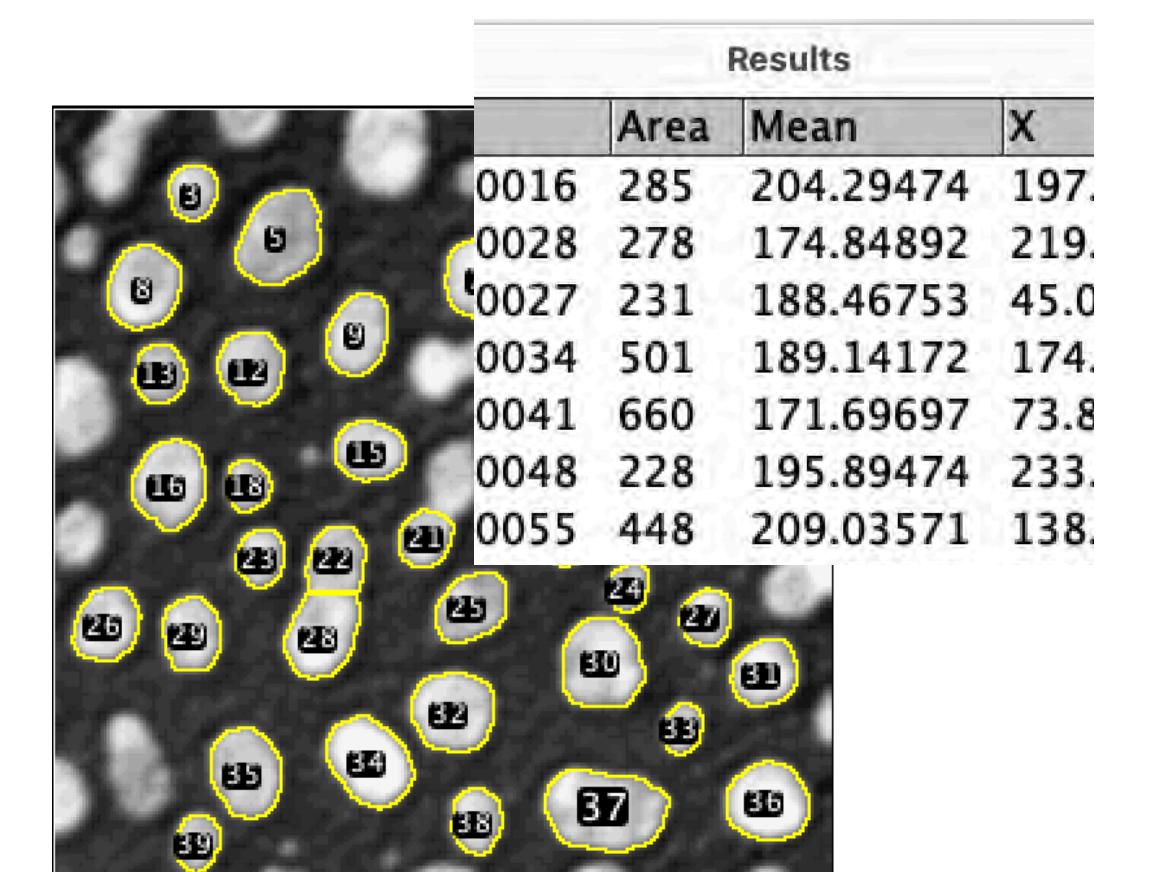


# Analyze particles: Add to ROI Manager

[Export](#)in **Fiji: Analyze > Analyze Particles...**

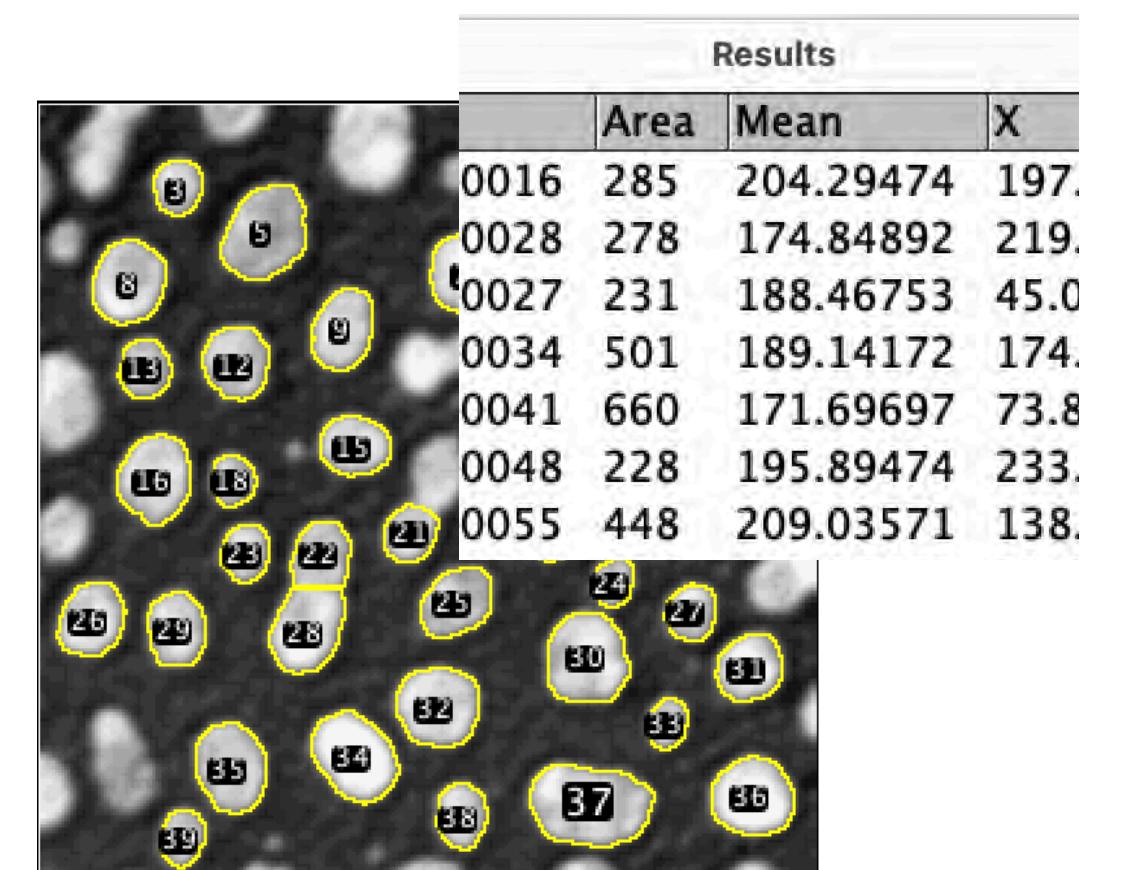
- Save the segmented image
- Add to ROI manager
- Generate and export measurements

## Export and Measure

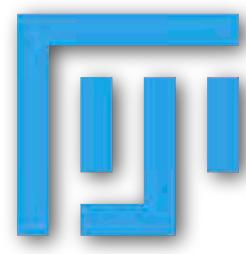


- Save the segmented image
- Add to ROI manager
- Generate and export measurements
  - Select what to measure
  - Measure

