



**Das Deutsche Zentrum für  
Neurodegenerative Erkrankungen**

**Introduction to Bioimage analysis for  
Master of Science (M.Sc.) in Neurosciences**

**13/09/2021**

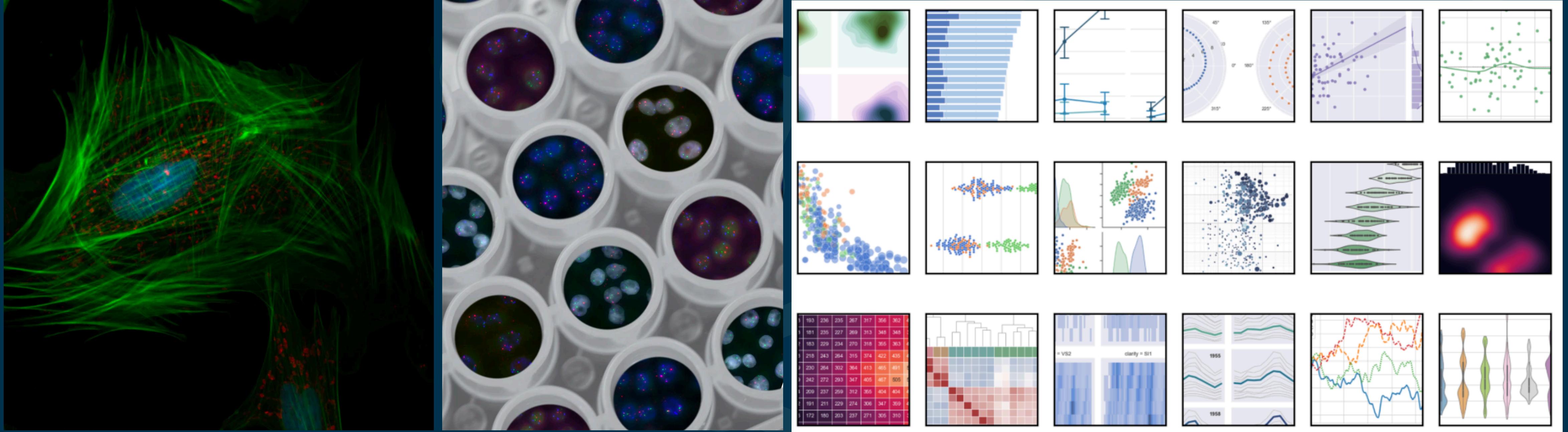
**Miguel Fernandes**  
**DZNE e. V. – Bonn / CRFS - Image and Data Analysis  
Facility (IDAF)**



# Talk outline:

- What is bioimage analysis?
- Basics of image processing:
  - What is a pixel?
  - Pixel size vs resolution
  - Voxels
  - Bit depth
  - Preprocessing
  - Segmentation and labelling
- Fiji:
  - Macro recorder
  - Improving a macro
  - Hands-on
- What shall we improve?
  - GPU-accelerated image analysis (CLIJ2)
  - AI vs ML vs DL
  - Trainable Weka Segmentation
  - Visualization of results and statistical analysis with Python

# What is bioimage analysis?



“The purpose of computing is insight, not numbers.”  
— Richard Hamming

Images from Unsplash and Seaborn website

# What is bioimage analysis?

One possible definition:

*The extraction of relevant information from digital images, at large scale and high throughput in the context of biological research*

- Precise
- Unbiased
- Reproducible
- Scalable

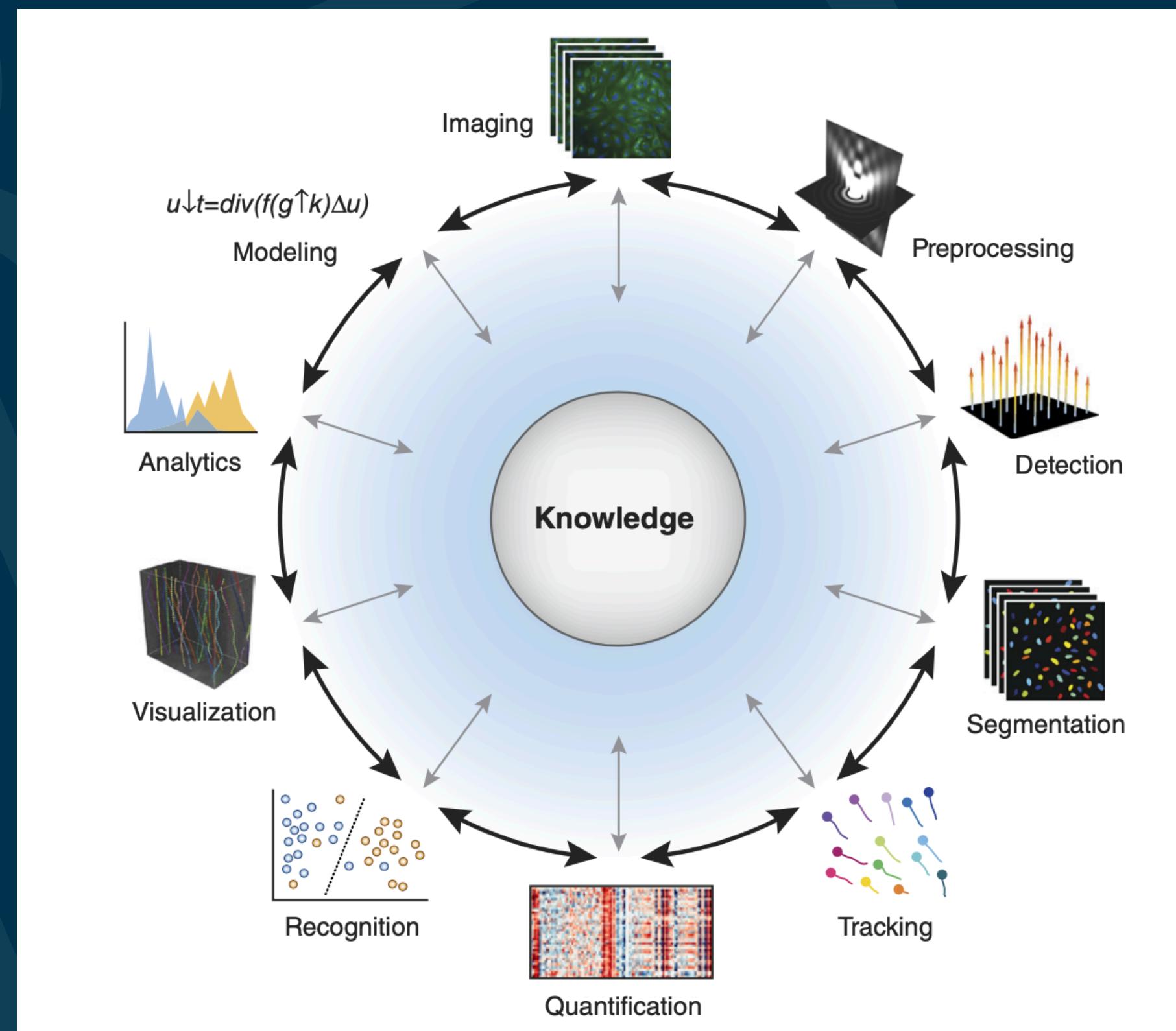
```
31     self._init_()
32     self.file = None
33     self.fingerprints = set()
34     self.logdups = True
35     self.debug = debug
36     self.logger = logging.getLogger(__name__)
37     if path:
38         self.file = open(os.path.join(path, 'request_fp.log'), 'w')
39         self.file.seek(0)
40         self.fingerprints.update(fp.readlines())
41
42     @classmethod
43     def from_settings(cls, settings):
44         debug = settings.getbool('BIOIMAGEREQUEST_DEBUG')
45         return cls(job_dir(settings), debug)
46
47     def request_seen(self, request):
48         fp = self.request_fingerprint(request)
49         if fp in self.fingerprints:
50             return True
51         self.fingerprints.add(fp)
52         if self.file:
53             self.file.write(fp + os.linesep)
54
55     def request_fingerprint(self, request):
56         return request_fingerprint(request)
```

Image from Unsplash

# What is the goal of Bioimage analysis?

*Obtain useful insights out of complicated and heterogeneous image and related metadata.*

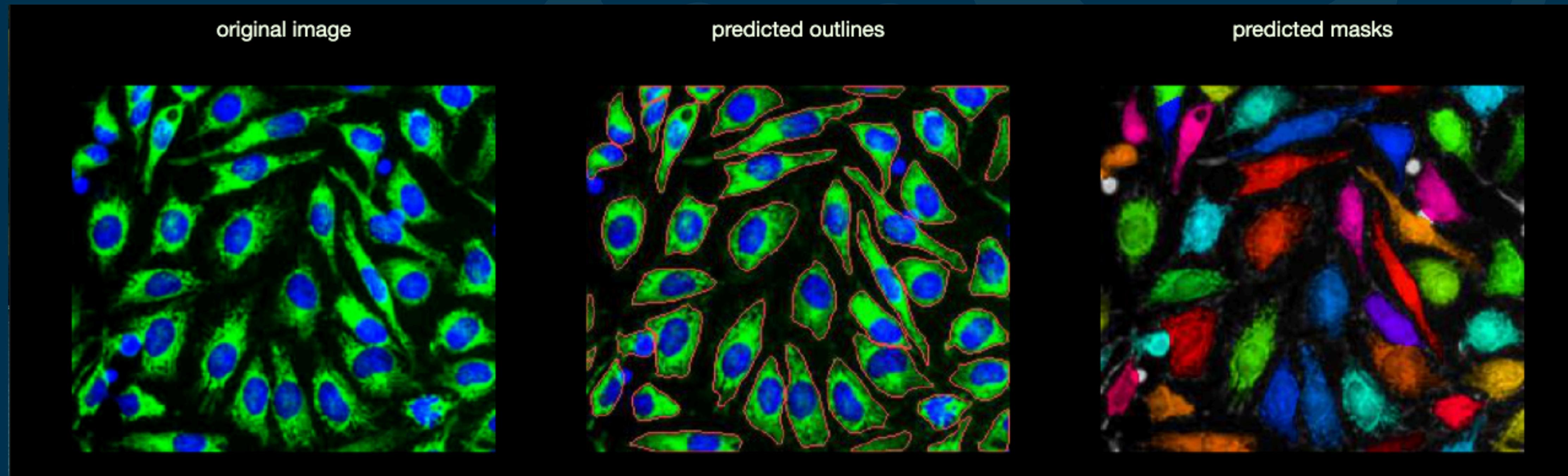
See [https://en.wikipedia.org/wiki/Bioimage\\_informatics](https://en.wikipedia.org/wiki/Bioimage_informatics) for details)



From <https://www.nature.com/articles/nbt.3722>

Meijering, E. et al. "Imagining the future of bioimage analysis."

# Example data analysis



Example Cellpose workflow

# Many different tools and steps

How to approach it?

