

Bio-image Analysis using Large Language Models

Robert Haase

Funded by



Bundesministerium
für Bildung
und Forschung

SACHSEN



Diese Maßnahme wird gefördert durch die Bundesregierung
aufgrund eines Beschlusses des Deutschen Bundestages.
Diese Maßnahme wird mitfinanziert durch Steuermittel auf
der Grundlage des von den Abgeordneten des Sächsischen
Landtags beschlossenen Haushaltes.

Quick survey

Question	Number of raised hands
Who has tried ChatGPT, Dall-E or other large language model based tools before?	
Who is using them on a daily basis?	
Who knows how they work under the hood (roughly)?	



Download slides:
<https://doi.org/10.5281/zenodo.10784549>

Bio-image Analysis

.... is about assembling [image] processing-tools in workflows to gain quantitative measurements and new insights.

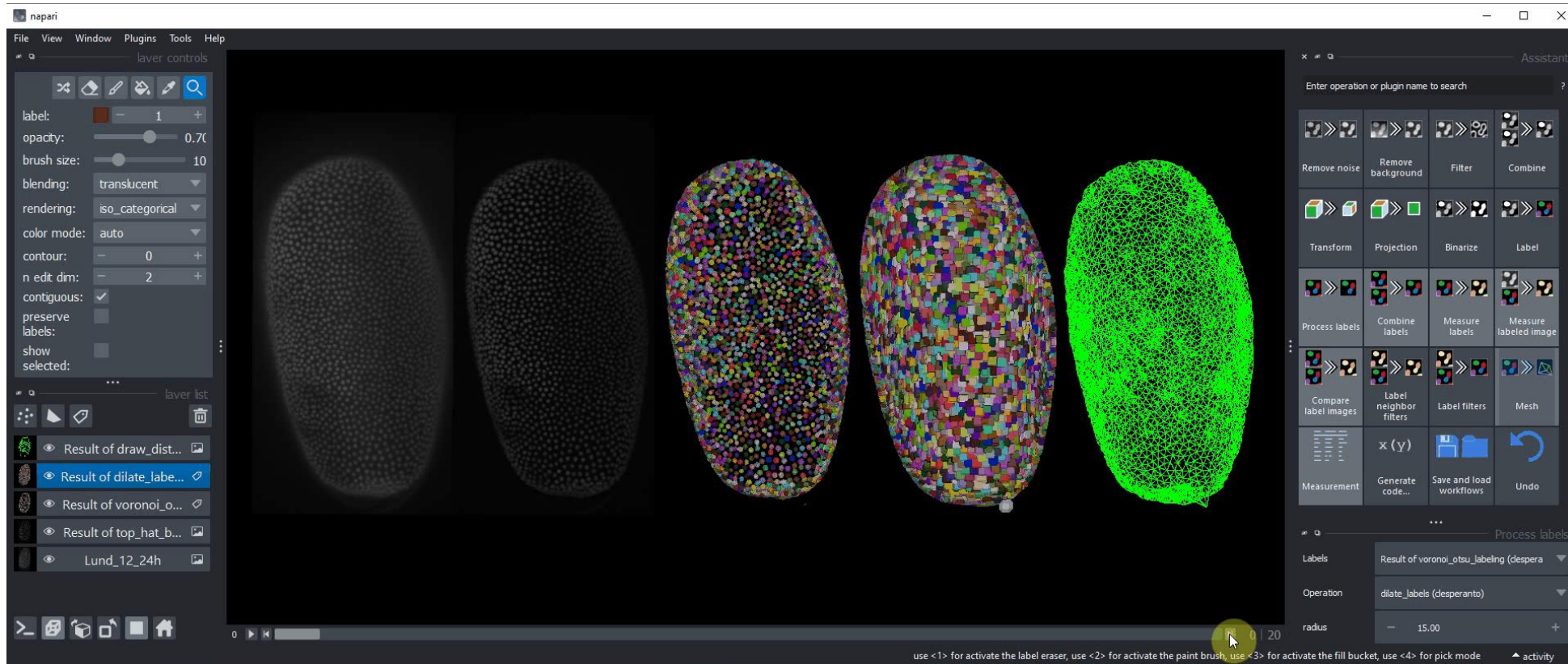


Image data source: Daniela Vorkel, Myers lab, MPI-CBG/CSBD

Bio-image Analysis

... programming languages vary over time, according to my experience



2008-2015
C++, Insight ToolKit



2015-2021
**Java, ImageJ macro,
Fiji**



2019-today
**Python, napari,
Jupyter**

Challenge: Data load / processing time

- State-of-the-art software for more than 20 years: ImageJ / Fiji

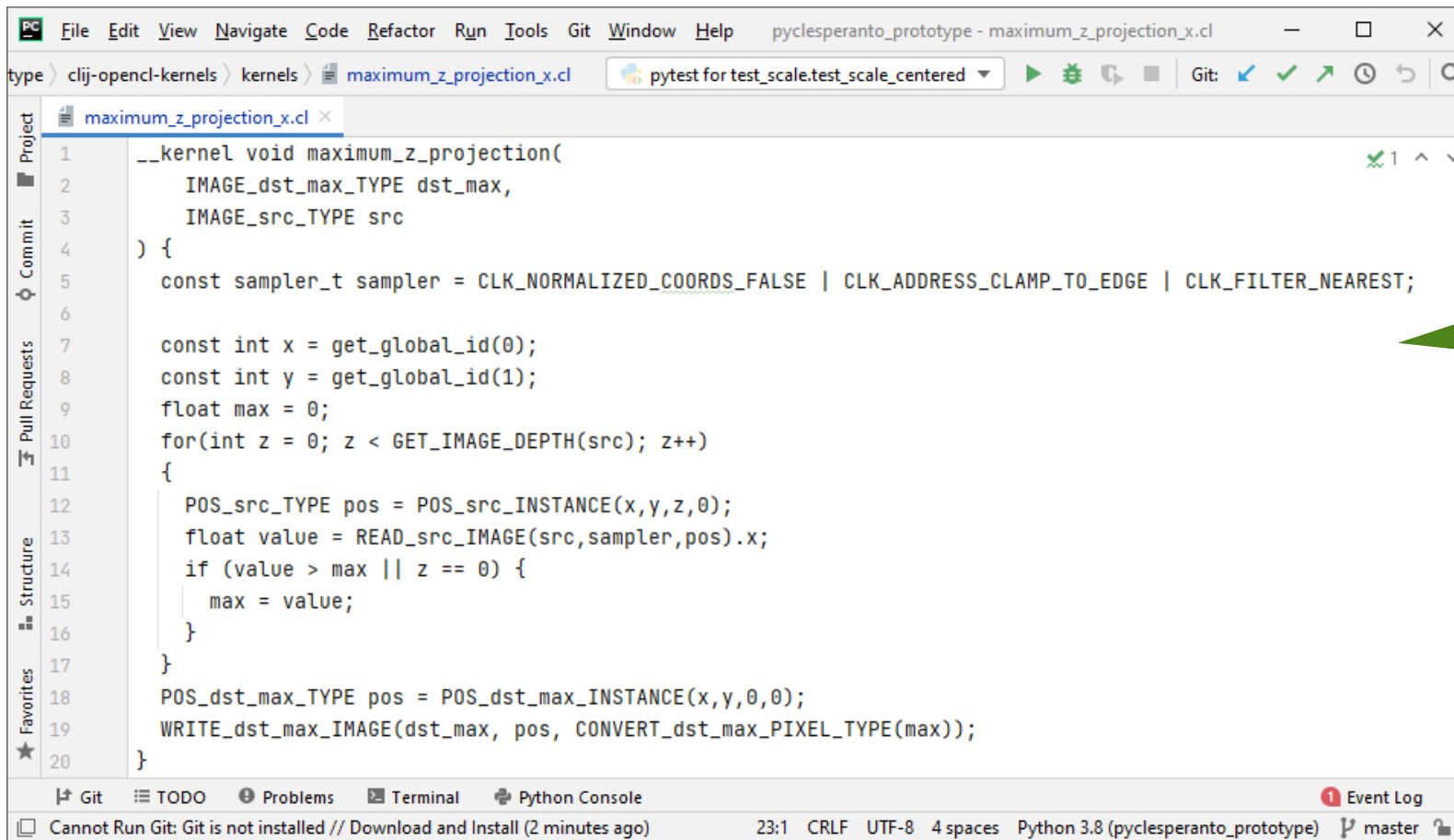


2x

Image data source: Daniela Vorkel, Myers lab, MPI-CBG/CSBD

Challenge: Programming languages

GPU-acceleration? Learn the Open Computing Language (OpenCL)!



```
1  __kernel void maximum_z_projection(  
2      IMAGE_dst_max_TYPE dst_max,  
3      IMAGE_src_TYPE src  
4  ) {  
5      const sampler_t sampler = CLK_NORMALIZED_COORDS_FALSE | CLK_ADDRESS_CLAMP_TO_EDGE | CLK_FILTER_NEAREST;  
6  
7      const int x = get_global_id(0);  
8      const int y = get_global_id(1);  
9      float max = 0;  
10     for(int z = 0; z < GET_IMAGE_DEPTH(src); z++)  
11     {  
12         POS_src_TYPE pos = POS_src_INSTANCE(x,y,z,0);  
13         float value = READ_src_IMAGE(src,sampler,pos).x;  
14         if (value > max || z == 0) {  
15             max = value;  
16         }  
17     }  
18     POS_dst_max_TYPE pos = POS_dst_max_INSTANCE(x,y,0,0);  
19     WRITE_dst_max_IMAGE(dst_max, pos, CONVERT_dst_max_PIXEL_TYPE(max));  
20 }
```

Maximum intensity projection along Z

GPU-accelerated image processing

Speedup depends on operation, image size, parameters, hardware,



vs.