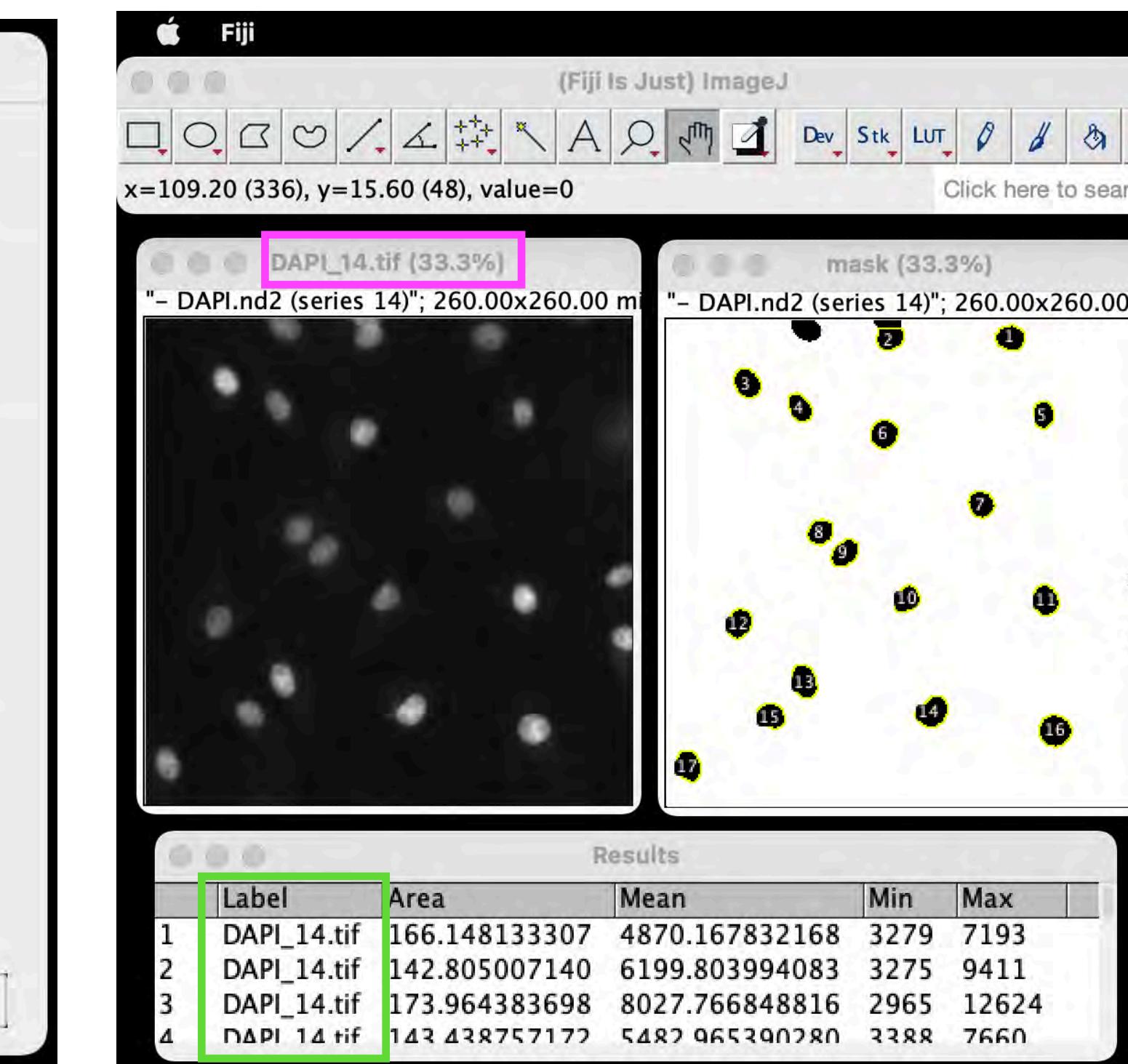
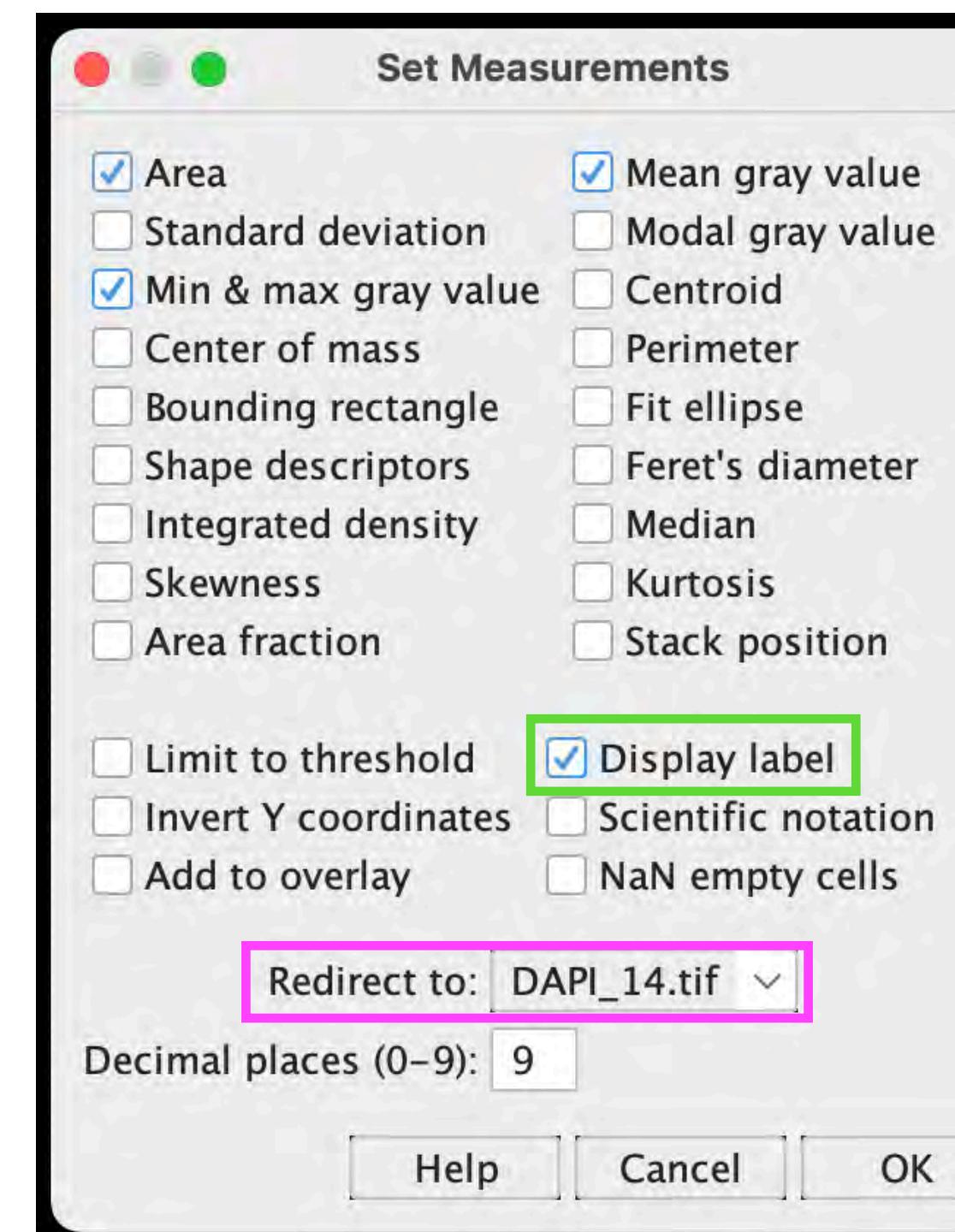
## Export and Measure