Many resources freely available:

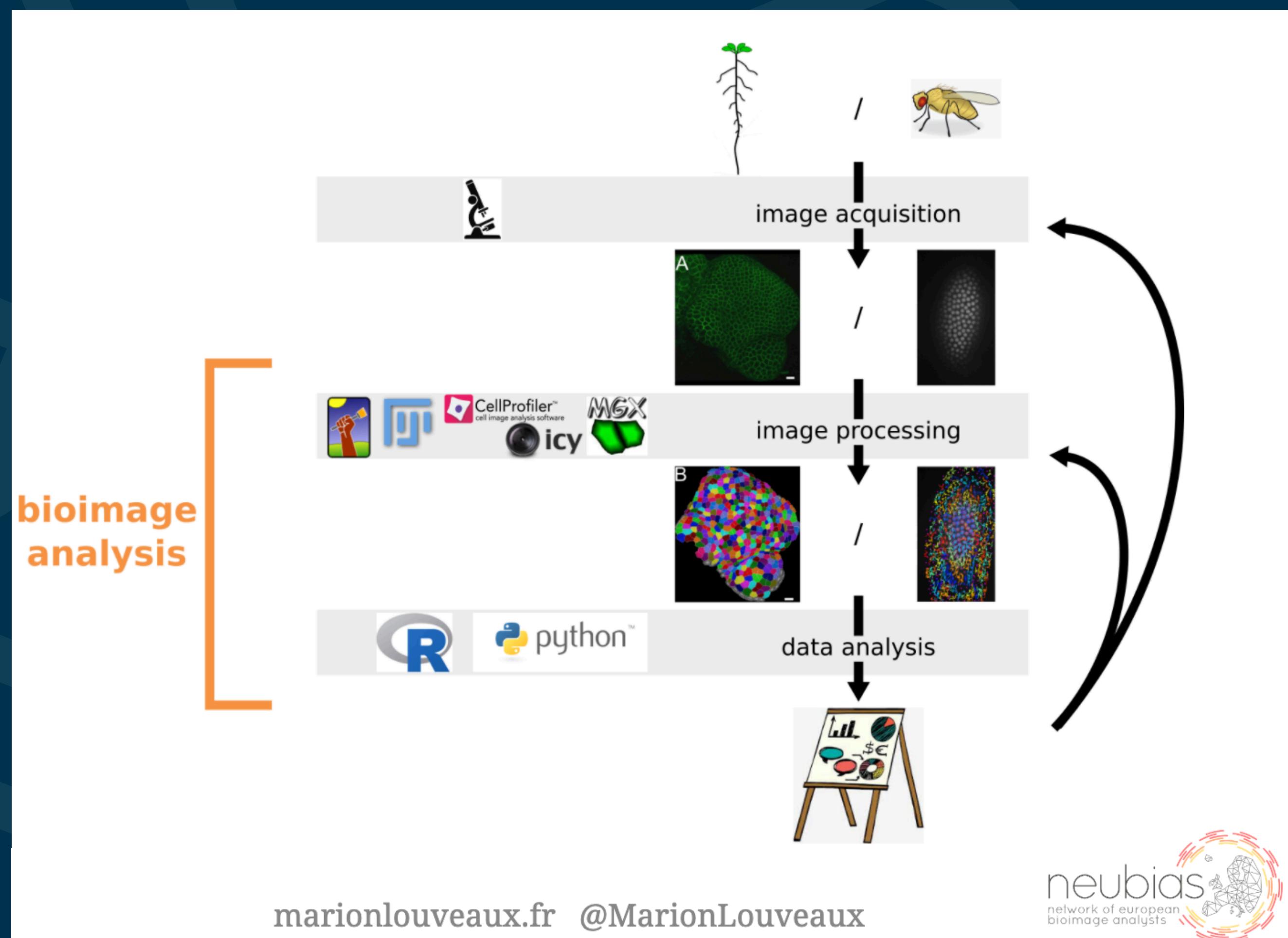
<https://forum.image.sc/>

<http://eubias.org/NEUBIAS/>

<http://biii.eu/>

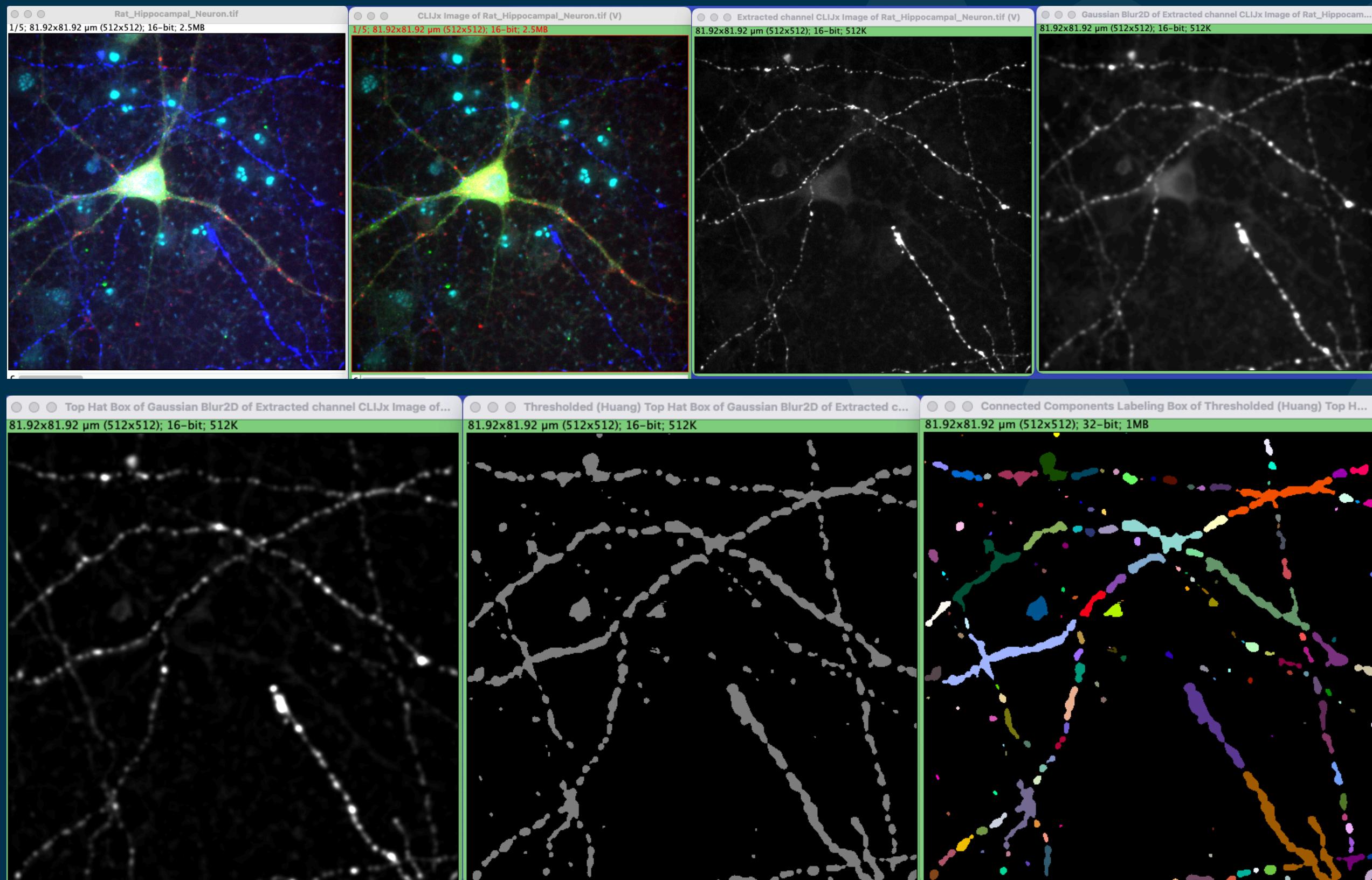
<https://www.youtube.com/c/haesleinhuepf/videos>

... and more



Modified from [https://github.com/marionlouveaux/NEUBIAS\\_TS15\\_DataViz](https://github.com/marionlouveaux/NEUBIAS_TS15_DataViz)

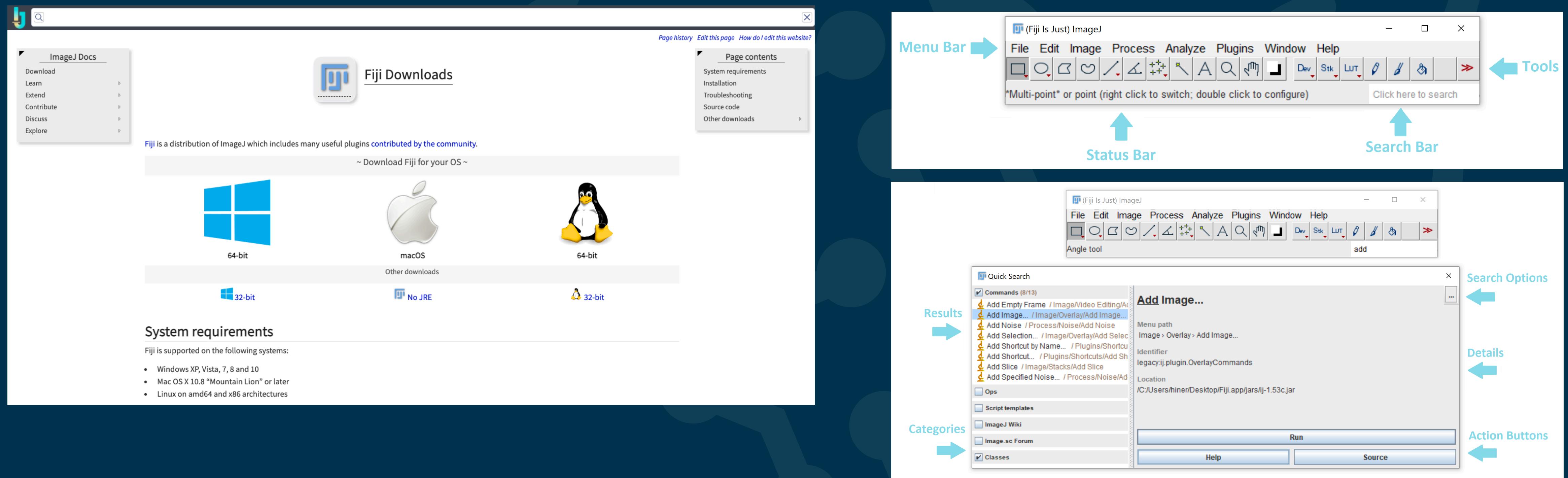
# Example today: Image J / Fiji



Fiji (Fiji Is Just ImageJ):  
distribution of ImageJ2, bundling a lot of  
plugins which facilitate scientific image  
analysis

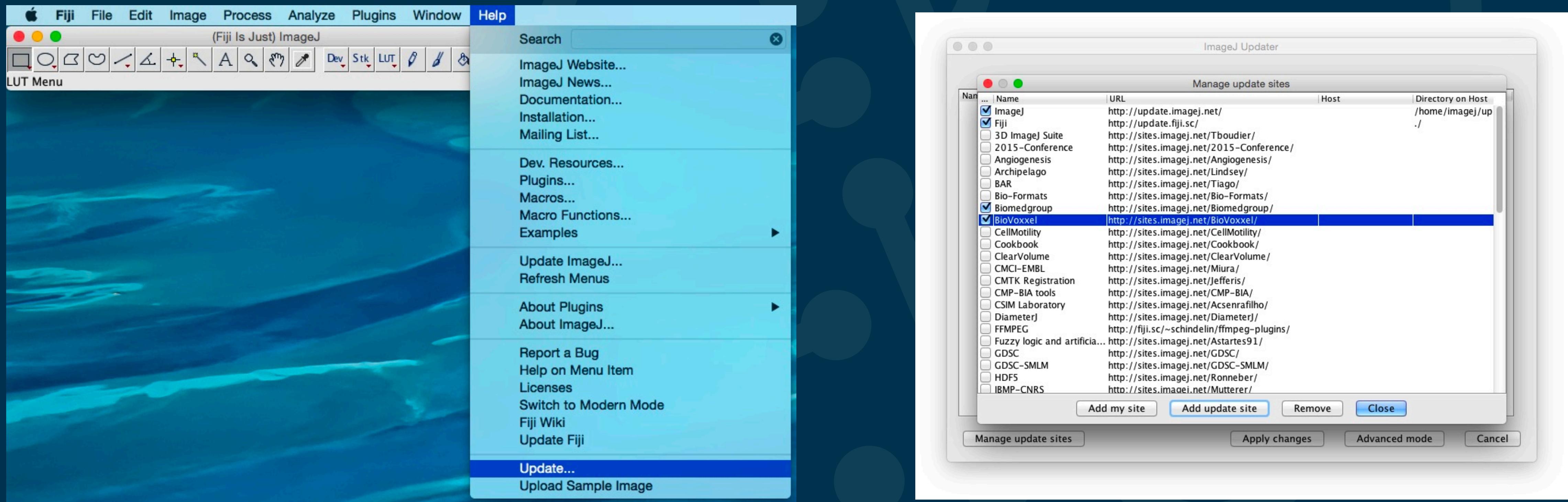
Example CLIJ2 analysis workflow  
<https://github.com/clij/clij2>