 Workstation CPU

2x Intel Xeon Silver
4110

 Laptop CPU

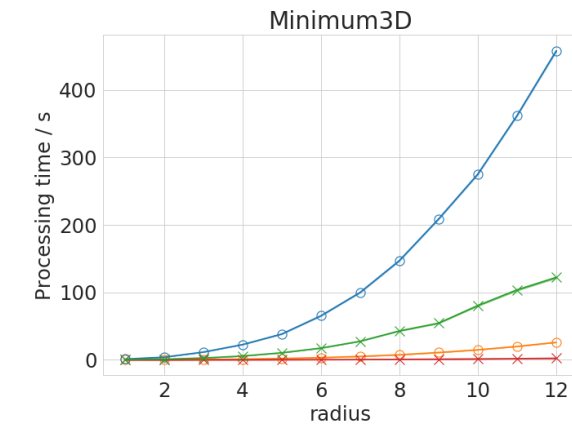
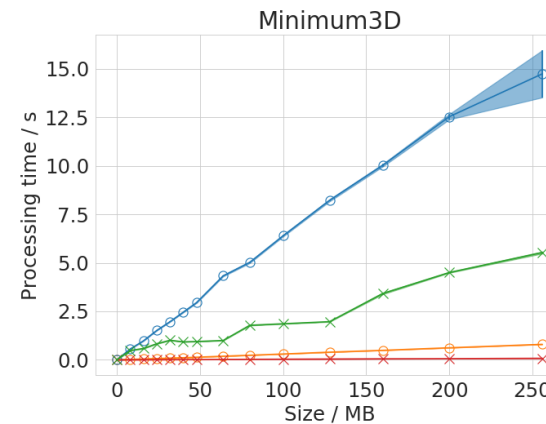
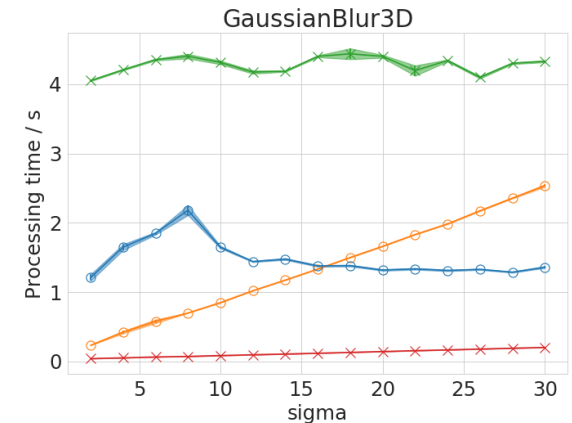
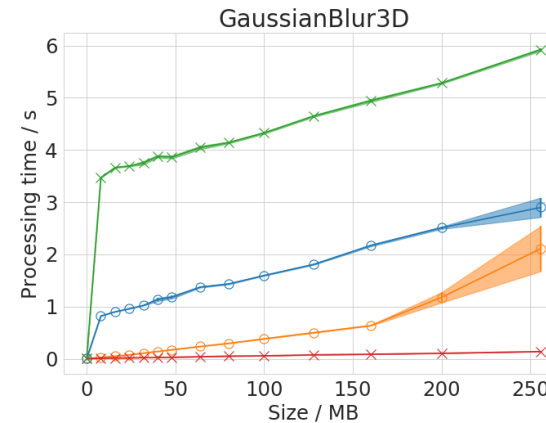
Intel Core i7-8650U

 Workstation GPU

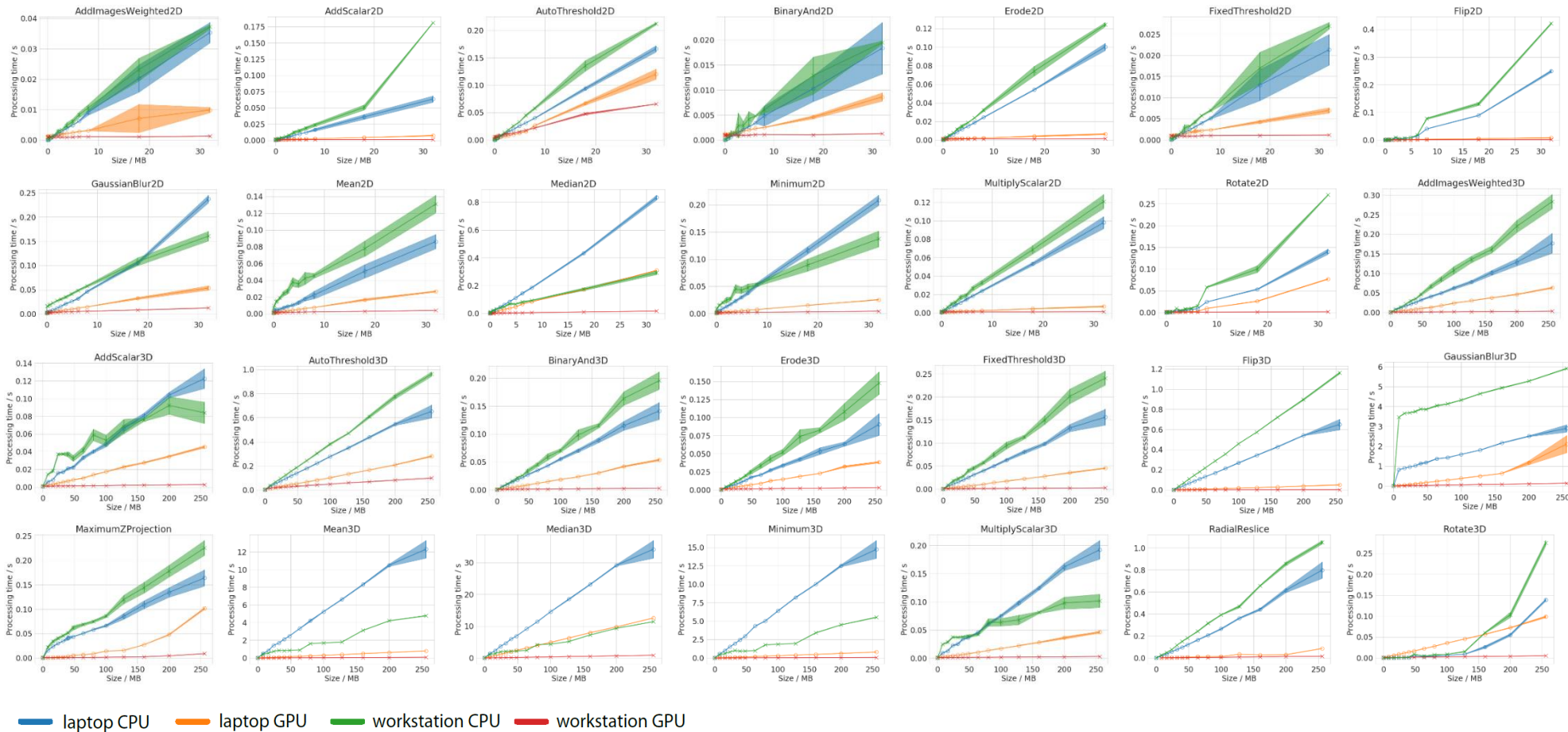
Nvidia Quadro
P6000

 Laptop GPU

Intel UHD 620 GPU



GPU-accelerated image processing



Challenge: User-friendly GPU-acceleration

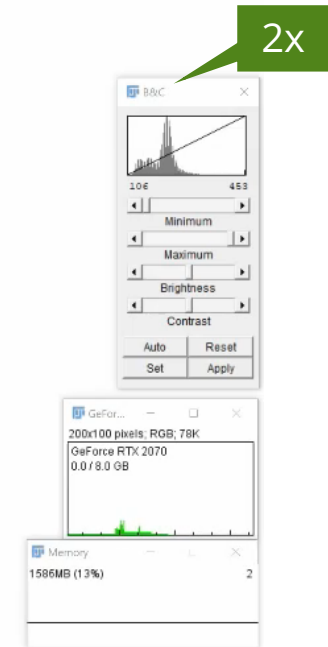
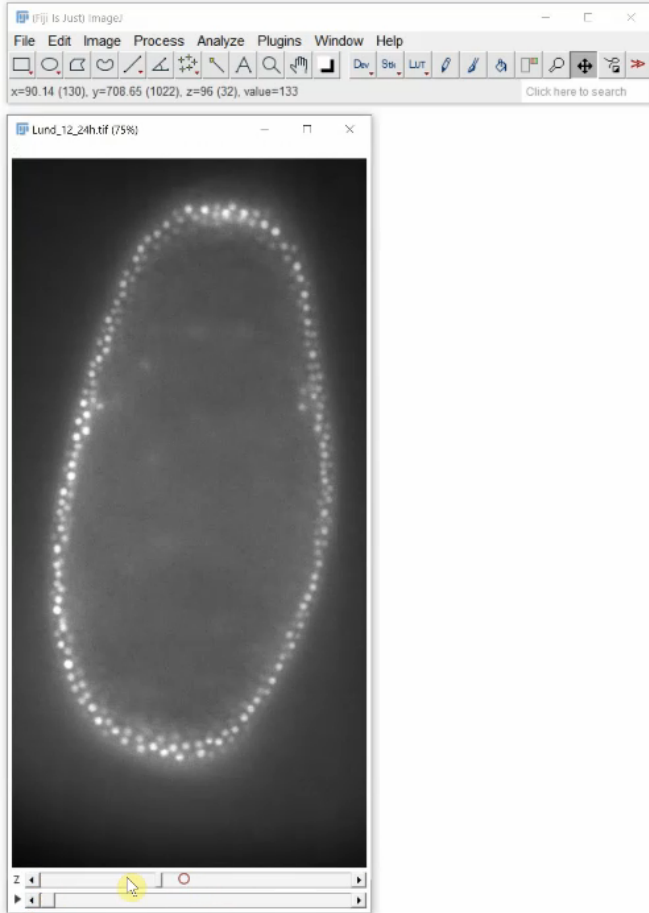
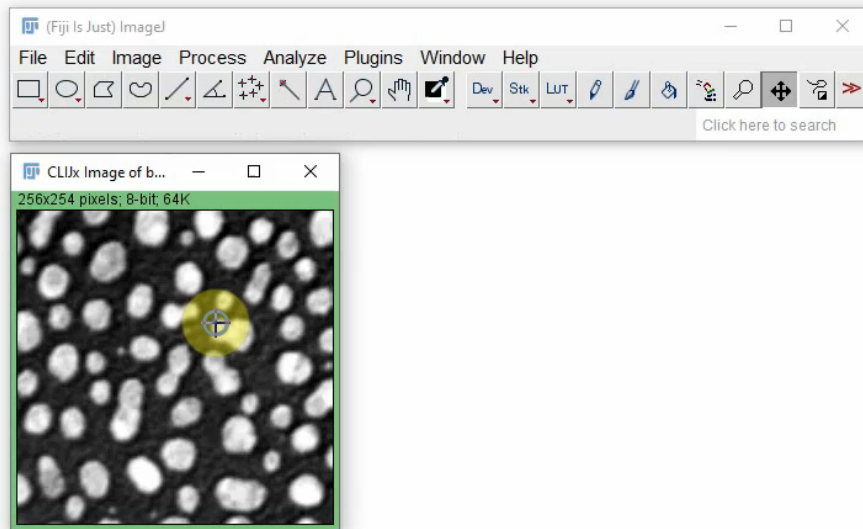