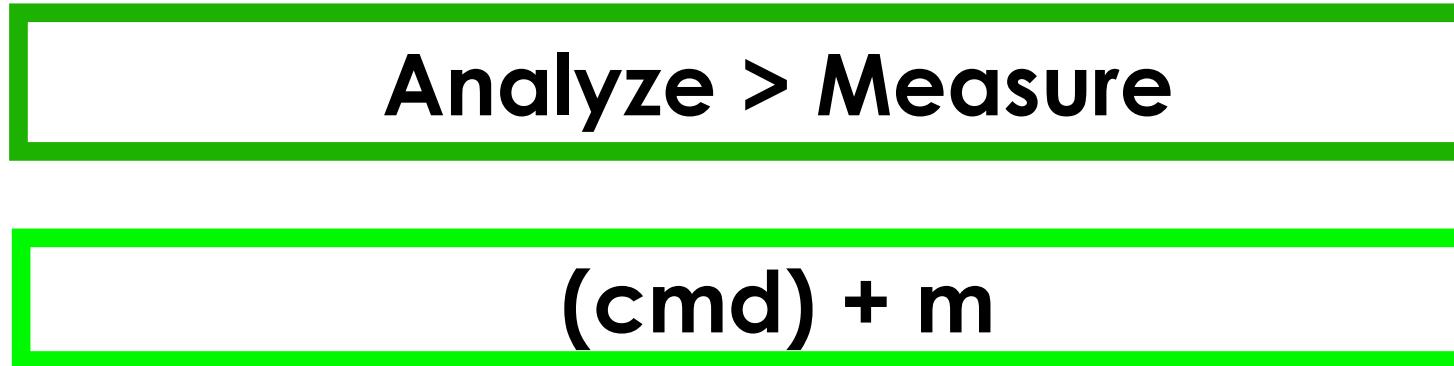
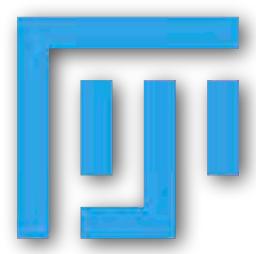
# Select what to measure: Set Measurements

[Export](#)**Analyze > Set Measurements...**

**Specifies which measurements have to be performed  
(e.g. area, mean grey value, max and min grey values, ...)**

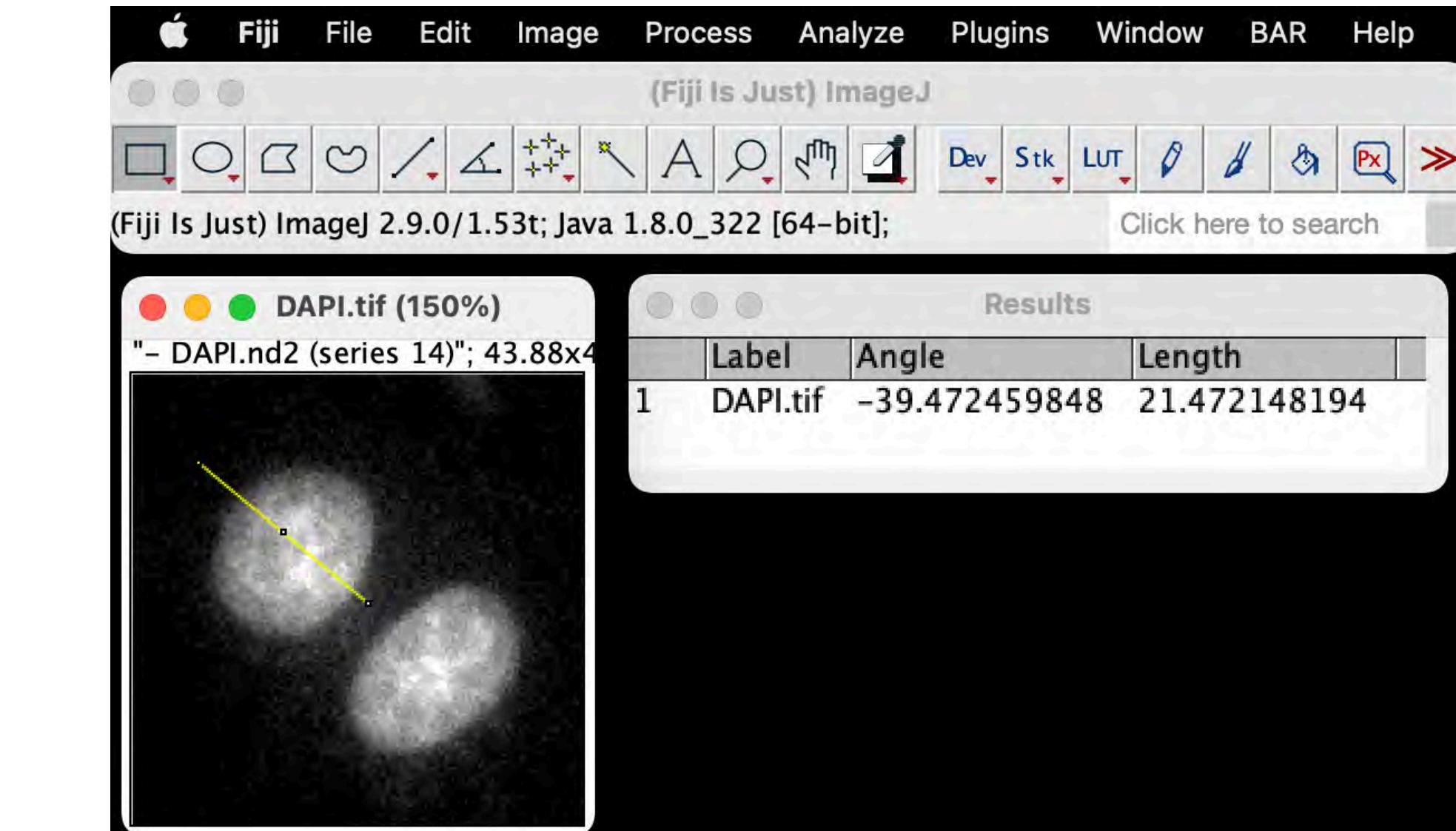
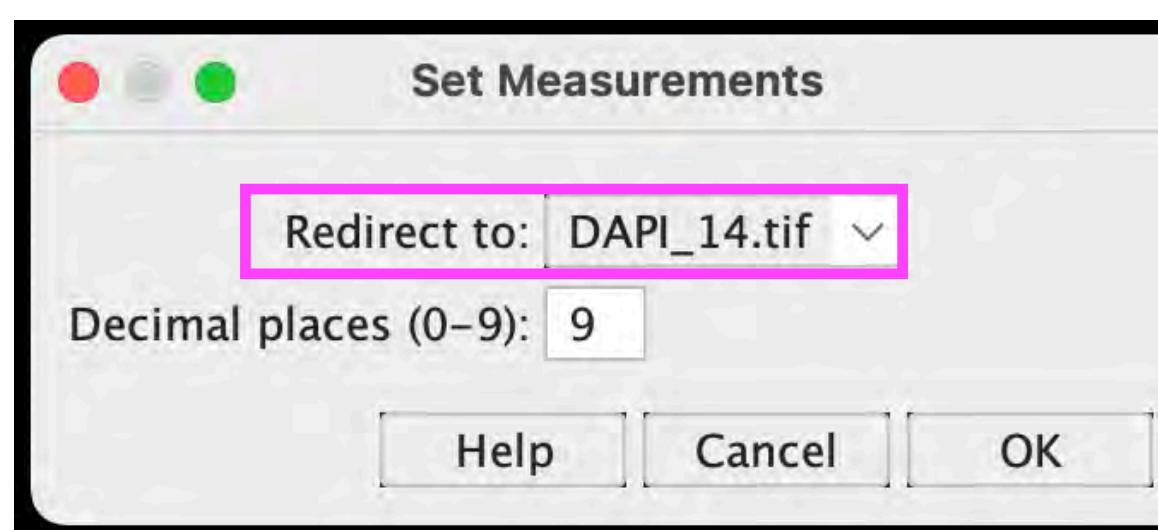