# Installing FIJI



From <https://imagej.net>

# Updating FIJI and installing plugins, scripts and macros



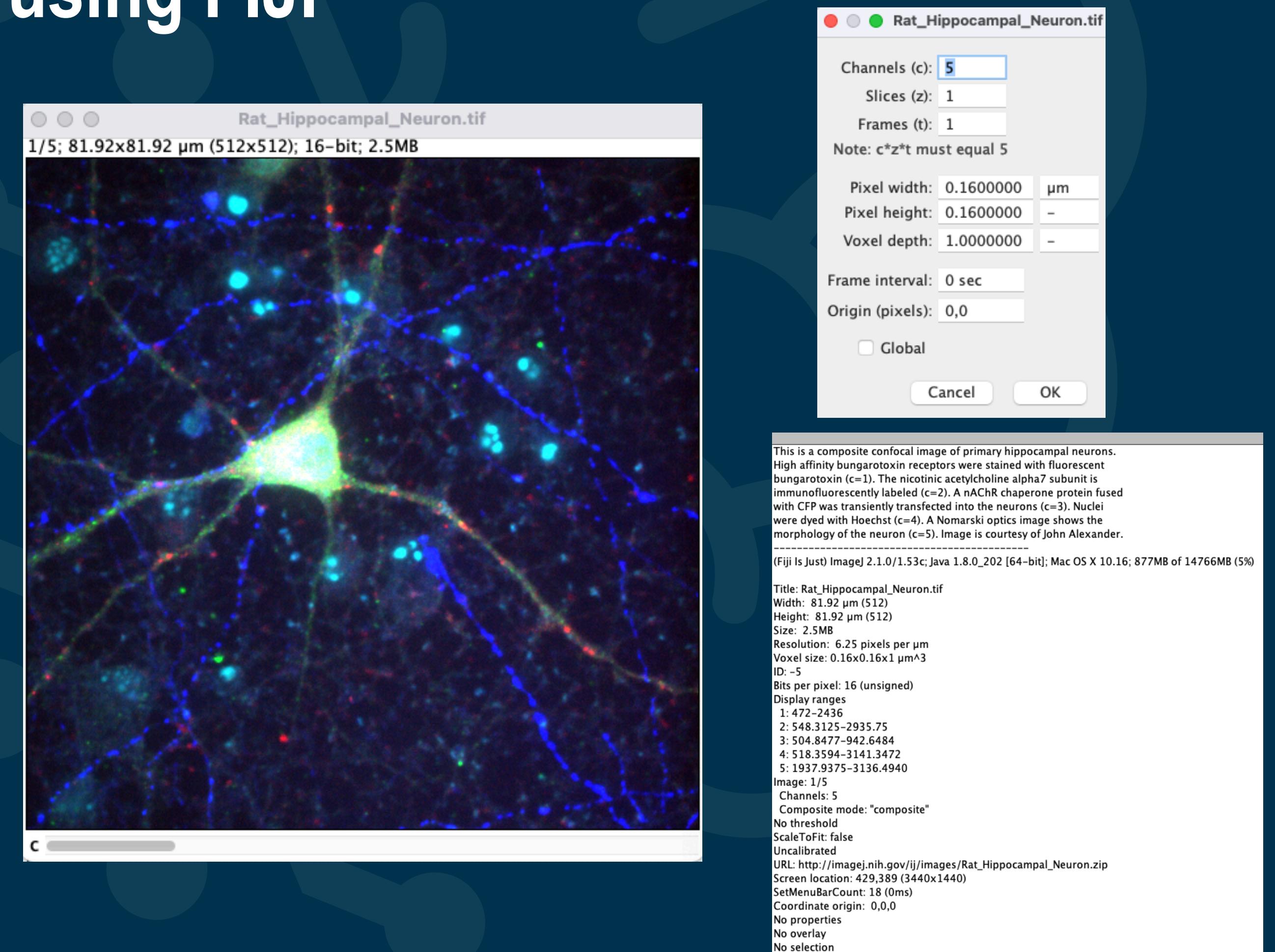
From <https://imagej.net>

# Check properties of the image using FIJI

Image > Properties

and

Image > Show Information



# Basics of image processing

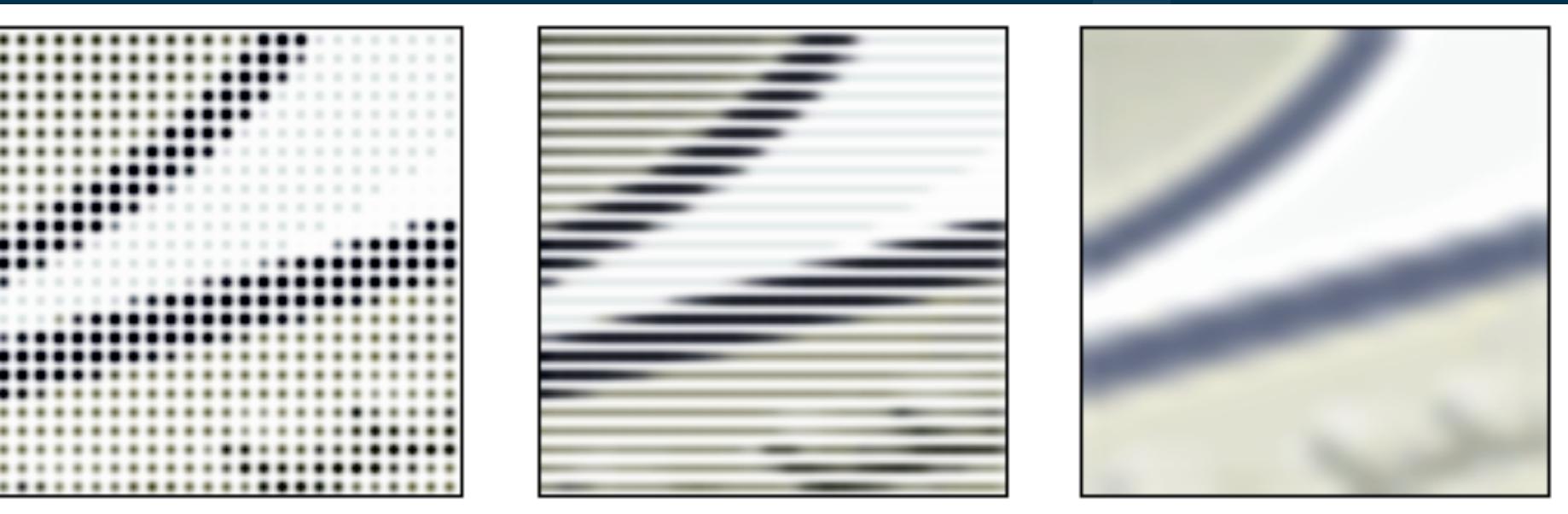
# What is a pixel?

A pixel (“picture element”) is the smallest unit of a digital image or graphic that can be displayed and represented on a digital display device.

Pixels are combined to form a complete image, video, text, or any visible thing on a computer display.

## A Pixel Is Not A Little Square!!!

A pixel is a data point, a point sample. A pixel does not need to be rendered as a small square.



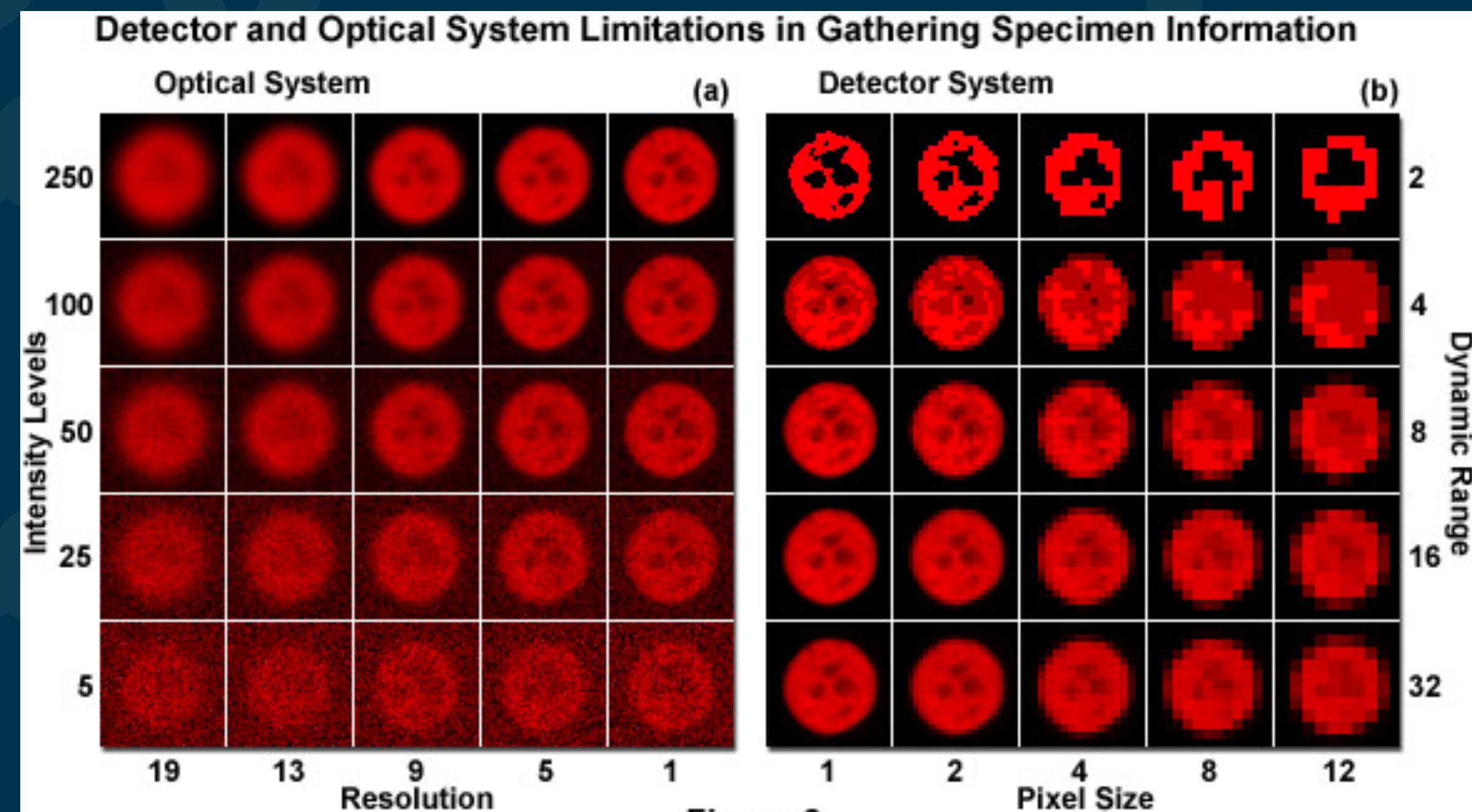
<https://en.wikipedia.org/wiki/Pixel>

# Pixel size vs resolution

Pixel size is a property you configure during imaging at the microscope

Resolution is a property of your imaging system.