Image data source: Daniela Vorkel, Myers lab, MPI-CBG/CSBD

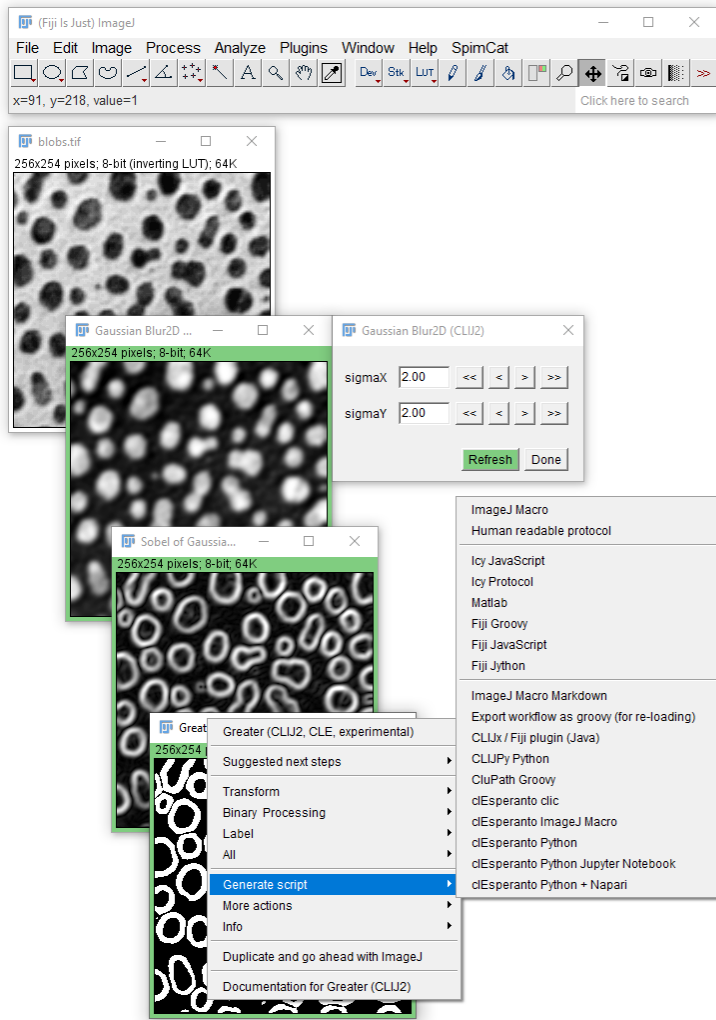
Challenge: Workflow-generation + understanding



Challenge: Code generation

IJ Macro

Python

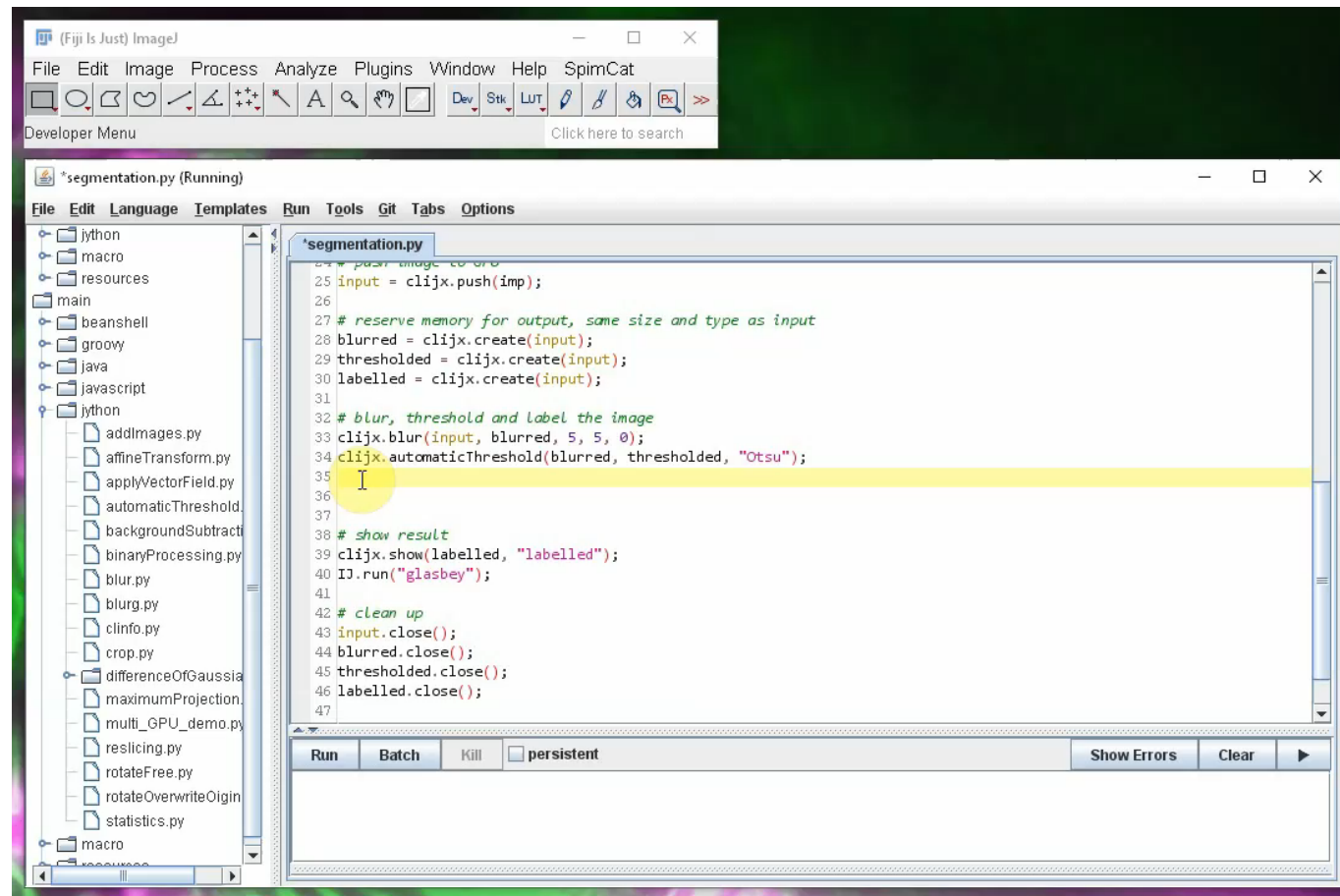


```
new1.ijm
File Edit Language Templates Run Tools Tabs Options
1 // To make this script run in Fiji, please activate
2 // the clij and clij2 update sites in your Fiji
3 // installation. Read more: https://clij.github.io
4
5 // Init GPU
6 run("CLIJ2 Macro Extensions", "cl_device=");
7
8 // Load image from disc
9 open("C:/structure/data/New folder/blobs.tif");
10 image1 = getTitle();
11 Ext.CLIJ2_push(image1);
12
13 // Gaussian Blur2D
14 sigmaX = 2;
15 sigmaY = 2;
16 Ext.CLIJ2_gaussianBlur2D(image1, image2, sigmaX, sigmaY);
17 Ext.CLIJ2_release(image1);
18
19 Ext.CLIJ2_pull(image2);
20
21 // Sobel
22 Ext.CLIJ2_sobel(image2, image3);
23
24 Ext.CLIJ2_pull(image3);
25
26 // Greater
27 Ext.CLIJ2_greater(image3, image2, image4);
28 Ext.CLIJ2_release(image3);
29 Ext.CLIJ2_release(image2);
30
31 Ext.CLIJ2_pull(image4);
32 Ext.CLIJ2_release(image4);
33
Run Batch [ ] persistent Show Errors Clear ▶
```

```
new2.py
File Edit Language Templates Run Tools Tabs Options
14 # Stay tuned and check out http://clesperanto.net to learn more.
15
16 import pyclesperanto_prototype as cle
17 from tifffile import imread
18
19 import numpy as np
20 import matplotlib
21 import matplotlib.pyplot as plt
22
23
24 # Load image
25 image = imread("C:/structure/data/New folder/blobs.tif")
26
27 # Push blobs.tif to GPU memory
28 image1 = cle.push_xyx(image)
29
30 # Gaussian Blur2D
31 image2 = cle.create_like(image1);
32 source = null
33 destination = image2
34 sigmaX = 2
35 sigmaY = 2
36 cle.gaussian_blur(source, destination, sigmaX, sigmaY)
37 # show result
38 plt.imshow(image2)
39 plt.show()
40
41
42 # Sobel
43 image3 = cle.create_like(image2);
44 source = image2
45 destination = image3
46 cle.sobel(source, destination)
47 # show result
48 plt.imshow(image3)
49 plt.show()
50
51
52 # Greater
53 image4 = cle.create_like(image3);
54 source1 = image3
55 source2 = image2
56 destination = image4
57 cle.greater(source1, source2, destination)
58 # show result
59 plt.imshow(image4)
60 plt.show()
61
62
Run Batch [ ] persistent Show Errors Clear ▶
```

Challenge: Code generation

The clesperanto project



The screenshot shows the Fiji (ImageJ) interface with a Python script editor open. The script, named 'segmentation.py', is being executed. The code performs the following steps:

```
25 input = clijx.push(input);
26
27 # reserve memory for output, same size and type as input
28 blurred = clijx.create(input);
29 thresholded = clijx.create(input);
30 labelled = clijx.create(input);
31
32 # blur, threshold and label the image
33 clijx.blur(input, blurred, 5, 5, 0);
34 clijx.automaticThreshold(blurred, thresholded, "Otsu");
35
36
37
38 # show result
39 clijx.show(labelled, "labelled");
40 II.run("glasbey");
41
42 # clean up
43 input.close();
44 blurred.close();
45 thresholded.close();
46 labelled.close();
47
```



With Stéphane Rigaud
Image Analysis Hub, C2RT,
Institut Pasteur Paris

Challenge: Code generation

The clesperanto project

The screenshot displays a Jupyter Notebook environment with the following code in a cell:

```
labelled_without_edges = ...
% blur, threshold and label the image
42 clijx.blur(input, blurred, 5, 5, 0);
43 clijx.automatic_threshold(blurred, thresholded, "Otsu");
44 clijx.connected_components_labeling(thresholded, labelled);
45 clijx.exclude_labels_on_edges(labelled, labelled_without_edges);
46
47
48
49
50
51 result = clijx.merge(labelled_without_edges, ...
52 number_of_features, ...
53 lookuptable);
54 subplot(1,2,1);
55
56
57 input.close();
58 blurred.close();
59 thresholded.close();
60 labelled.close();
61
62
63
```

A tooltip for the `exclude_labels_on_edges` function is shown, with the following text:

exclude_labels_on_edges
Removes all labels from a label map which touch the edges. Remaining label elements are renumbered afterwards.