# Measure — Option 1: Manually

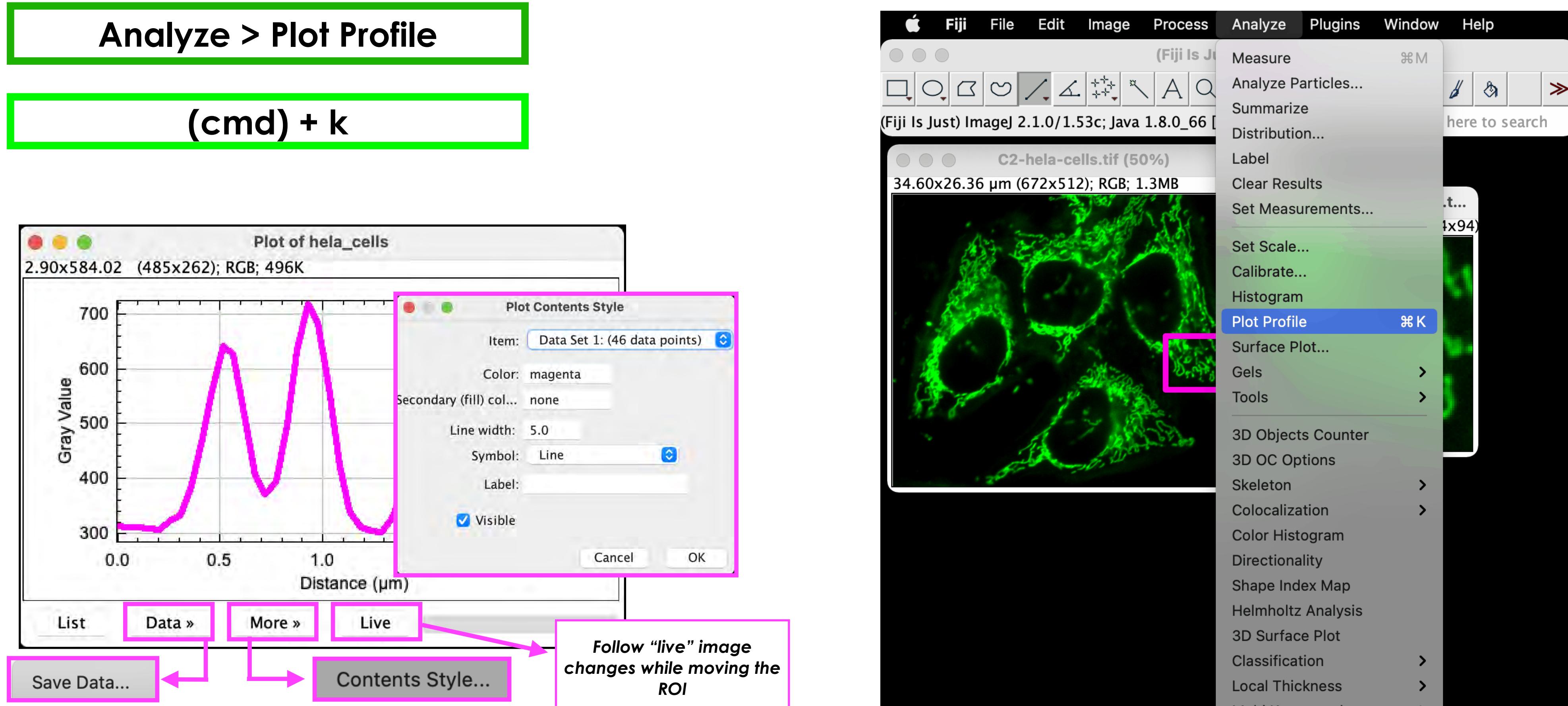
[Export](#)

*Measures the parameters chosen under  
“Analyze > Set Measurements...”  
in relation to the selected ROI.*

*Results are displayed in a Result Table  
(which can be saved as .csv, .xlsx, ...)*