Spatial resolution: measure of how closely lines can be located in an image while still being differentiable



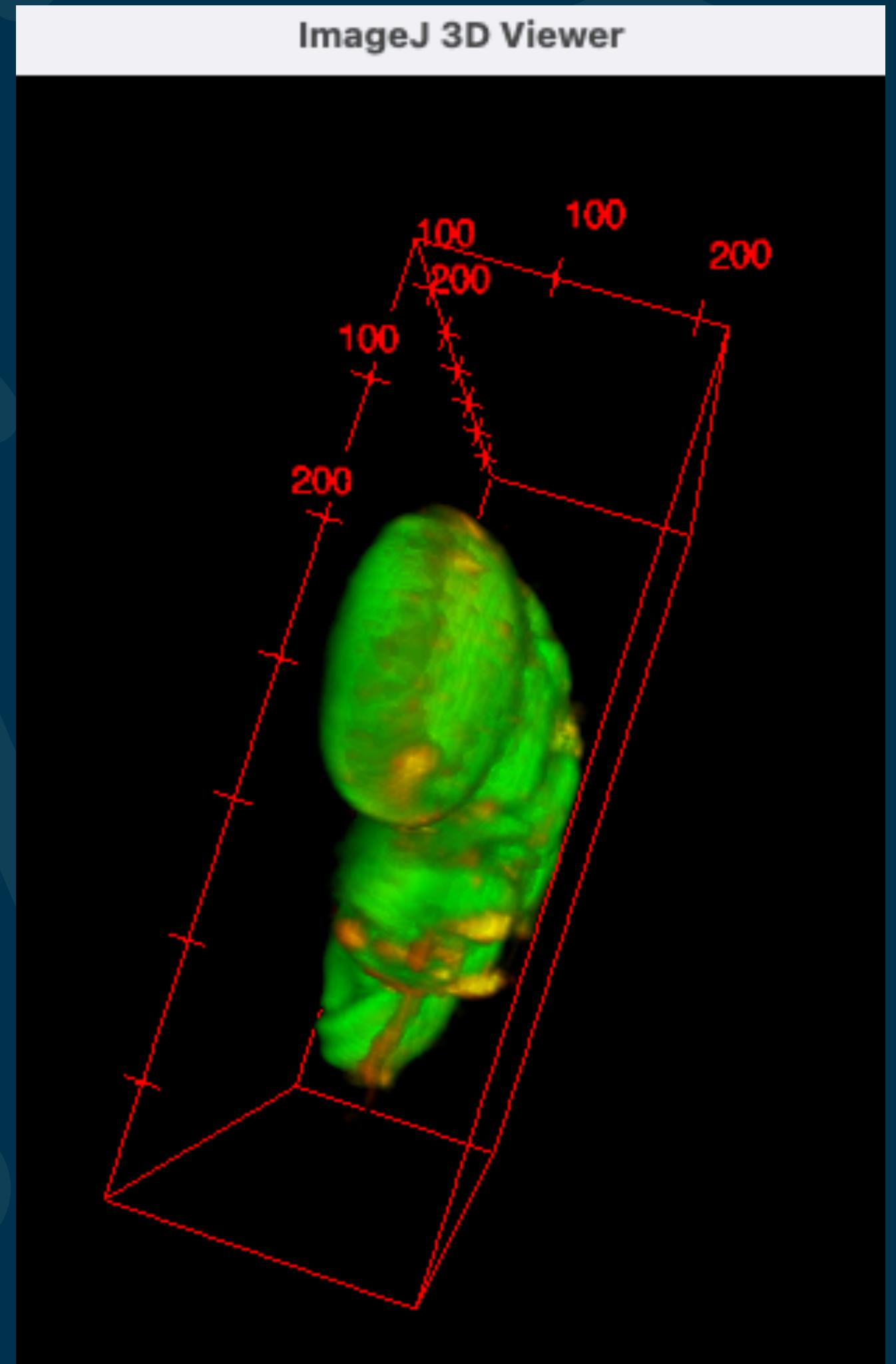
<http://zeiss-campus.magnet.fsu.edu/print/livecellimaging/digitalimaging-print.html>

# Voxels (3D data)

Voxel: “Volume element”, the smallest unit of the sampled 3D volume

Usually anisotropic: not the same in all directions

The section thickness together with the xy-pixel dimension defines the voxel size.