Parameters:
ClearCLBuffer label_map_input,
ClearCLBuffer label_map_destination



With Stéphane Rigaud
Image Analysis Hub, C2RT,
Institut Pasteur Paris

Napari

Viewer controls
(Napari core)

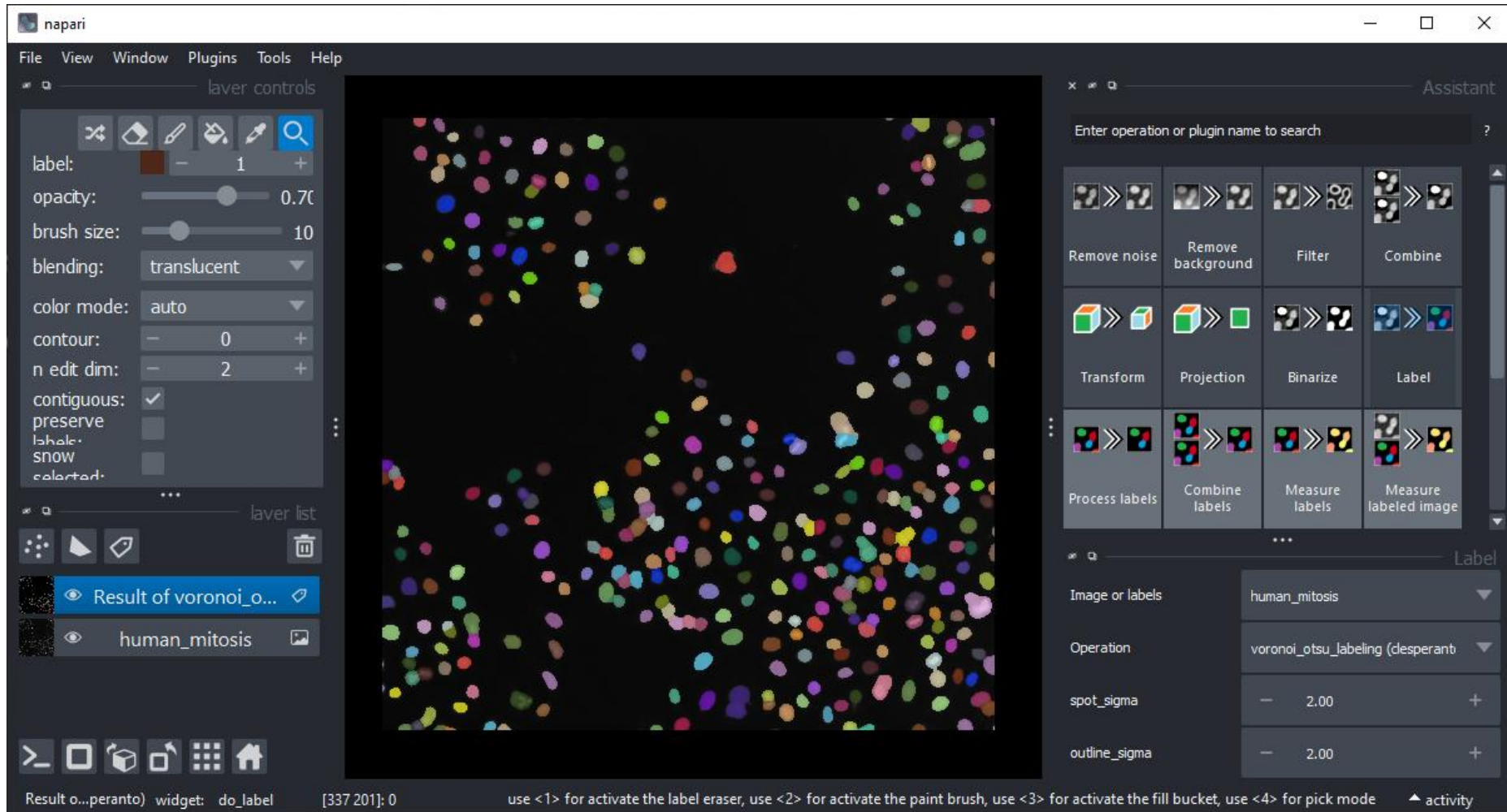
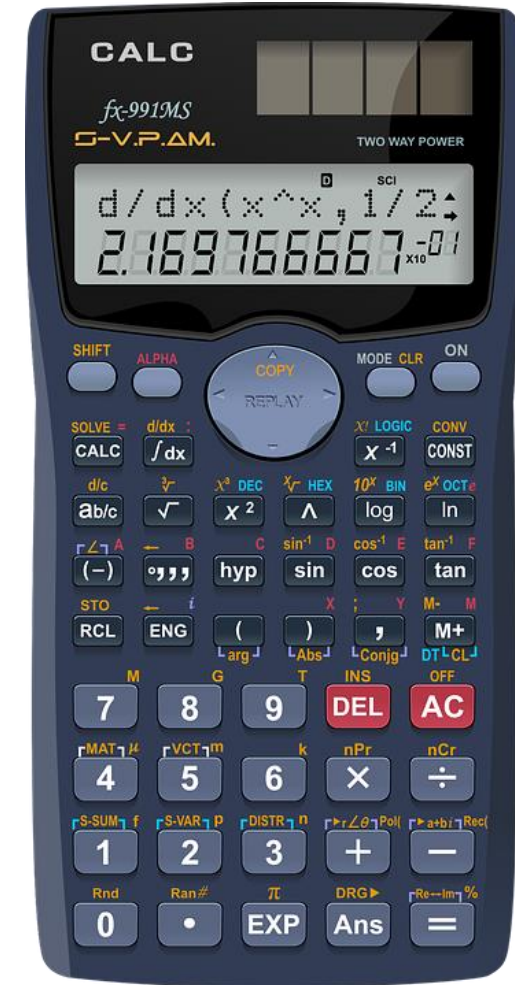
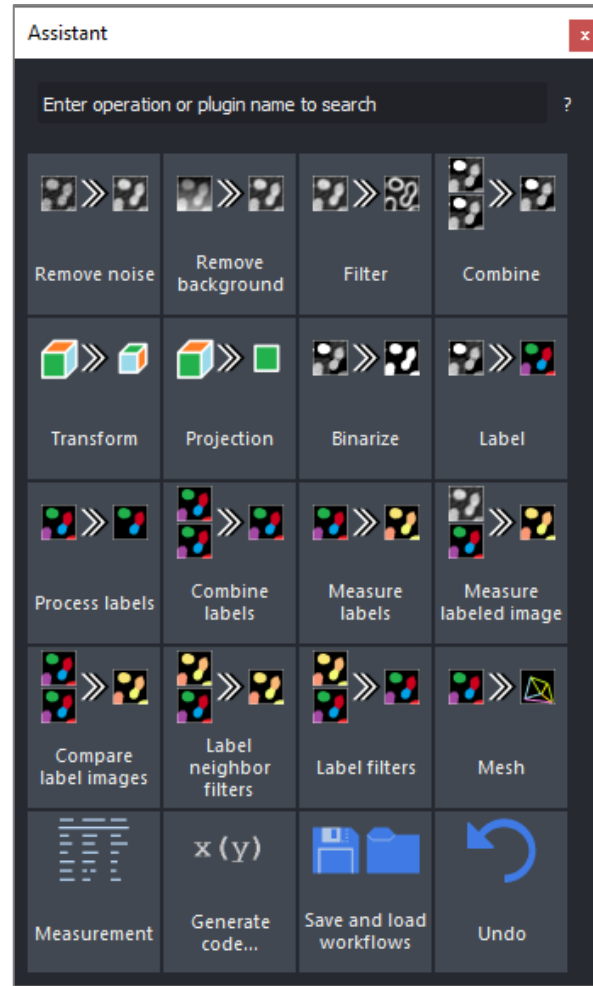


Image Analysis Plugins
(Napari Assistant)

The Napari Assistant

A pocket-calculator-like interface to build image analysis workflows

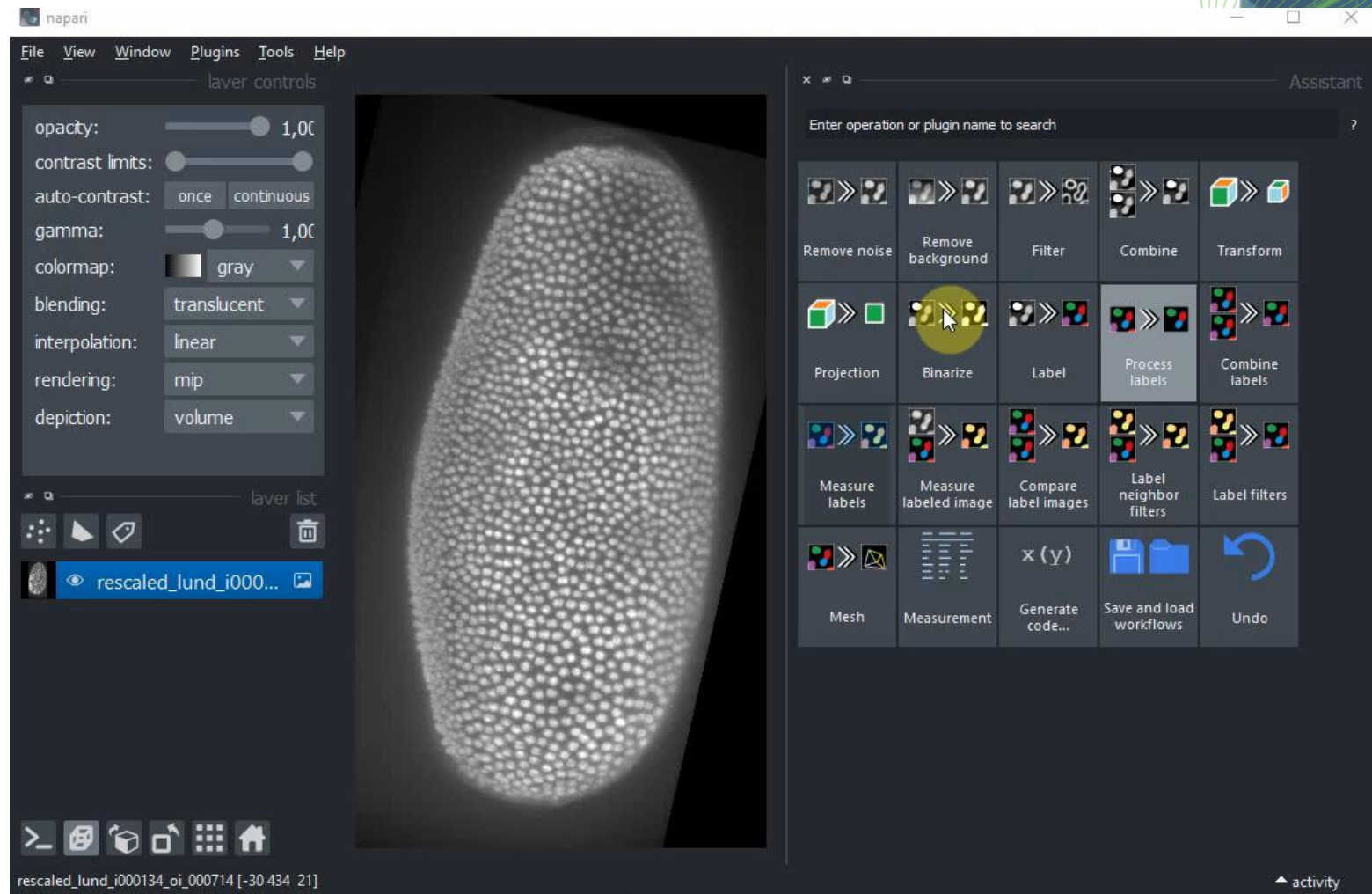


The Napari Assistant

- Classical image processing operations + advanced tools
- Saving&loading supported
- Undo [redo]
- Hints for next steps



Ryan Savill
(now at MPI-CBG) @RyanSavill4



Versatile data types

- Images, labels, tables, surface meshes, ...