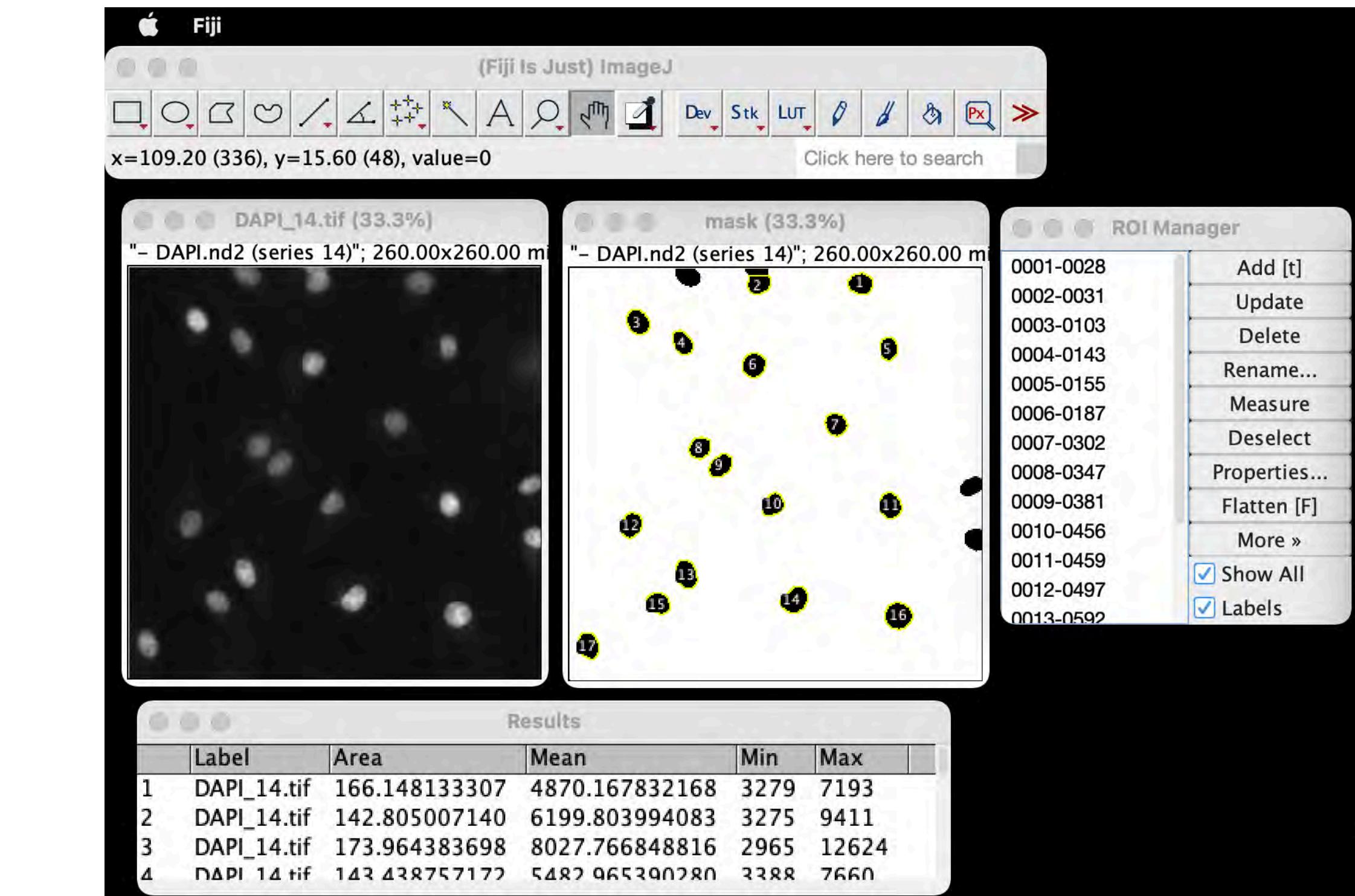
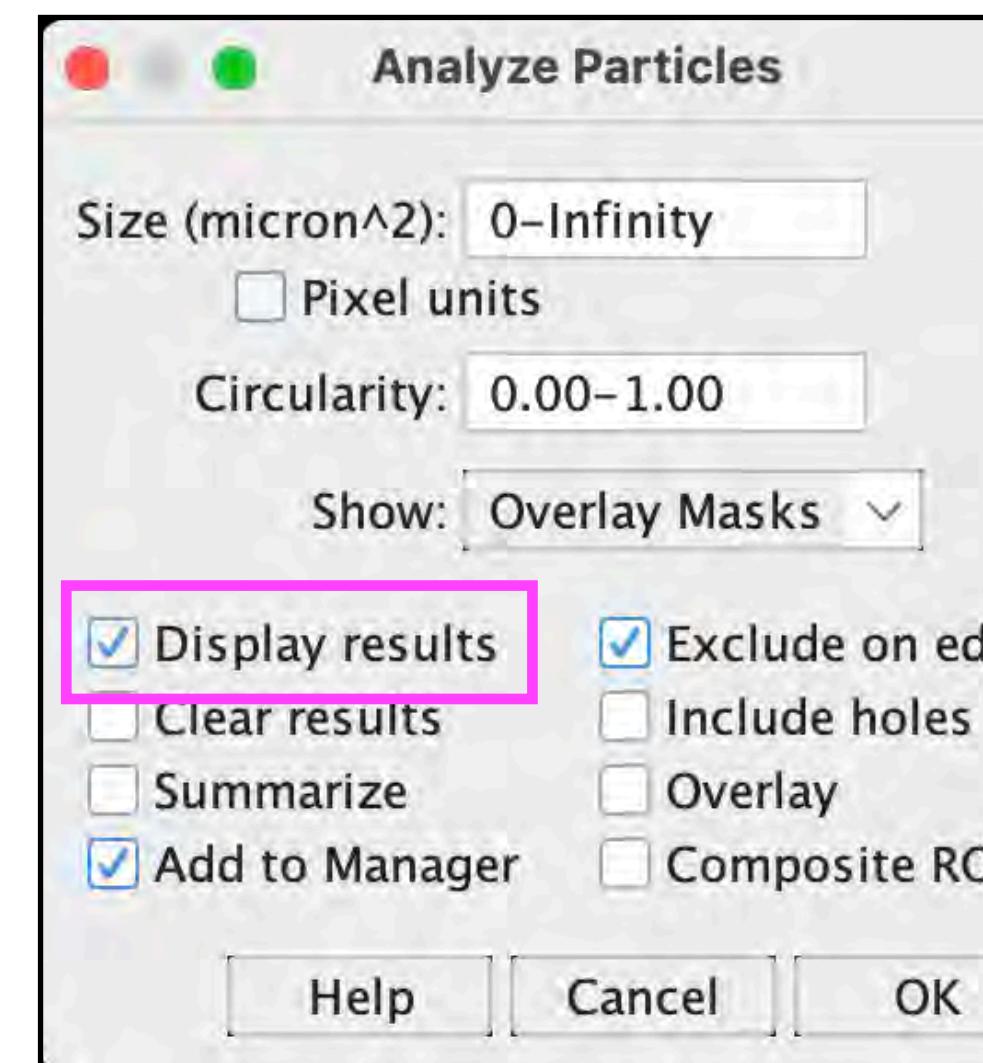
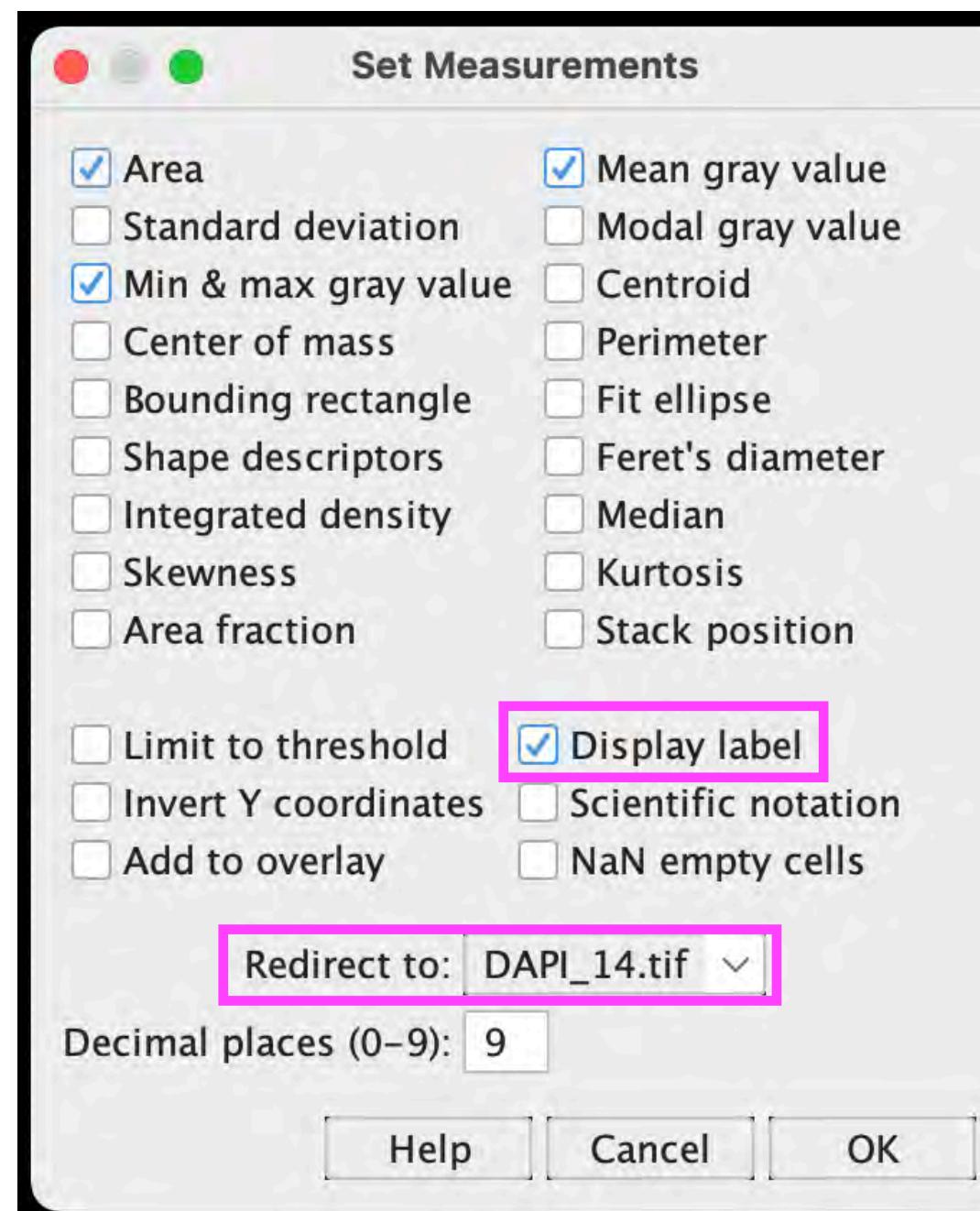
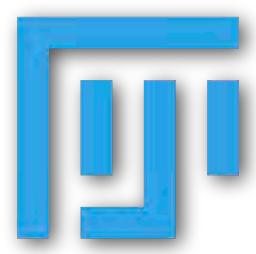
# Side note: other kinds of measurements