# Bit depth

The bit-depth  $n$  enumerates how many different intensity values are present in an image:  $2^n$  grey values

In microscopy, images are usually stored as 8, 12 or 16-bit images.

Higher bit depths corresponding to a greater range of useful image information available

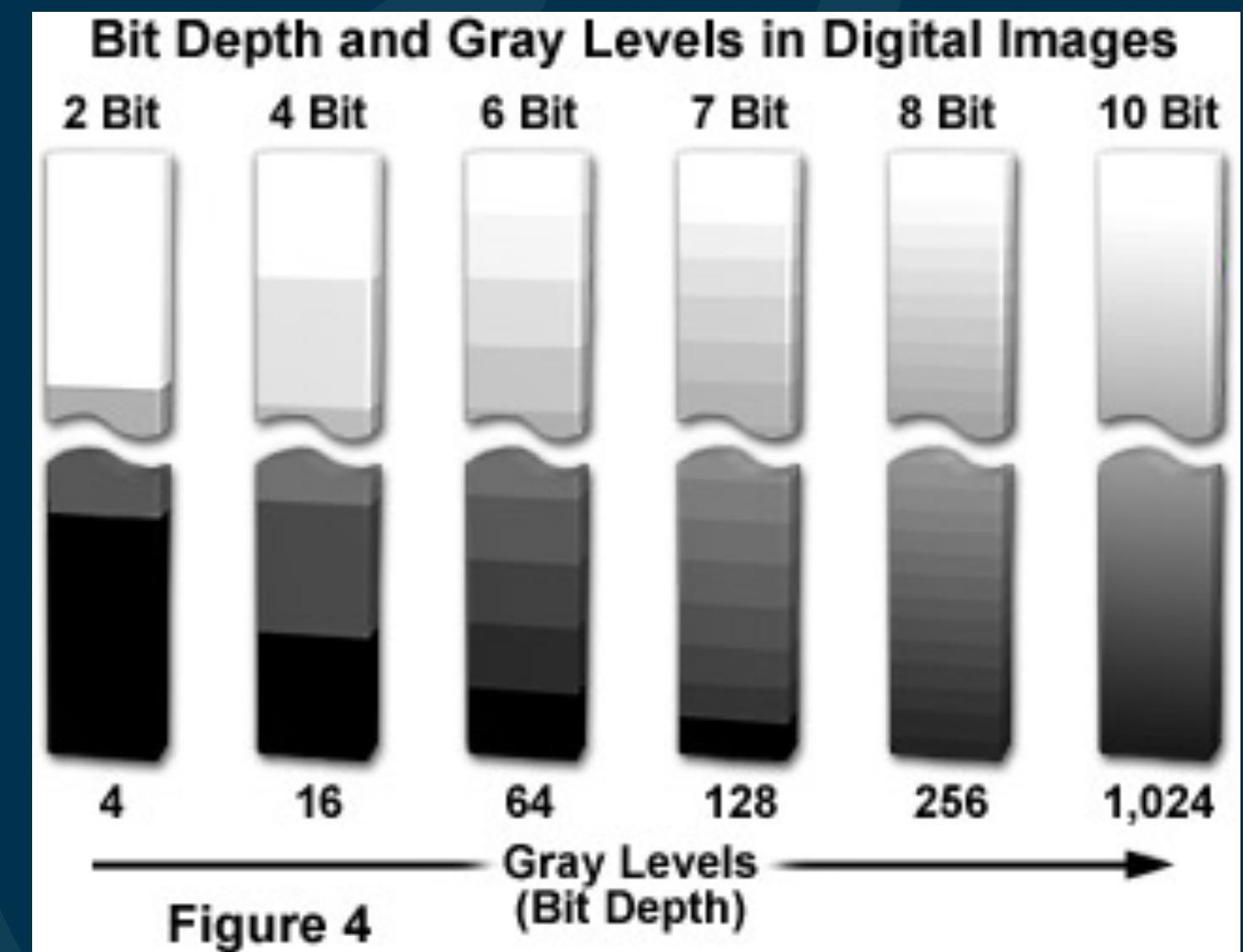
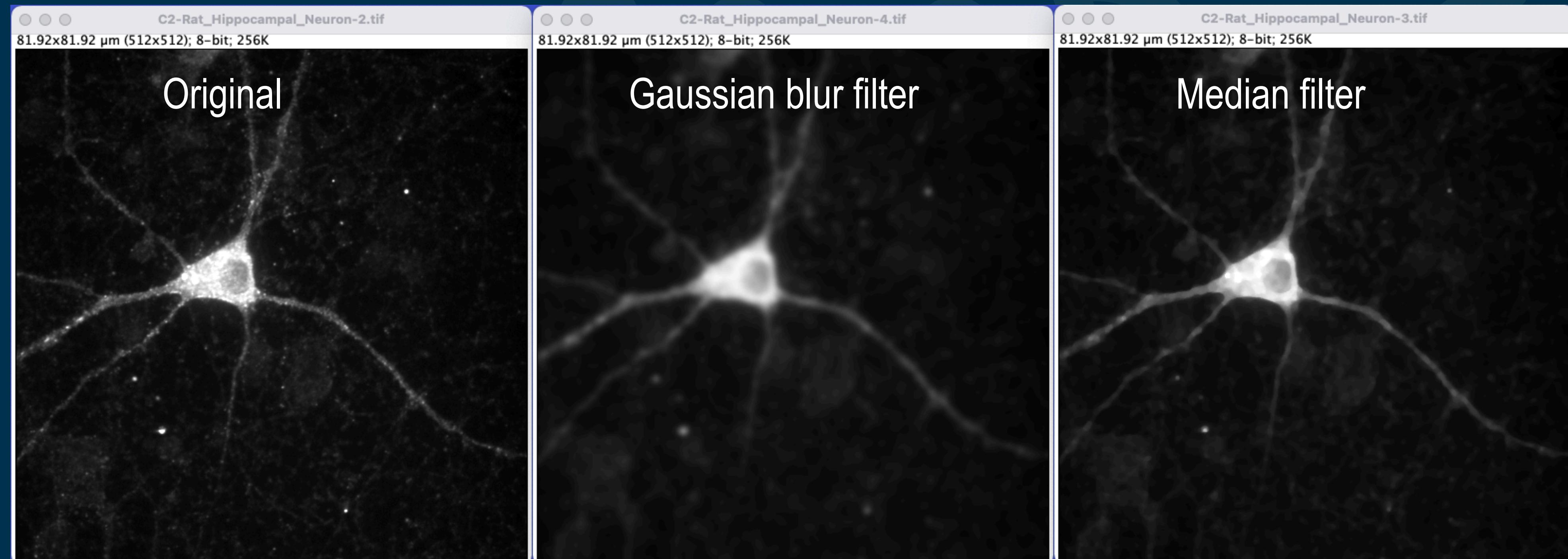
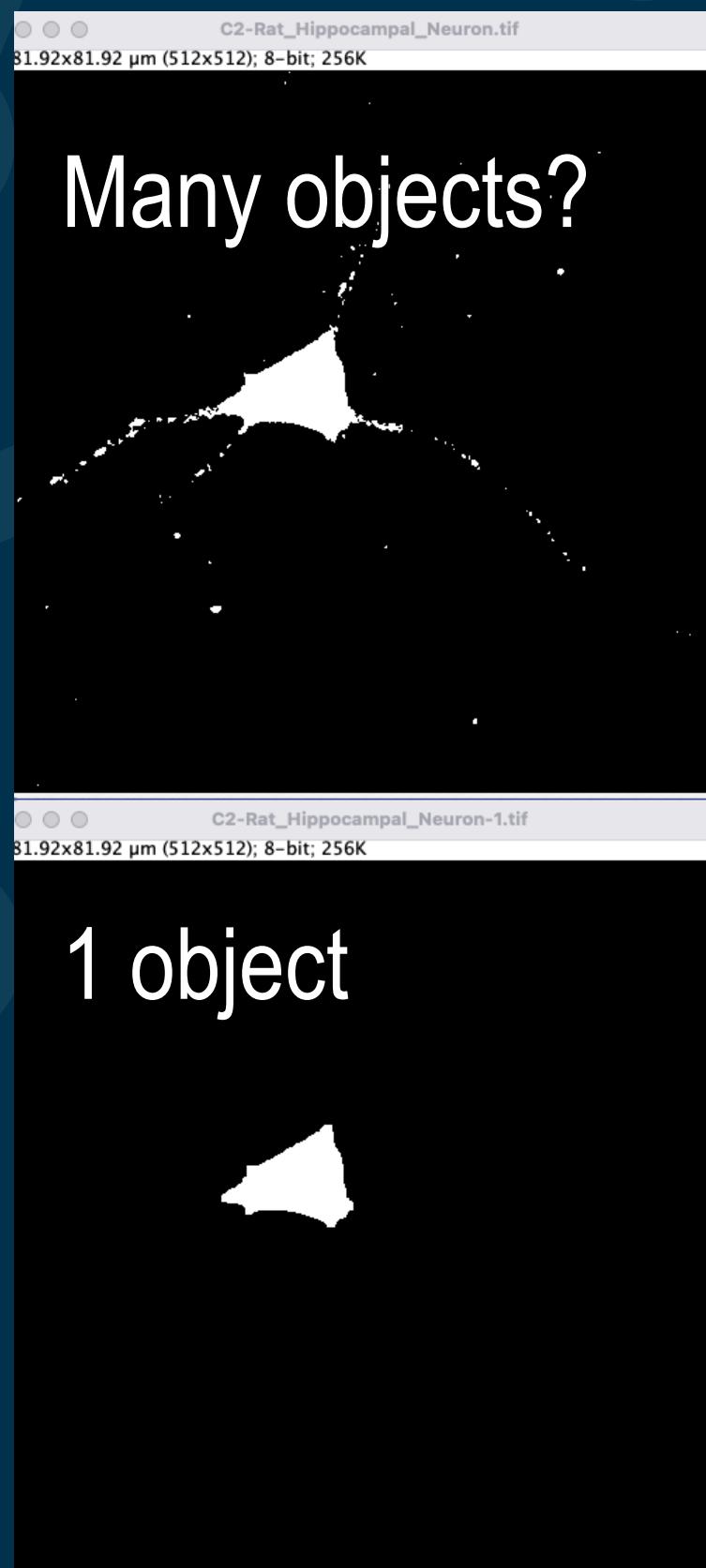
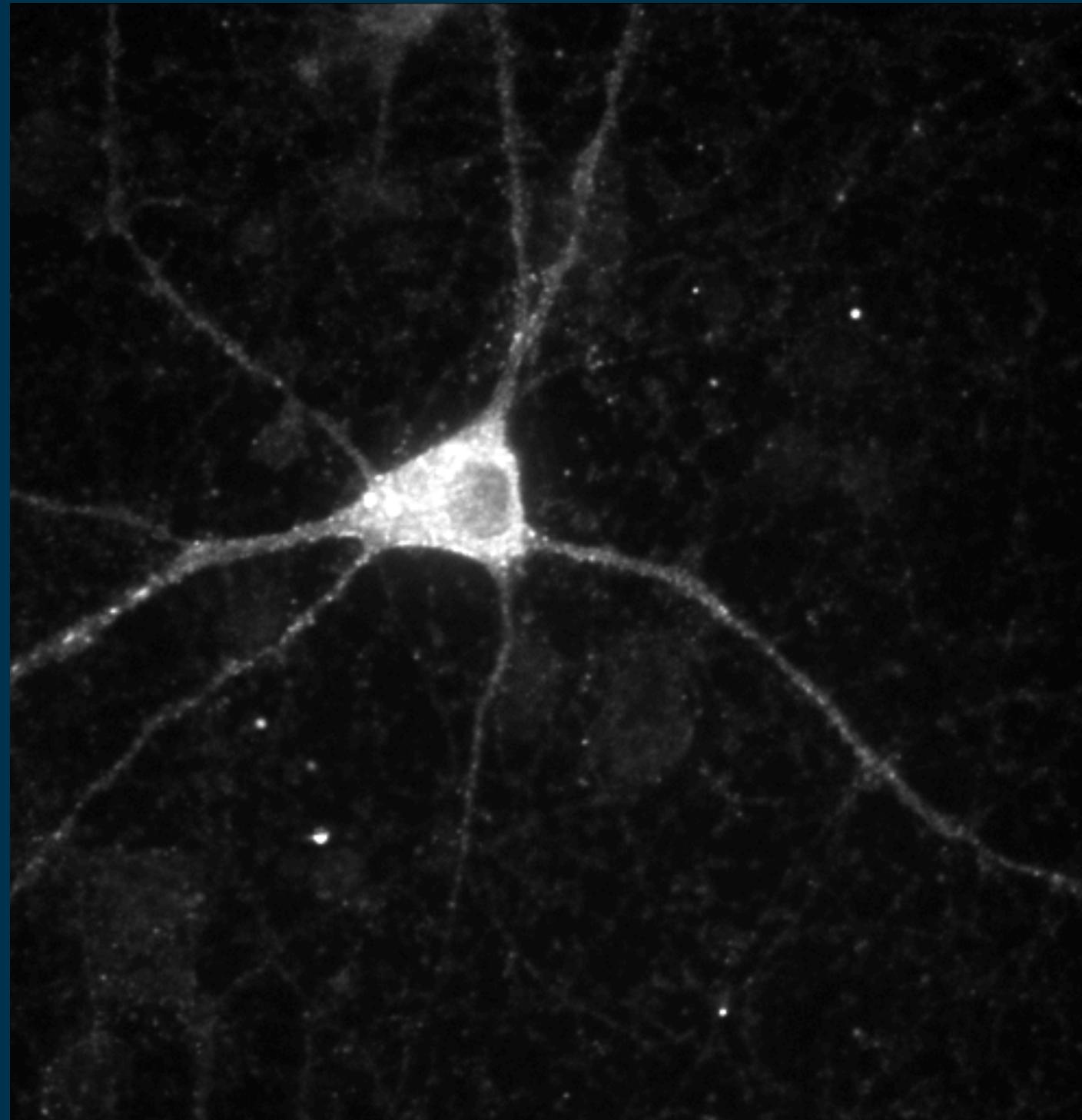