The image shows several overlapping screenshots of the Napari software interface. The central focus is on different data types supported by the application:

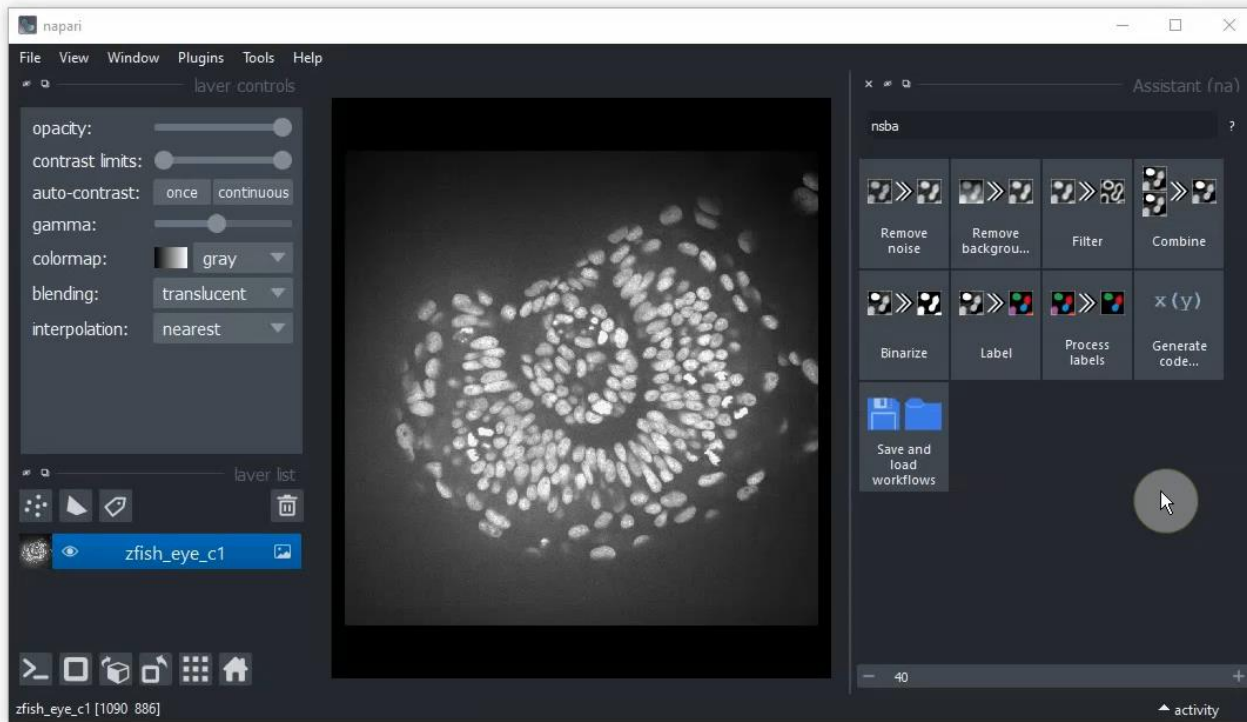
- Image data:** A screenshot showing a multi-channel microscopy image with various colored channels (green, red, blue, purple).
- Label images:** A screenshot showing the same image with segmented regions, where different colors represent different object classes.
- Tabular data:** A screenshot of a 'Properties of Result of connected_component_labeling' dialog box, which displays a table of statistical data for each labeled component.
- Surface meshes:** A screenshot showing a 3D surface mesh of a biological structure, colored by curvature. A 'Surface quality table' is also visible, listing various quality metrics.

label	maximum	mean	median	minimum
1	232.0	190.8545034642...	200.0	128.0
2	224.0	179.2864864864...	184.0	128.0
3	248.0	205.6170212765...	208.0	128.0
4	248.0	217.3271889400...	232.0	128.0
5	248.0	212.1425576519...	224.0	128.0
6	248.0	204.2947368421...	216.0	128.0
7	200.0	161.4814814814...	168.0	128.0

quality.MAX_ANGL	Quality.AREA	TTTED_CURVATU
90.13655098348...	29.99738909272...	0...
87.26057002937...	46.08704588352...	0...
81.7592704258664...	35.88633782170...	0...
93.34386099440...	22.88729572314...	0...
77.4352879333003...	29.95234659361...	0...
78.3220788968983...	41.15354134223...	0...
85.78147877259...	43.18364399753...	0...
89.7507378421122...	21.84920250054...	0...
93.90693966410...	36.67595405504...	0...

Napari – Code generation

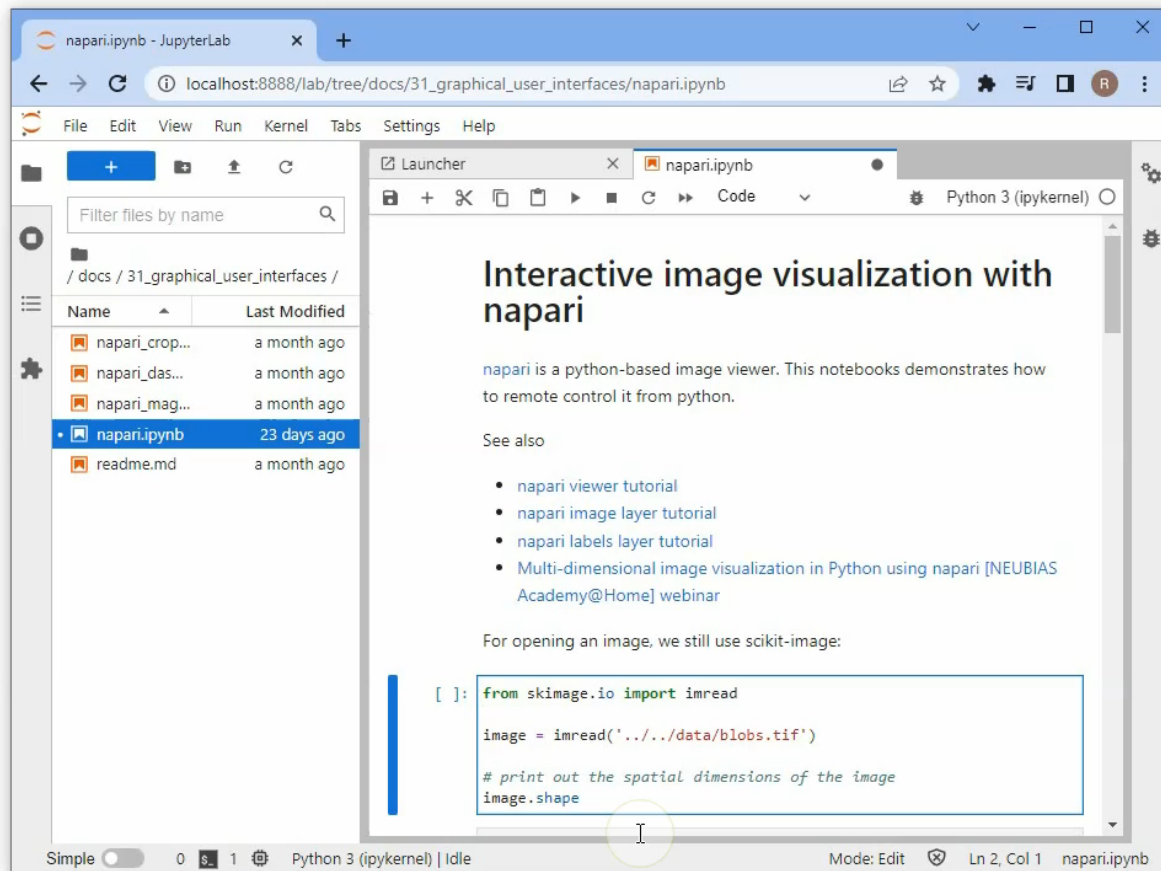
Design your image processing workflow using the Assistant and generate code!



Ryan Savill
(now at MPI-CBG)
@RyanSavill4

Napari – Using via Python / Jupyter Notebooks

Smooth interaction between viewer and code



The screenshot shows a JupyterLab browser window at localhost:8888. The left sidebar displays a file explorer for the directory / docs / 31_graphical_user_interfaces /, listing files like napari_crop..., napari_das..., napari_mag..., napari.ipynb (selected), and readme.md. The main area shows a notebook with the following content:

Interactive image visualization with napari

napari is a python-based image viewer. This notebooks demonstrates how to remote control it from python.

See also

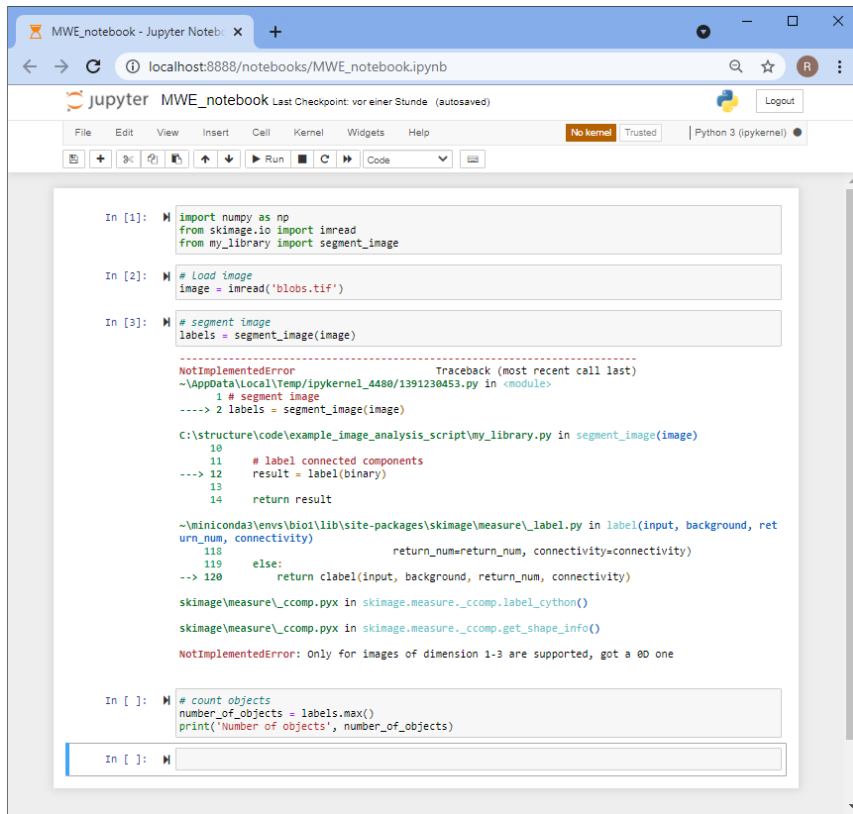
- [napari viewer tutorial](#)
- [napari image layer tutorial](#)
- [napari labels layer tutorial](#)
- [Multi-dimensional image visualization in Python using napari \[NEUBIAS Academy@Home\] webinar](#)