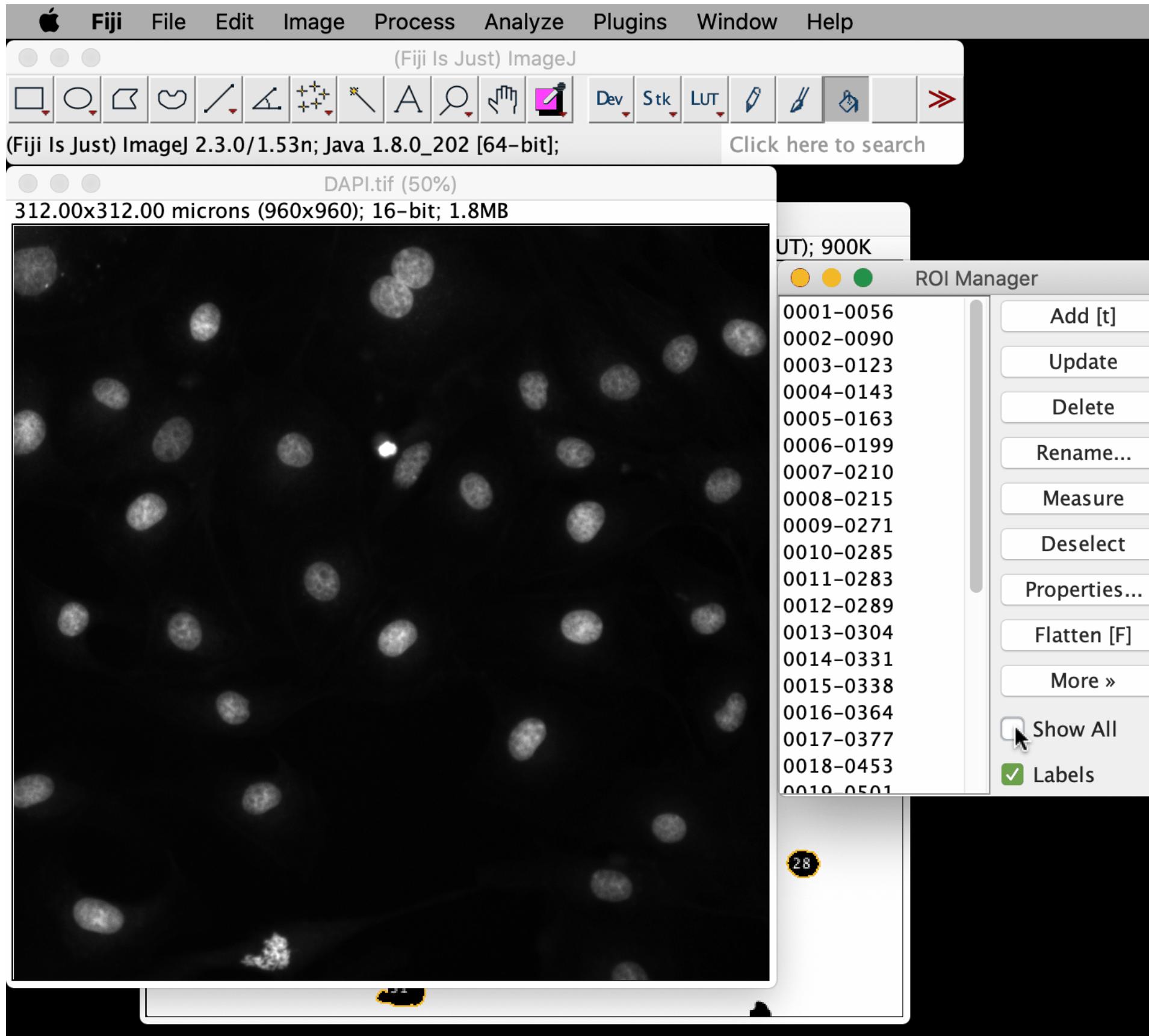
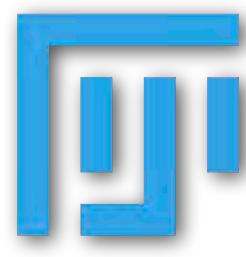
\*Plots can be saved as .csv file ("Save Data...") and also as images (e.g. "File > Save AS > PNG")

# Measure — Option 2: from Analyze particles

Export



# Measure — Option 3: Using the ROI manager

[Export](#)

1. Select the “ORIGINAL” image.
2. To measure all the ROIs stored in the “ROI Manager”, **no ROI has to be selected** in the “ROI Manager”. Use the “Deselect” button.
3. Click on the “**Measure**” button.
4. A “**Results**” table will appear.

# Segmentation with pixel based classifier—exercises



Continue with the “Analyze Particle” step  
form ONE of the workflow exercises below.