Image from <http://zeiss-campus.magnet.fsu.edu/articles/basics/digitalimaging.html>

# Removing noise / preprocessing the data before segmentation is important

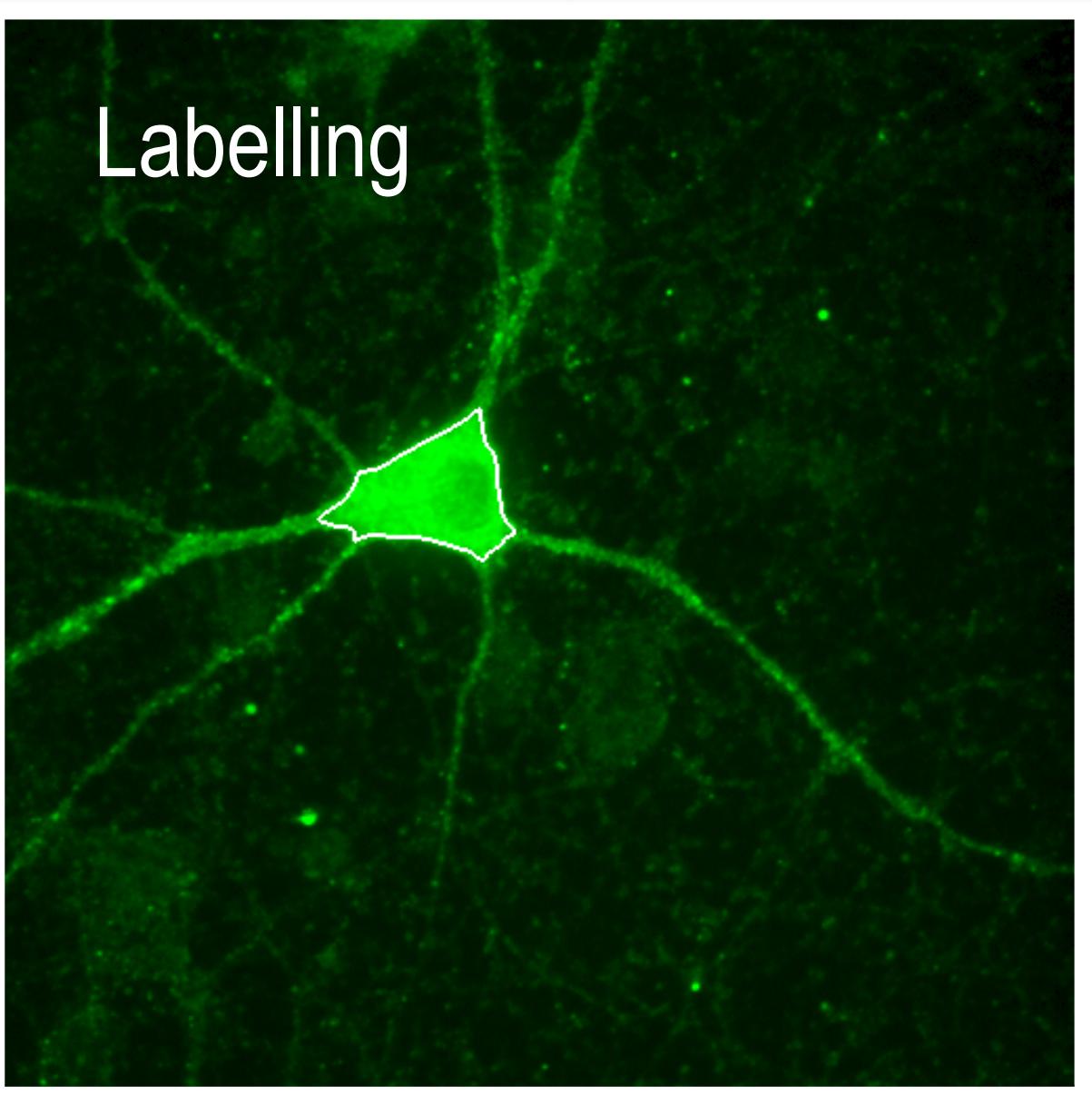
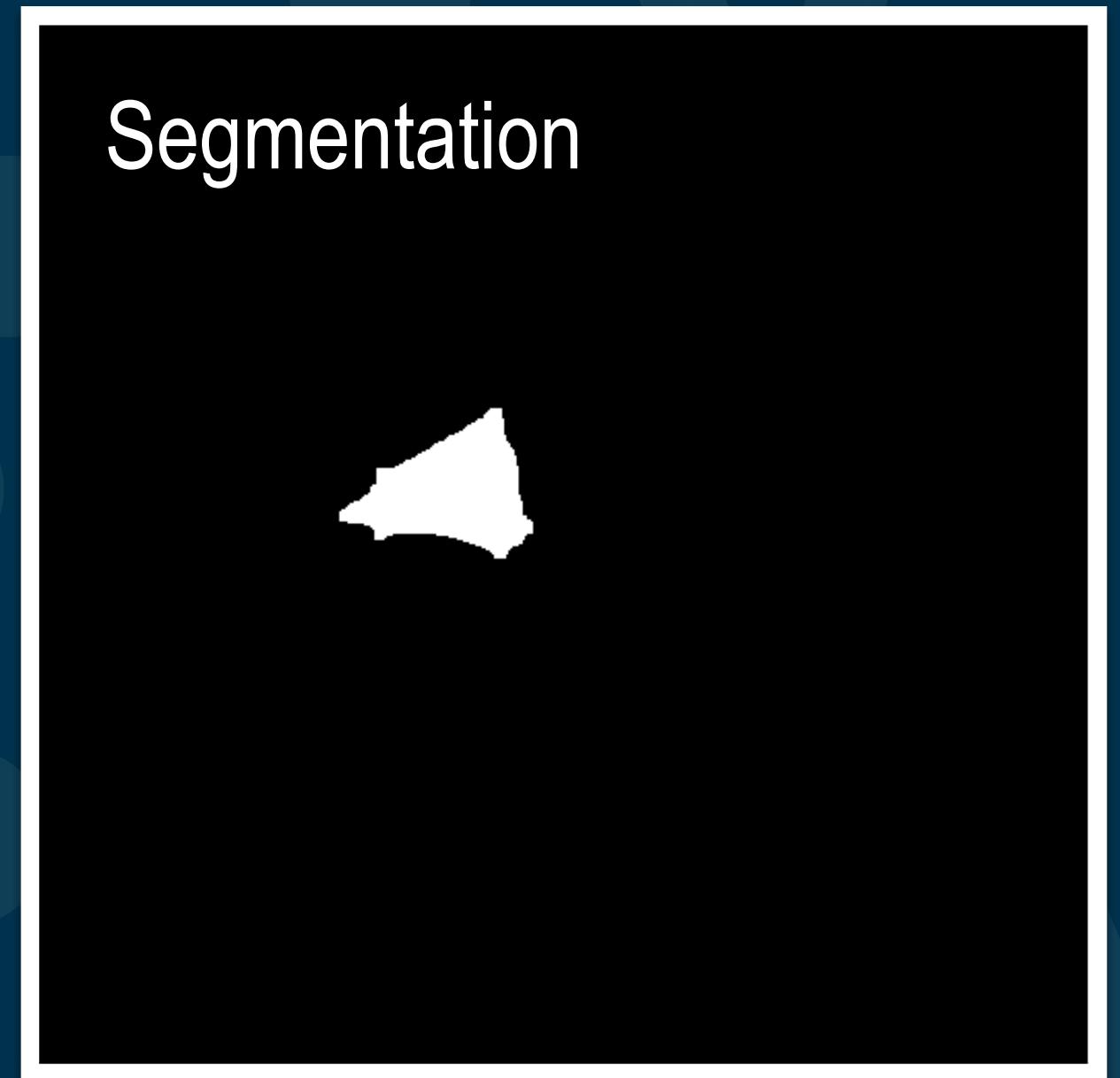
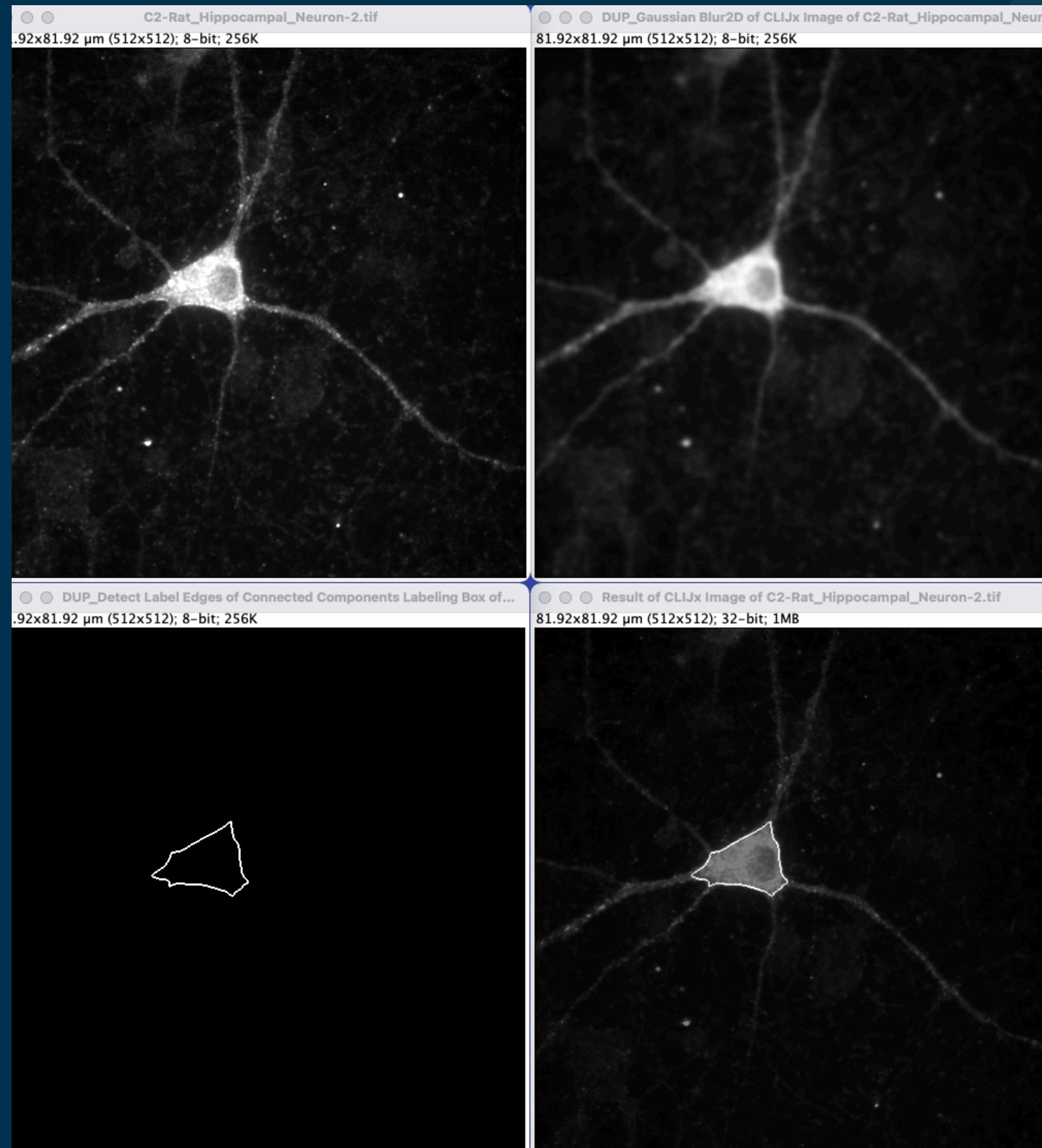


# Removing noise / preprocessing the data before segmentation is important



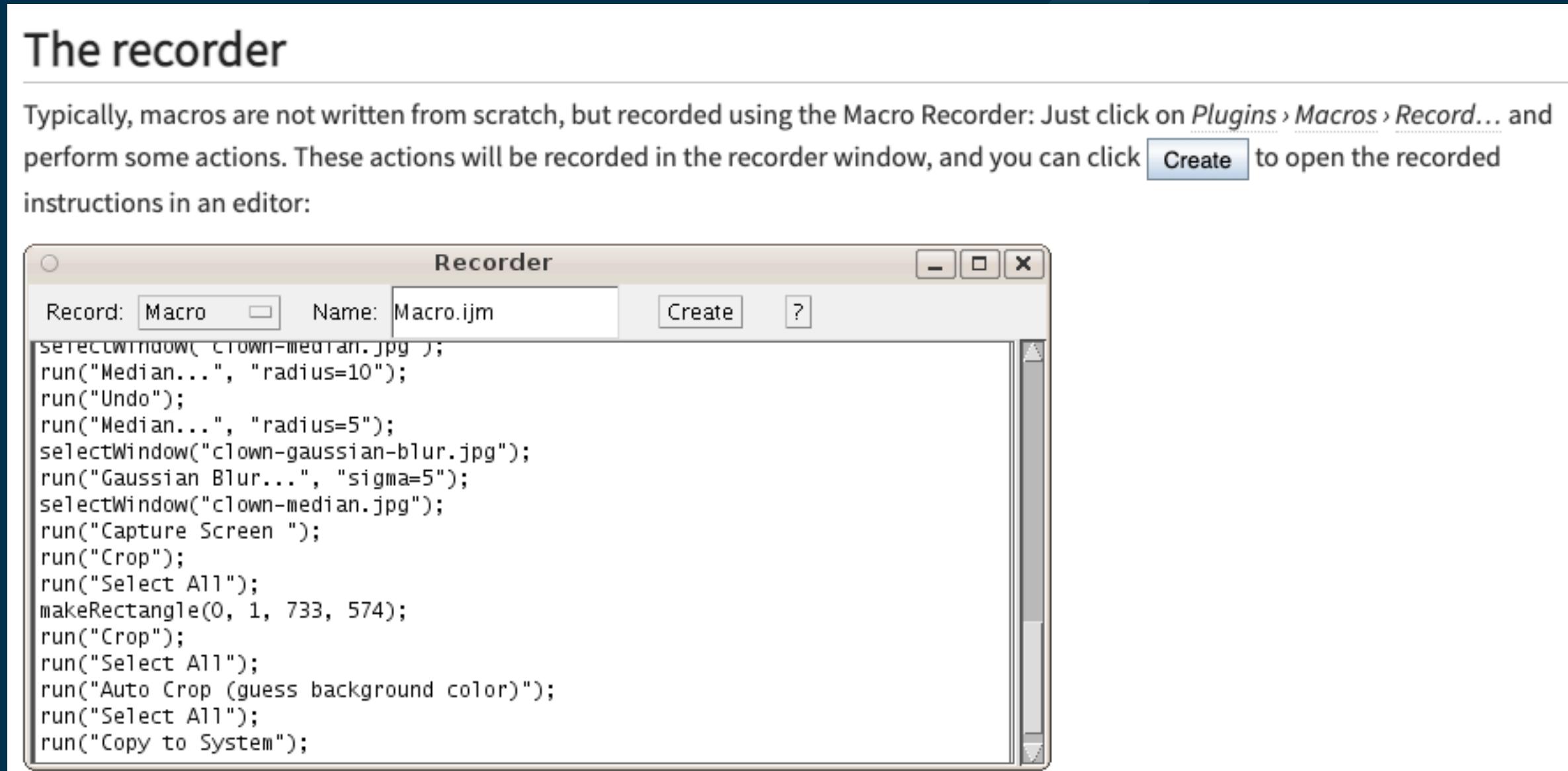
Today we are going to have a hands-on on segmentation and labelling

# Segmentation and labelling



# Macro recorder

Create our first macro



```
macro_v1.ijm
1 input = getTitle();
2 nameOnly = File.nameWithoutExtension;
3 print("Image name without extension= " + nameOnly);
4
5 rename("original");
6
7 run("8-bit");
8 run("Morphological Filters", "operation=[White Top Hat] element=Square radius=2");
9 run("Subtract Background...", "rolling=50");
10
11 run("Find Maxima...", "prominence=10 output=[Maxima Within Tolerance]");
12 //run("Brightness/Contrast...");
13 run("Enhance Contrast", "saturated=0.35");
14
15
16 setAutoThreshold("Otsu dark");
17 //setThreshold(41, 255);
18 setOption("BlackBackground", true);
19 run("Convert to Mask");
20 run("Connected Components Labeling", "connectivity=4 type=[16 bits]");
21 run("glasbey on dark");
22
23
24 run("Duplicate...", " ");
25
26
27
28 run("Find Maxima...", "prominence=10 output=[Maxima Within Tolerance]");
29
30 run("Set Measurements...", "area mean standard min centroid center fit area_fraction redirect=original decimal=3");
31 run("Analyze Particles...", "size=0.0001-1 show=Outlines display exclude clear summarize add");
32
33 // introduce a variable tableName for the results table custom-name
34 tableName = "Results_Stats" + "_" + nameOnly;
35
36 print(tableName);
37 selectWindow("Summary");
38
39 // save the results table to saveDir using saveAs()
40 saveAs("Results", tableName + ".csv");
```