For opening an image, we still use scikit-image:

```
[ ]: from skimage.io import imread
image = imread('../data/blobs.tif')
# print out the spatial dimensions of the image
image.shape
```

Export code to Jupyter Notebooks

- Jupyter notebooks are an ideal way for communicating about code and solving issues collaboratively



```
In [1]: import numpy as np
from skimage.io import imread
from my_library import segment_image

In [2]: # Load image
image = imread('blobs.tif')

In [3]: # segment image
labels = segment_image(image)

-----
NotImplementedError: Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_4480\1391230453.py in <module>
      1 # segment image
----> 2 labels = segment_image(image)

c:\structure\code\example_image_analysis_script\my_library.py in segment_image(image)
     10
     11 # label connected components
----> 12 result = label(binary)
     13
     14 return result

~\miniconda3\envs\biol\lib\site-packages\skimage\measure\_label.py in label(input, background, ret
urn_num, connectivity)
     118         return_num=return_num, connectivity=connectivity)
     119     else:
--> 120         return clabel(input, background, return_num, connectivity)

skimage\measure\_ccomp.pyx in skimage.measure._ccomp.label_cython()

skimage\measure\_ccomp.pyx in skimage.measure._ccomp.get_shape_info()

NotImplementedError: Only for images of dimension 1-3 are supported, got a 0D one

In [ ]: # count objects
number_of_objects = labels.max()
print('Number of objects', number_of_objects)

In [ ]: #
```



FocalPlane
Where biology meets microscopy

Hosted by
Journal of
Cell Science

Home About us Topics Gallery Jobs Events Resources Network

Contact us Log in/register

Home / How to / Collaborative bio-image analysis script ...

Collaborative bio-image analysis script editing with git

Posted by Robert Haase, on 4 September 2021

TL;DR: I'm a computer scientist who often collaborates with biologists on bio-image analysis scripts. We are using more and more *git*, a version control program, for working on code collaboratively. When using *git*, we speak about *repositories*, *commits* and *pushing to the origin*. We also make *forks*, send *pull-requests* and *merge* code. This blog post explains these terms and demonstrates how a typical collaborative bio-image analysis scripting project looks like.

Write a 'How to' post
Create an account or log in to post your story on FocalPlane.
[Log in/register](#)

Sign up
Register and join our mailing list to stay up-to-date with newly



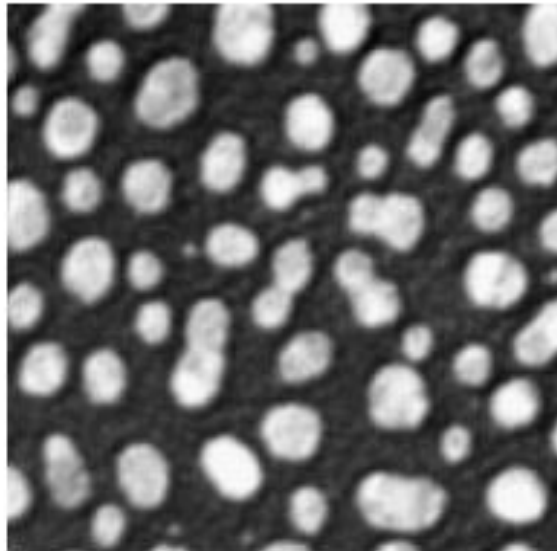
chatGPT

ChatGPT for analyzing scientific image data

- ChatGPT can perform [yet simple] analysis tasks,
- and tell you what it did.

ChatGPT 4

You



Segment the bright blobs in this image

Segment the bright blobs in this image

```
python
import cv2 as cv
import numpy as np
from matplotlib import pyplot as plt

# Load the image
image_path = '/mnt/data/blobs.jpg'
image = cv.imread(image_path, cv.IMREAD_GRAYSCALE)

# Check if image is loaded properly
if image is None:
    print('Could not open or find the image:', image_path)
else:
    # Threshold the image to get the blobs in binary
    _, binary = cv.threshold(image, 128, 255, cv.THRESH_BINARY)

    # Perform connected component analysis to segment the
    num_labels, labels_im = cv.connectedComponents(binary)

    # Map component labels to hue val, 0-179 is the hue range
    label_hue = np.uint8(179 * labels_im / np.max(labels_im))
    blank_ch = 255 * np.ones_like(label_hue)
    labeled_img = cv.merge([label_hue, blank_ch, blank_ch])

    # Converting cvt to BGR
    labeled_img = cv.cvtColor(labeled_img, cv.COLOR_HSV2BGR)

    # Set background label to black
    labeled_img[label_hue == 0] = 0

    # Display the image with segments
    plt.imshow(labeled_img)
```

Short detour: Generative Artificial Intelligence

Definition: "Generative artificial intelligence [...] is a type of artificial intelligence (AI) system capable of generating text, images, or other media in response to prompts."¹

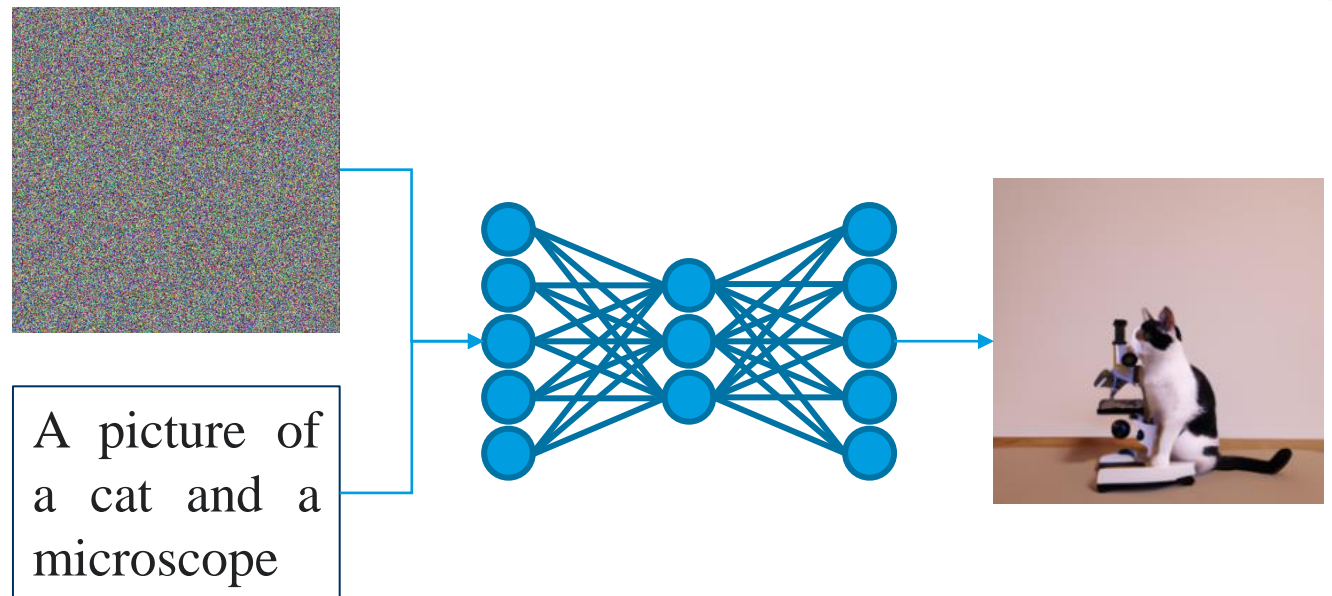
Commonly based on Neural Networks

Bridges fields:

- Natural Language Processing (NLP)
- Computer Vision (CV)

Use-cases

- Translating text
- Writing emails, text, grant proposals
- Summarizing articles
- Writing code
- General question answering
- Image generation



How does it work?

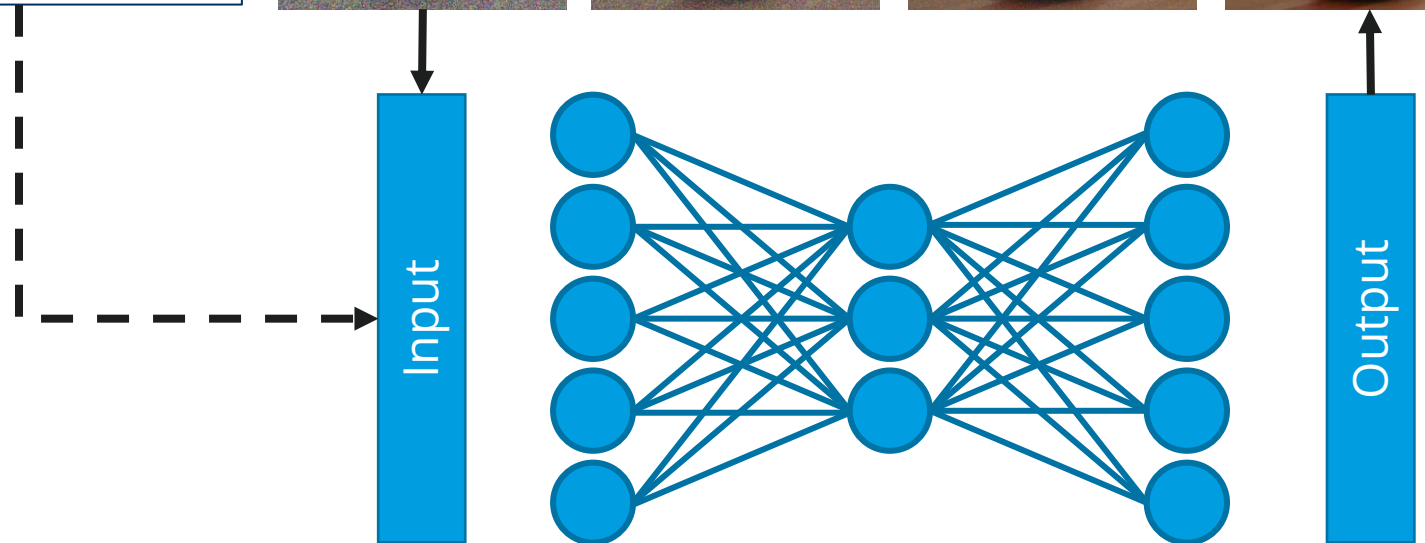
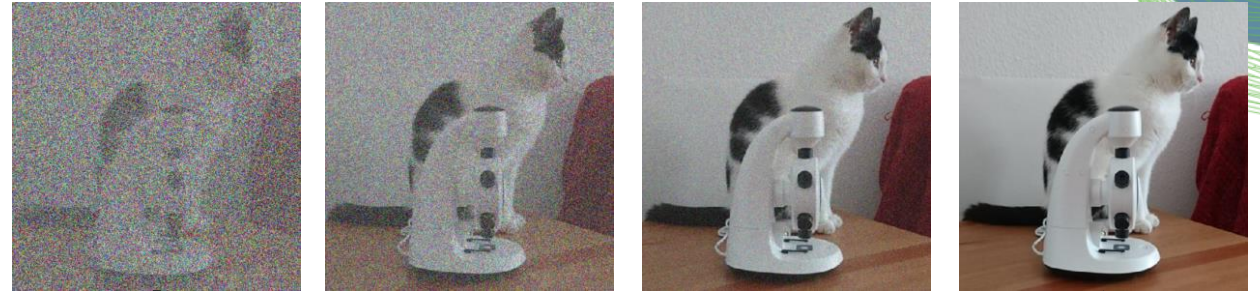
Combination of neural networks + other elements + various [multi-modal] data sources

Examples: GPT / DALL-E, Stable Diffusion

Use-case:

- Generate image from noise + text

The cat's fur is black and white.