4.1 DAPI segmentation with thresholding

OR

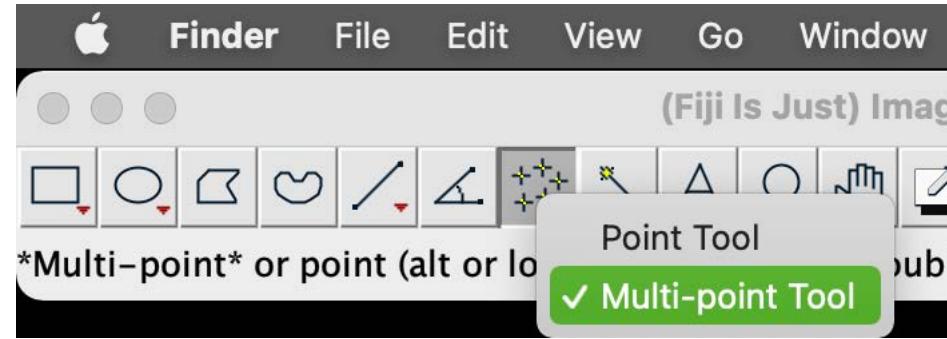
4.3 - DAPI segmentation with Labkit

# Spot Detection (Point tool and Find Maxima)

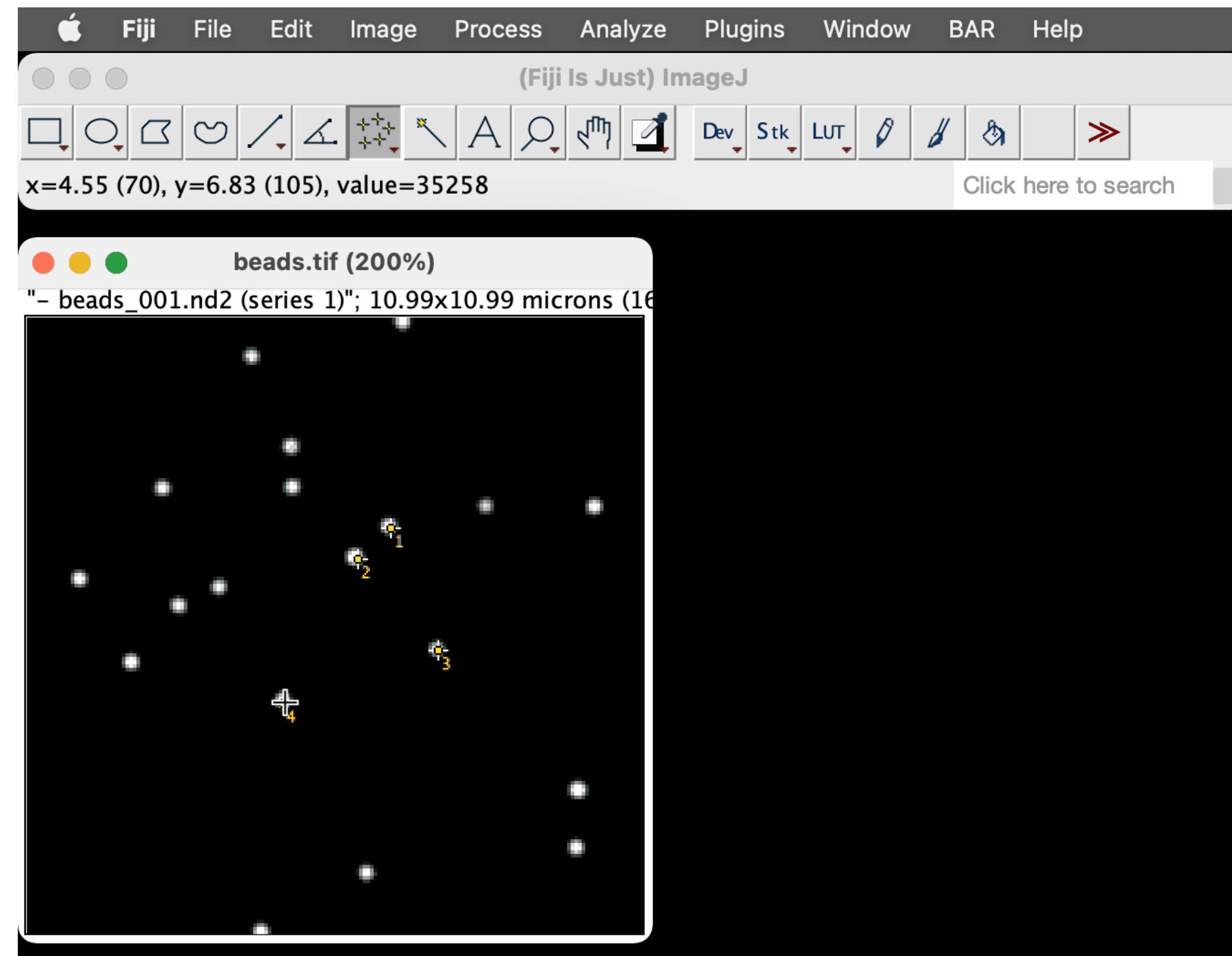
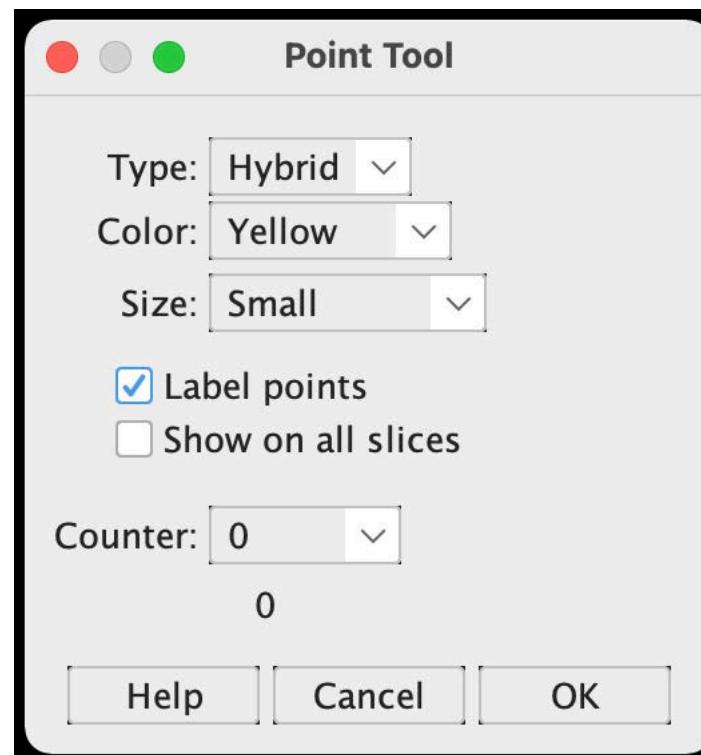
# Point and Multi-Point tool

*Select and Measure multiple points*

**Mouse right-click**



**Mouse double-click**



**to clear all points:**

**Edit > Selection > Select None \***

**(cmd) + shift + a**

**to measure all points:**

**Analyze > Measure \*\***

**(cmd) + m**

**to add point selection to ROI Manager:**

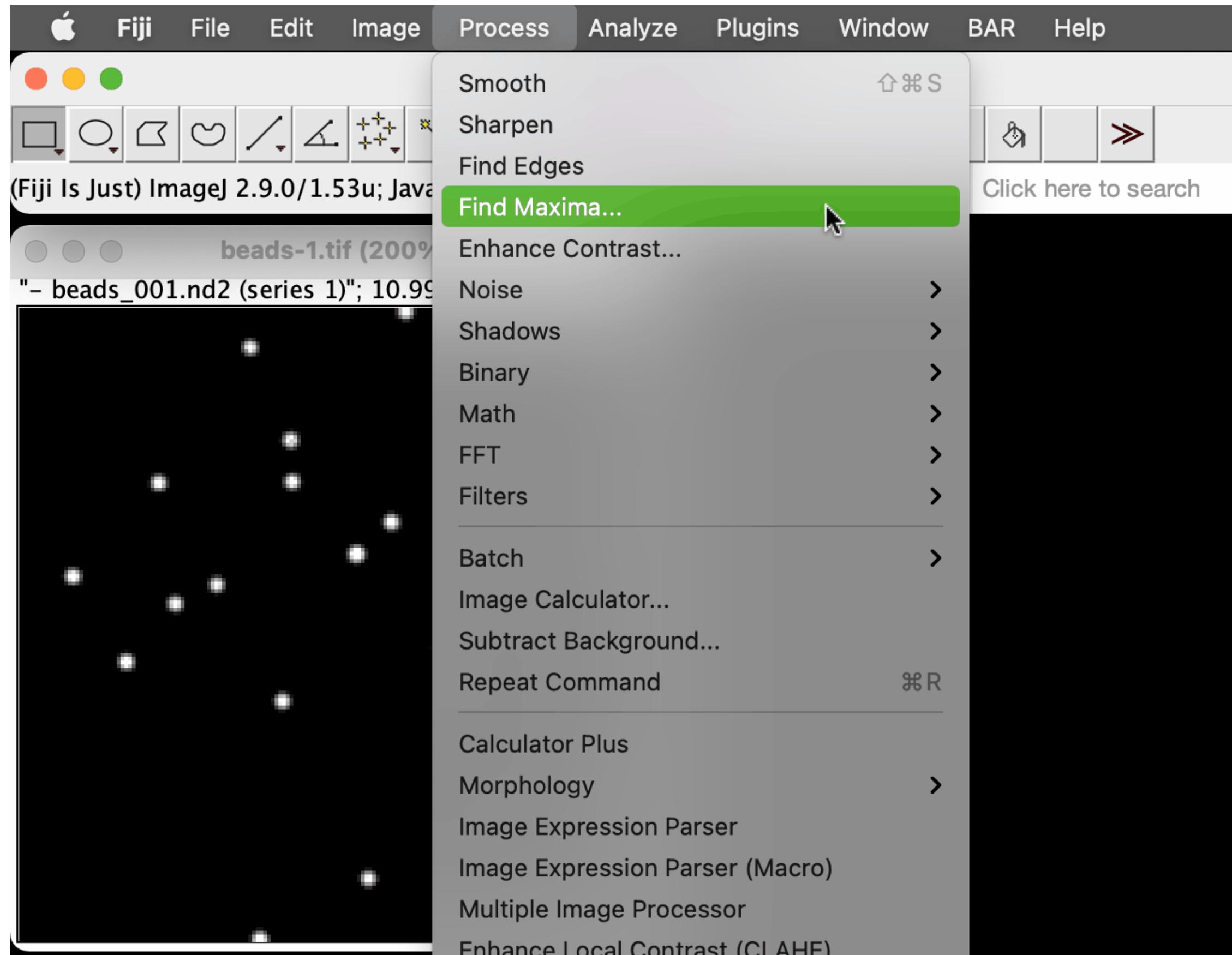
**t**

\*you can also toggle the “Show all” checkbox in the ROI Manager.