Example script is provided

From <https://imagej.net>

# What if you want to apply these operations to all images in a given directory?

```
1 /* This macro counts the number of plaques
2  * in a folder with multiple 2D images
3  *
4 Author: Miguel Fernandes (IDAF)
5 Date: 10 Sep 2021
6 */
7 /*
8 Usage:
9 There should not be any " " in your directory or file names
10 At the moment working for 2D image data
11
12 Only one channel is expected to perform analysis
13 Keep it consistent across experiments!
14
15 Run this macro and select a folder containing imaging files
16 (inputFolder)
17
18 Select type of input stacks with the format defined in suffix
19 (example .tif files)
20
21 Select output folder to save results for each individual file and summary across all images
22 (outputFolder)
23
24 At the moment only for 2D but in principle could be extended to 3D
25
26 TODO: improve documentation for each step
27
28 */
29
30
31
32 #@ File (label="Select a folder to process", style="directory") inputFolder
33 #@ String (label = "File suffix input folder", value = ".tif") suffix
34 #@ File (label="Select a folder to save results", style="directory") outputFolder
35
36 flist=getFileList(inputFolder);
37 dir=File.getDirectory(inputFolder);
38 print (inputFolder);
39
40
41 print("User selected input folder: " + inputFolder);
42 print("User selected save folder: " + outputFolder);
43
44
45 setBatchMode(true); //batch mode on
```

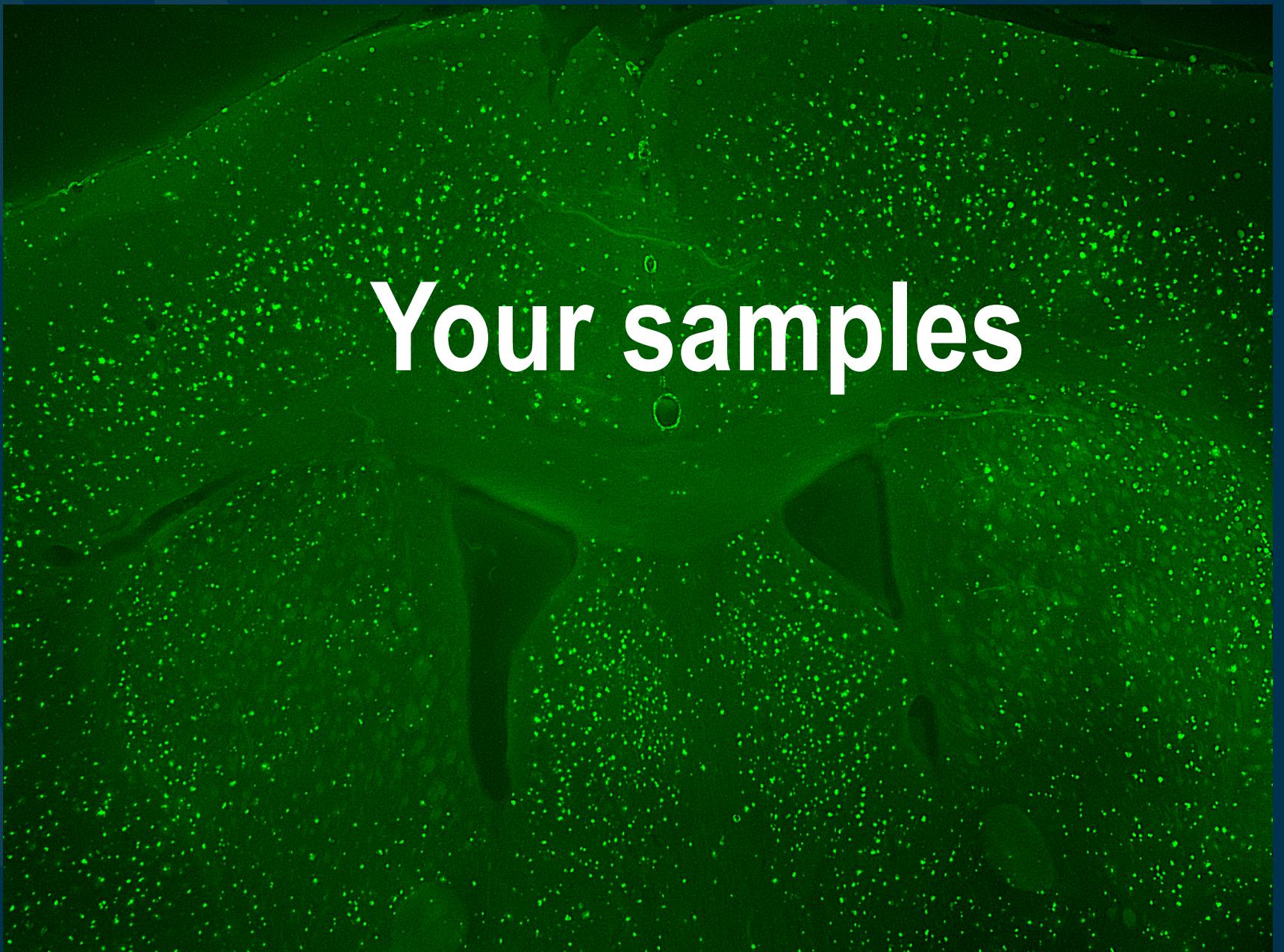
Example script is provided

# Now is time to perform the analysis

## FIJI time!!!



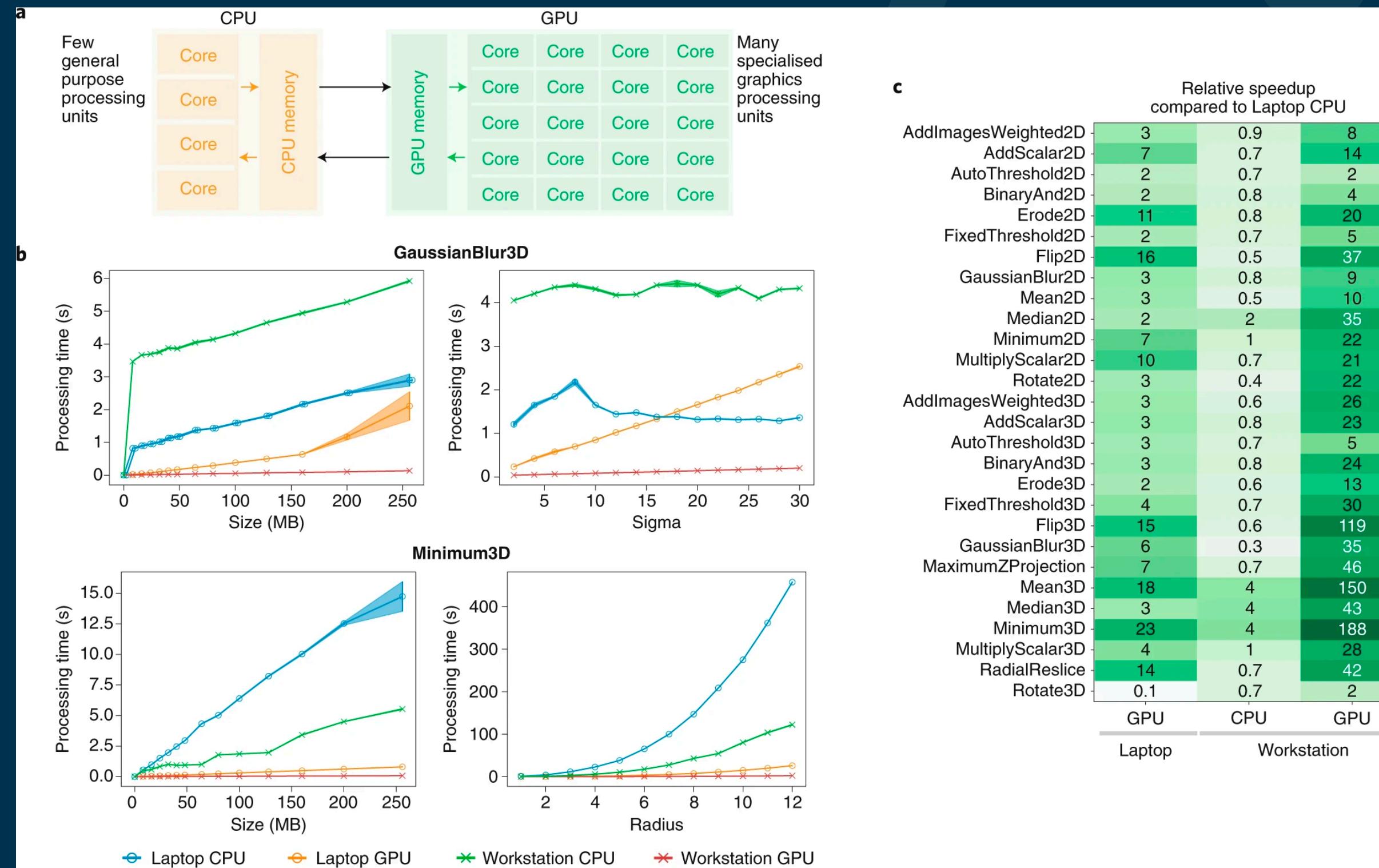
Example script is provided



# What needs to be improved? Ideas?

- Improve preprocessing and thresholds: what would you try?
- Accelerate imaging processing (e.g. GPU)
- Use Machine learning (ML) to classify our ROIs (Weka)
- Plotting the results and perform statistical tests (Python, see Jupyter Notebook)