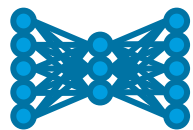
How does it work?

Combination of neural networks + other elements + various data sources

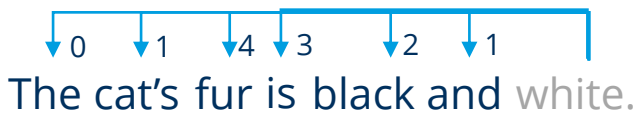
Examples: GPT / DALL-E, Stable Diffusion

Use-case:

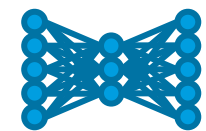
- Generate image from noise + text



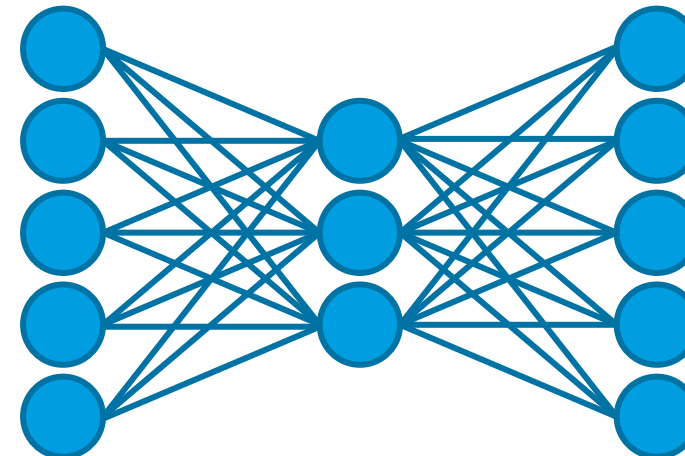
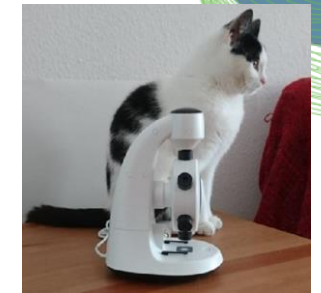
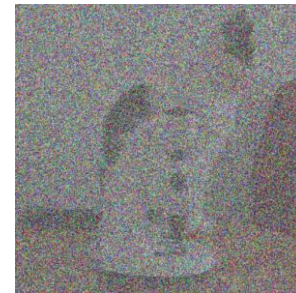
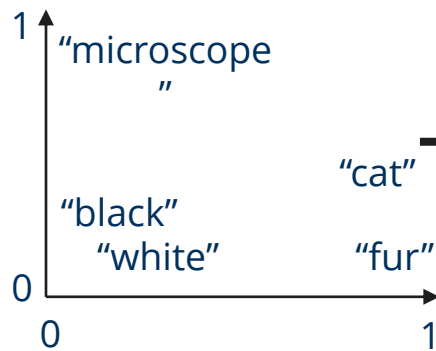
Self-attention



The cat's fur is black and white.



Word Embedding



What are large language models good in?

Translation tasks

```
[4]: %%prompt
      Translate the following joke to German:

      What is every parent's favorite Christmas song? Silent Night!

      Was ist das Lieblingsweihnachtslied aller Eltern? Stille Nacht!
```

Restructuring data

```
[6]: %%prompt
      Given a story, please restructure the contained data.

      Story:
      The compute center of the university recently conducted a study to figure
      out needs of students for computational resources. After interviewing 100
      students they concluded that most of them were happy with 250 GB of cloud
      storage.

      Restructure the story above to extract these numbers:
      * Number of students asked:
      * Cloud storage size:
```

Number of students asked: 100 Cloud storage size: 250 GB

What are large language models **not so good** in?

Querying knowledge
(in particular recent information)

[15]: %%prompt
What's Robert Haase's (Uni Leipzig) research field ?

I couldn't find information specific to Robert Haase from the University of Leipzig. It's possible that he may not be a prominent or well-known researcher in a specific field or his information may not be easily accessible online.

[18]: %%prompt
What's Robert Haase's (TU Dresden) research field ?

Robert Haase from TU Dresden is a researcher in the field of computer science and medical image analysis.

Math
(in particular counting)

[25]: %%prompt
How many o are in Woolloomooloo ?

There are 6 "o" letters in Woolloomooloo.



Woolloomooloo
New South Wales 2011
Australia

Partly cloudy · 23°C
9:24 PM

Prompt Engineering

- Set context / assign role
- Define output style
- Give hints / conditions
- List tasks
- Provide data

```
%%prompt
```

```
You are a professional academic consultant who gives advice to students for writing their thesis.
```

```
Give these hints:
```

- * Submit the thesis registration form only **if** you're sure you can finish writing in 5 months.
- * Start writing your thesis early, e.g. **the** fundamentals chapter can be written **from** day one.
- * Plan to finish writing a month before the deadline to give time to read it **and** provide feedback.

Here are some important hints to consider while writing your thesis:

1. Evaluate your readiness: Before submitting the thesis registration form, carefully assess your commitment and availability. Ensure that you are confident in completing your thesis within the given timeframe of five months. This will prevent unnecessary stress and help you maintain a focused mindset throughout the writing process.
2. Begin early: Starting your thesis early is crucial. Begin by working on the fundamentals chapter, as it lays the foundation for your entire thesis. By initiating this chapter from day one, you will establish a good writing routine and make steady progress as you move forward.
3. Plan ahead: Create a realistic schedule to guide your writing process. Break down the work into smaller tasks, setting deadlines for each chapter or section. This will help you maintain a structured approach and make the workload more manageable. Be sure to allocate time for research, data analysis, and revisions.
4. Aim to finish ahead of time: Plan to complete your thesis writing at least one month before the official deadline. This buffer time will allow you to thoroughly review and revise your work before final submission. It will also provide an opportunity to seek feedback from professors or peers, improving the overall quality of your thesis.

Prompt Engineering

- Set context / assign role
- Define output style
- Give hints / conditions
- List tasks
- Provide data

```
%%prompt
```

```
You are a professional academic consultant who gives advice to students for writing their thesis.  
Keep it short.
```

```
Give these hints:
```

- * Submit the thesis registration form only **if** you're sure you can finish writing in 5 months.
- * Start writing your thesis early, e.g. the fundamentals chapter can be written **from** day one.
- * Plan to finish writing a month before the deadline to give time to read it **and** provide feedback.

As a professional academic consultant, here are some helpful hints for writing your thesis:

1. Only submit the thesis registration form if you are confident about completing it within the given timeframe of 5 months.
2. Begin writing your thesis early to ensure a smooth process. For instance, you can start with the fundamentals chapter right from day one.
3. Plan your schedule effectively to finish writing your thesis at least a month before the submission deadline. This will allow ample time for reading, revising, and receiving feedback.

By following these recommendations, you'll have a better chance of successfully completing your thesis on time and with improved quality. Good luck!

Prompt Engineering

- Set context / assign role
- Define output style
- Give hints / conditions
- List tasks
- Provide data

```
%%prompt
```

```
You are a professional academic consultant who gives advice to students for writing their thesis.  
Keep it short.
```

```
Write like Yoda would.
```

```
Give these hints:
```

- * Submit the thesis registration form only **if you're sure you can finish writing in 5 months.**
- * Start writing your thesis early, e.g. **the** fundamentals chapter can be written **from** day one.
- * Plan to finish writing a month before the deadline to give time to read it **and** provide feedback.

Hints for thesis writing, I have. Follow them, you must.

- Only submit the thesis registration form, if in 5 months, complete writing, you are sure.
- From day one, start writing your thesis, you must. Begin with the fundamentals chapter.
- A month before the deadline, finish writing, you should. Time to read and provide feedback, it will give you.

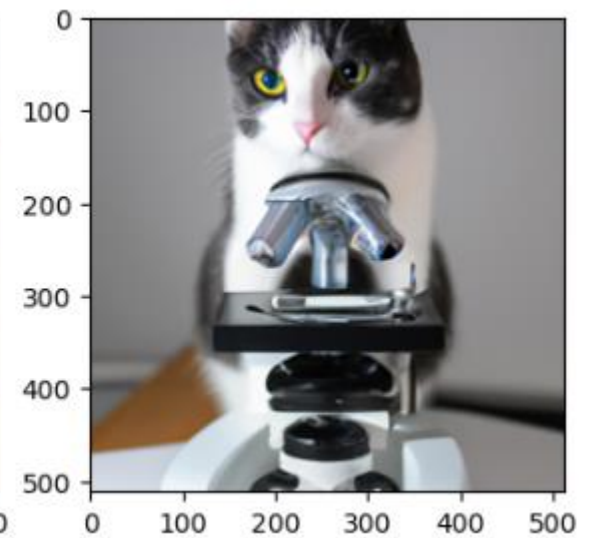
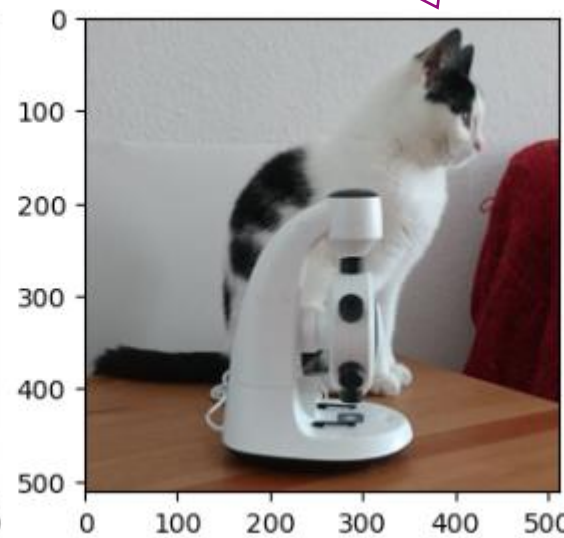
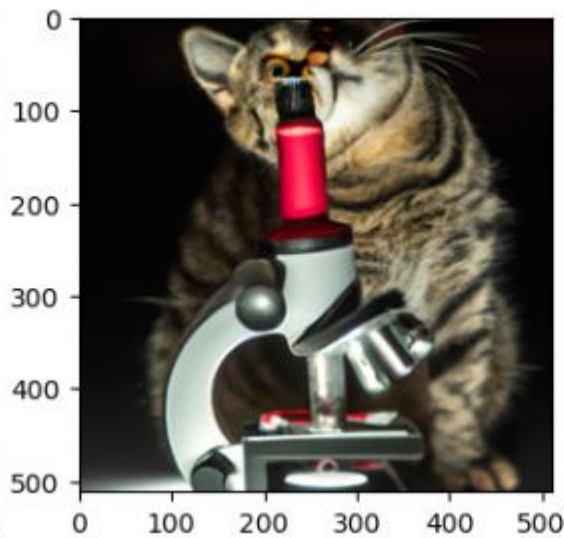
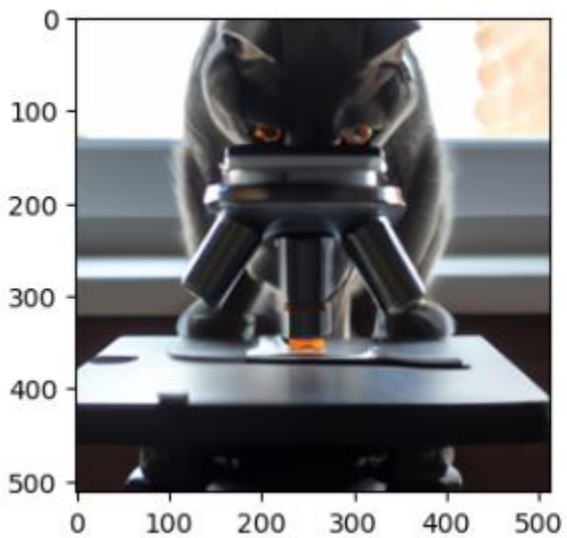
Follow these hints, you must. Successful thesis writing, you will achieve. Hmmm.