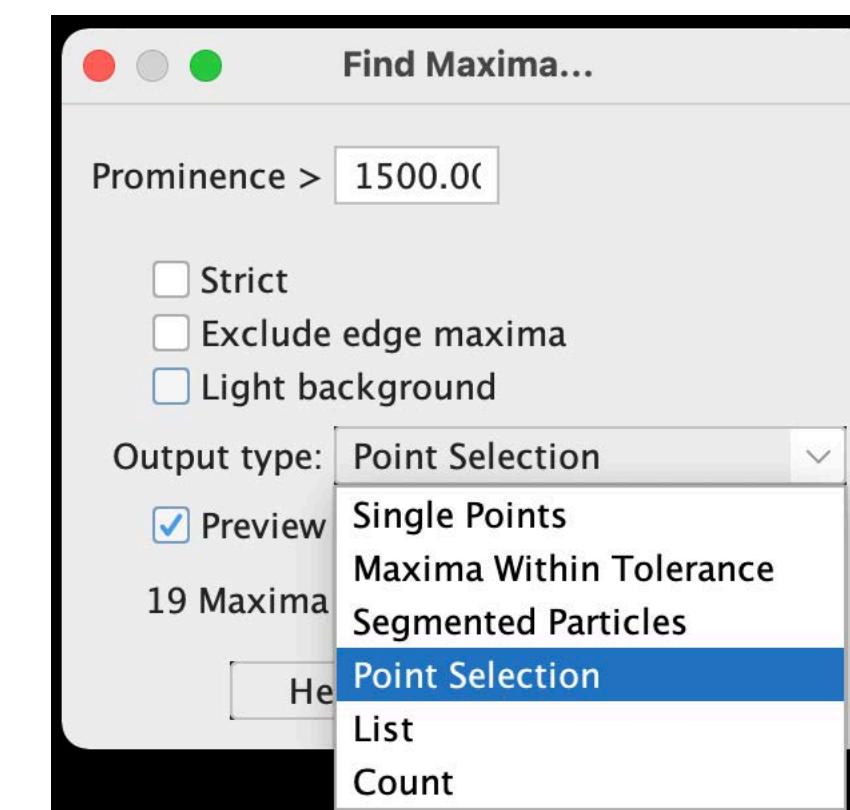
\*\*you can also use the “Measure” button in the ROI Manager.

# Point and Multi-Point tool

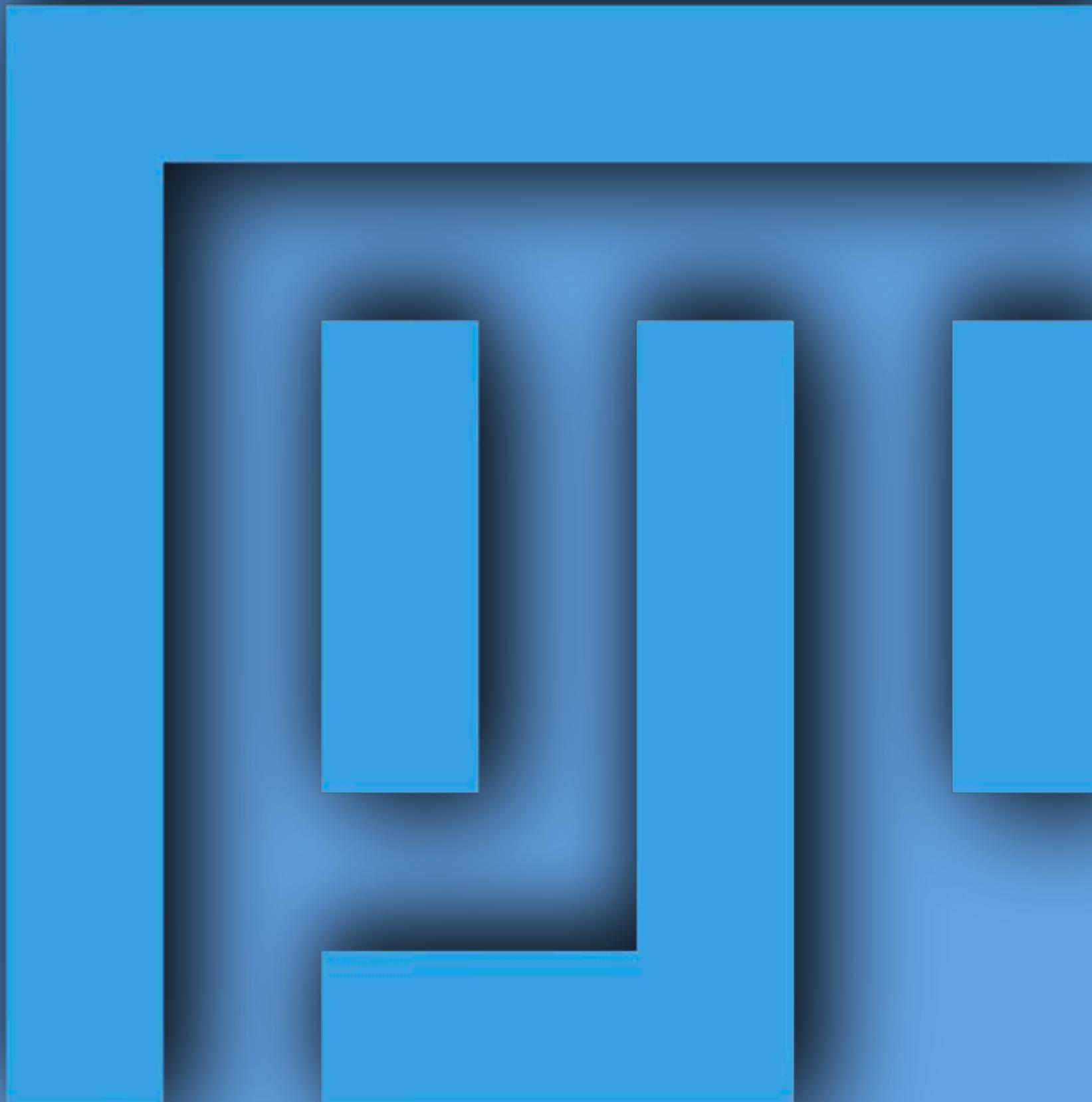
**Select and Measure multiple points**



**Process > Find Maxima...**



# Segmentation with pixel based classifier—exercises



**5.1 - Manual spot detection with the Multi-point Tool**

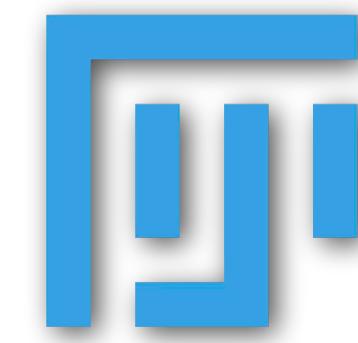
**5.2 Algorithmic spot detection with Find Maxima**

**5.3 Automatic spot segmentation with thresholding**

**5.4. Spot detection with noise**

**5.5. Spot detection with variable background**

# Further Learning (<https://iac.hms.harvard.edu/resources/>)



## Forum: Knowledge exchange and support

- <https://forum.image.sc/>

## Online book with code: Introduction to Bioimage Analysis

- <https://bioimagebook.github.io/>

## Online training: NEUBIAS Academy

- <https://eubias.org/NEUBIAS/training-schools/neubias-academy-home/>
- <https://www.youtube.com/c/NEUBIAS>

## Fiji manual from Monash University

- [https://bridges.monash.edu/articles/educational\\_resource/Fiji\\_Training\\_Manual\\_v6\\_4\\_20033513](https://bridges.monash.edu/articles/educational_resource/Fiji_Training_Manual_v6_4_20033513)

## BioImaging North America (BINA)

- <https://www.bioimagingnorthamerica.org/>