# GPU Accelerates your image analysis



```

1 // To make this script run in Fiji, please activate
2 // the clij and clij2 update sites in your Fiji
3 // installation. Read more: https://clij.github.io
4
5 // Generator version: 2.5.1.1
6
7 // Init GPU
8 run("CLIJ2 Macro Extensions", "cl_device=AMD");
9
10 // Load image from disc
11 open("/Users/fernandesm/Documents/5x FAD gr3 5.tif");
12
13
14 rename("original");
15
16 image_1 = getTitle();
17 Ext.CLIJ2_pushCurrentZStack(image_1);
18 // The following auto-generated workflow is made for processing a 2D or 3D dataset.
19 // For processing multiple channels or time points, you need to program a for-loop.
20 // You can learn how to do this online: https://www.youtube.com/watch?v=u1Sq-x5\_in4
21
22 // Copy
23 Ext.CLIJ2_copy(image_1, image_2);
24 Ext.CLIJ2_release(image_1);
25
26 Ext.CLIJ2_pull(image_2);
27
28 // Copy
29 Ext.CLIJ2_copy(image_2, image_3);
30 Ext.CLIJ2_release(image_2);
31
32 Ext.CLIJ2_pull(image_3);
33
34 // Top Hat Box
35 radiusX = 10.0;
36 radiusY = 10.0;
37 radiusZ = 10.0;
38 Ext.CLIJ2_topHatBox(image_3, image_4, radiusX, radiusY, radiusZ);
39 Ext.CLIJ2_release(image_3);
40
41 Ext.CLIJ2_pull(image_4);
42
43 // Threshold Otsu
44 Ext.CLIJ2_thresholdOtsu(image_4, image_5);
45 Ext.CLIJ2_release(image_4);
46

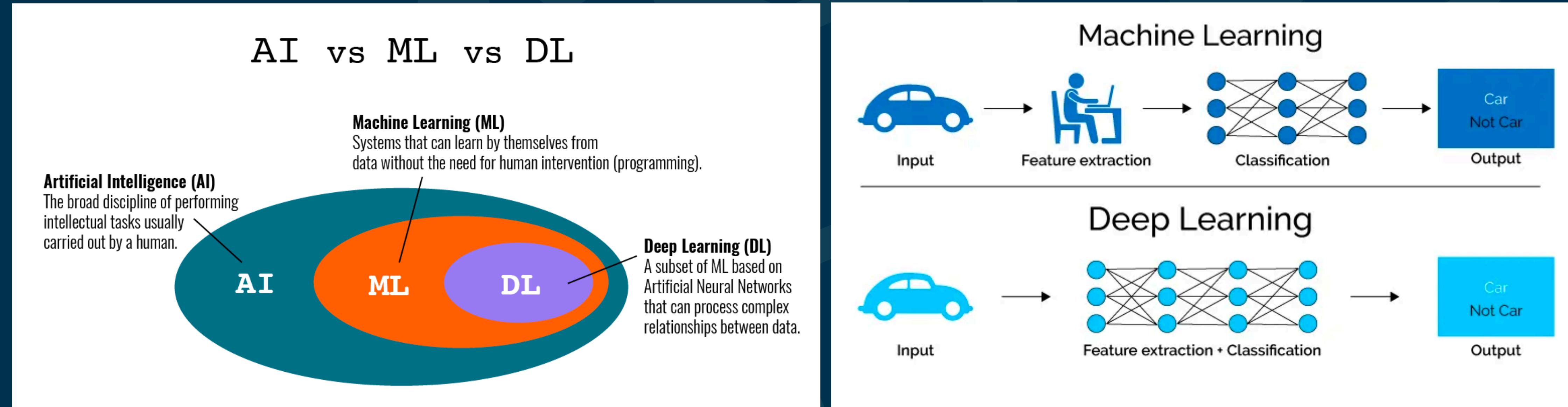
```

Haase et al 2020

Licence CC BY 4.0: <https://creativecommons.org/licenses/by/4.0/legalcode>

Example script is provided

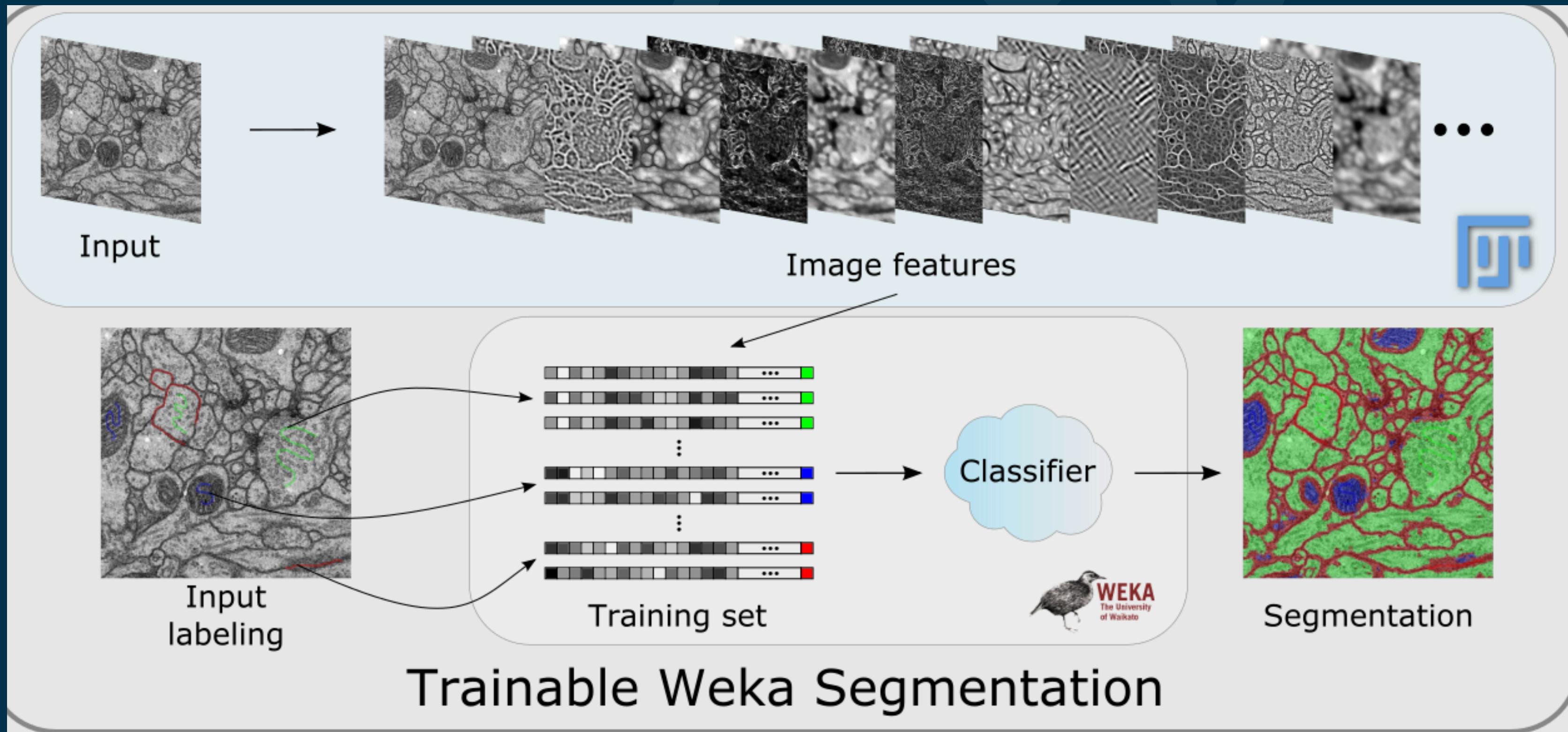
# Artificial intelligence vs Machine learning vs Deep learning



<https://mytechme.com/artificial-intelligence-vs-machine-learning-vs-deep-learning/>

<https://semiengineering.com/deep-learning-spreads/>

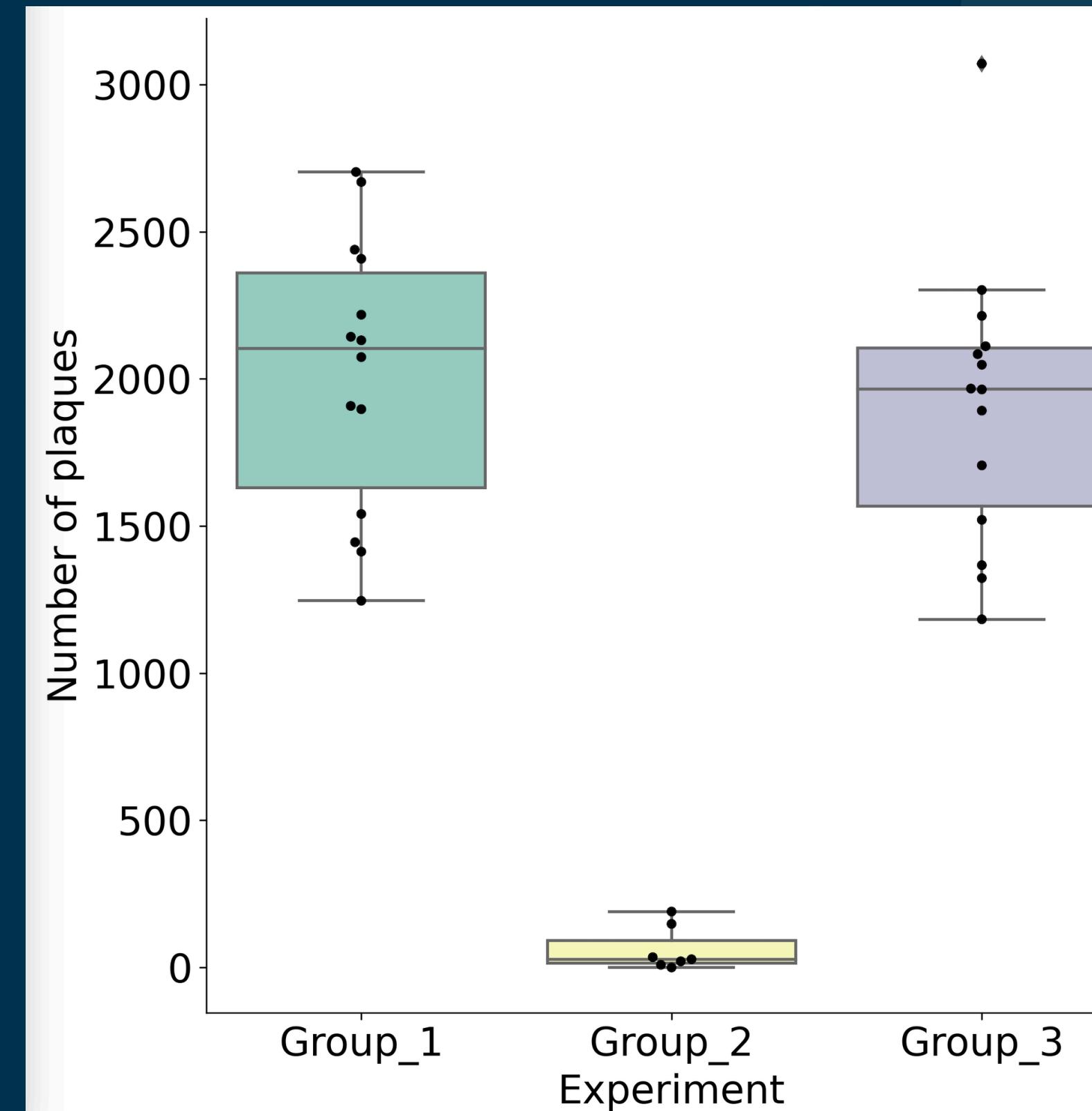
# Trainable Weka Segmentation



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

# Visualization of results and statistical analysis (Python and Jupyter Notebook)

Pandas, Matplotlib, Seaborn



The Mann–Whitney U test (also called Wilcoxon rank-sum test) is a non-parametric test of the null hypothesis

```
pg.mwu(summary_df_gr1.Count, summary_df_gr2.Count)
```

U-val	alternative	p-val	RBC	CLES
MWU	two-sided	0.000017	-1.0	1.0

```
pg.mwu(summary_df_gr1.Count, summary_df_gr3.Count)
```

U-val	alternative	p-val	RBC	CLES
MWU	two-sided	0.370264	-0.204082	0.602041

```
pg.mwu(summary_df_gr2.Count, summary_df_gr3.Count)
```

U-val	alternative	p-val	RBC	CLES
MWU	two-sided	0.000017	1.0	0.0

**P-values correction for multiple comparisons.**

```
'''Benjamini–Hochberg FDR correction of an array of p-values'''
```

```
pvals = [0.000017, 0.370264, 0.000017]
reject, pvals_corr = pg.multicomp(pvals, method='fdr_bh')
print(reject, pvals_corr)
```

```
[ True False  True] [2.55000e-05 3.70264e-01 2.55000e-05]
```

Pingouin can be used for statistical analysis

Notebook with analysis is provided



Thank you!!!

# Additional Resources

- [https://git.mpi-cbg.de/rhaase/lecture applied bioimage analysis 2020](https://git.mpi-cbg.de/rhaase/lecture_applied_bioimage_analysis_2020)
- <https://montpellierressourcesimagerie.github.io/mri-workshop-machine-learning/>
- <https://imagej.nih.gov/ij/docs/examples/index.html>
- <https://petebankhead.gitbooks.io/imagej-intro/content/>
- Bioimage Data Analysis Workflows book: <https://www.springer.com/gp/book/9783030223854>