Prompt engineering

Prompts can be used to write code, but also to generate images, e.g. with DALL-E. One can generate quite realistic images given a detailed prompt.

```
cat_microscope_prompt = """"  
Image of a cat sitting behind a microscope.  
""""
```

One cat
is real.

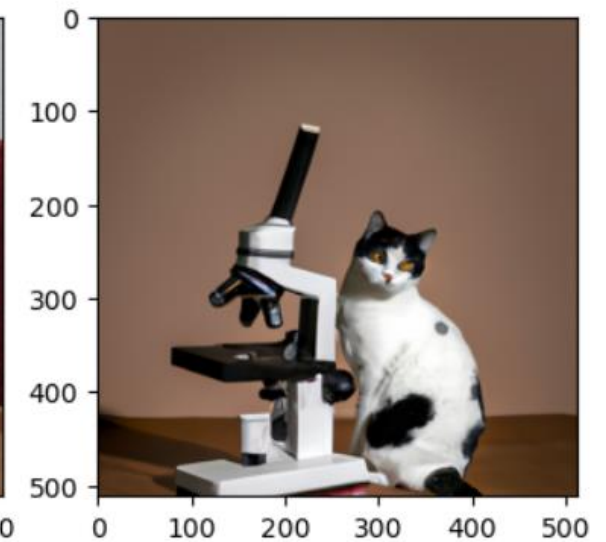
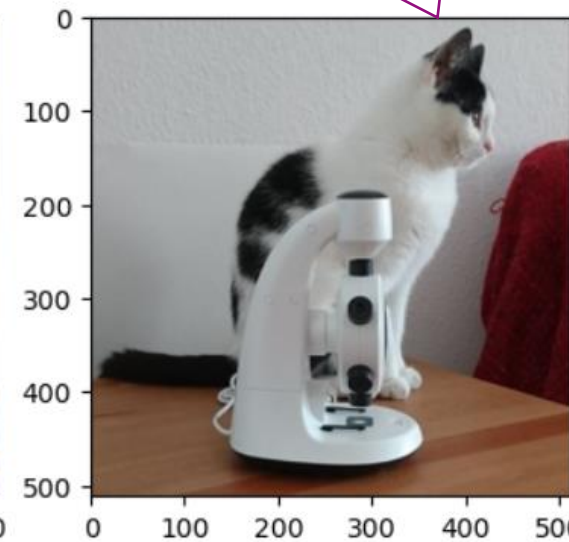
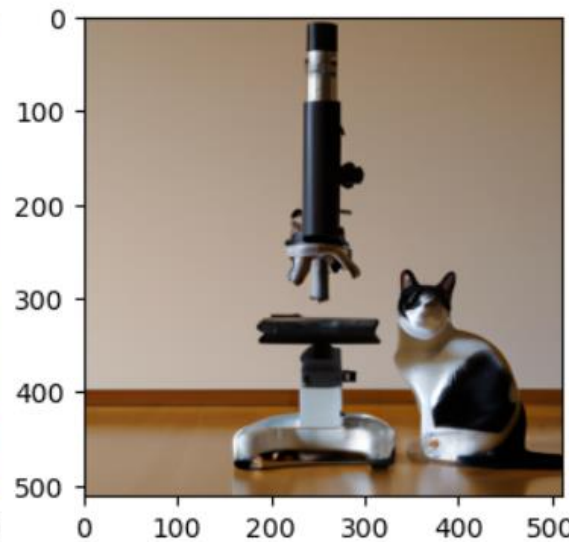
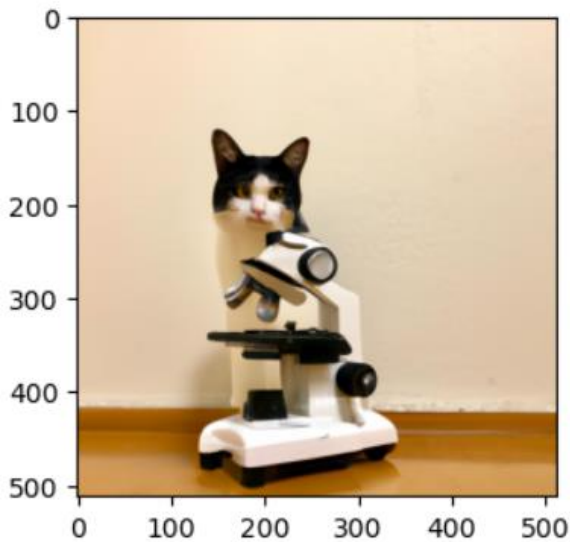


Prompt engineering

Prompts can be used to write code, but also to generate images, e.g. with DALL-E.

```
[5]: cat_microscope_prompt = """  
Image of a cat sitting behind a microscope.  
Both are on a brown floor in front of a white wall.  
The cat is mostly white and has some black dots.  
The cat sits straight.  
The cat is a bit larger than the microscope.  
"""
```

One cat
is real.



Prompt engineering

chatGPT can solve simple image analysis tasks

Prompt

```
simple_question = """
Write Python code only and no additional explanatory text.

Write a python program, that
* loads the file `../../data/blobs.tif`,
* labels objects in this image, and
* visualize results.

Assume this program would be executed in a Jupyter notebook.
It is not necessary to save the results. Show the results in Jupyter.
"""
```

Response

```
code = prompt(simple_question)
print(code)

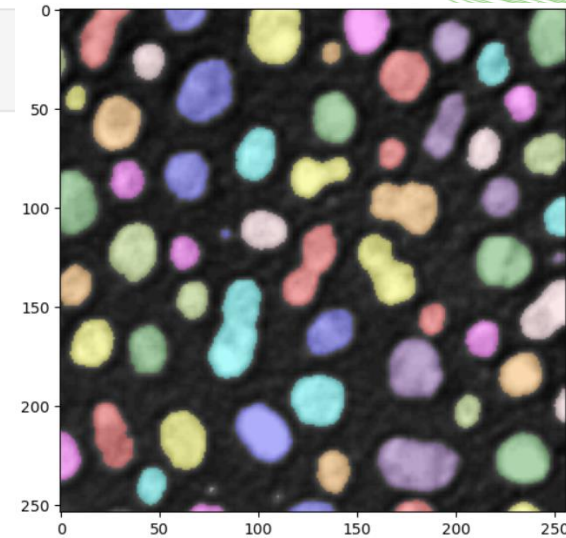
```python
from skimage.io import imread
from skimage.filters import threshold_otsu
from skimage.measure import label
from skimage.color import label2rgb
import matplotlib.pyplot as plt

image = imread('../../data/blobs.tif')
thresh = threshold_otsu(image)
binary = image > thresh
labels = label(binary)
image_label_overlay = label2rgb(labels, image=image)

fig, ax = plt.subplots(figsize=(10, 6))
ax.imshow(image_label_overlay)

plt.show()
```
```

Result



ChatGPT for analyzing scientific image data

ChatGPT in comparison with the **Bio-image Analysis GPT**

ChatGPT 4 ▾



You

How can I measure the number of neighbors in a label image?



ChatGPT

Measuring the number of neighbors in a labeled image, where each unique label represents a different object or region, often involves analyzing the adjacency or connectivity between the labeled regions. Here's a general approach you might take using image processing techniques:

1. **Preprocessing:** Ensure that your image is segmented and labeled correctly. Each distinct object or region should have a unique label (e.g., 1, 2, 3, ...), and the background can be labeled as 0 or another distinct value.
2. **Define Connectivity:** Determine the type of connectivity you are interested in. In a 2D image, typically 4-connectivity (up, down, left, right) or 8-connectivity (includes diagonals) is used.
3. **Create Adjacency Matrix:** An adjacency matrix is a square matrix used to represent a finite graph. The elements of the matrix indicate whether pairs of vertices are adjacent or

ChatGPT for analyzing scientific image data

ChatGPT in comparison with the Bio-image Analysis GPT

```
python
from skimage.measure import label, regionprops
import numpy as np

# Let's assume 'image' is your labeled image matrix
labeled_image = label(image)

# Define an empty list to hold counts of neighbors for each label
neighbors_count = []

for region in regionprops(labeled_image):
    # Here, you would implement the logic to check neighbors for each
    # and count the unique labels encountered.
    # Append the count of neighbors for this region to the neighbors_count
    pass

# neighbors_count now holds the number of neighbors for each label
```

More concise responses with some domain knowledge

Limited BIA domain-specific knowledge

Bio-image Analysis GPT

You: How can I measure the number of neighbors in a label image?

Bio-image Analysis GPT: To measure the number of neighbors in a label image, you can use the 'cyclesperanto' (abbreviated as 'cle') library. Here's a concise guide on how to achieve this:

1. Create a Neighbor Count Map: Use the 'cle.touching_neighbor_count_map' function to count the touching neighbors for each labeled object (cell). This function creates a new image where each pixel's intensity represents the number of neighbors for the labeled object it belongs to.

```
python
neighbor_count_image = cle.touching_neighbor_count_map(cells)
```

2. Visualize the Neighbor Count: You can visualize this neighbor count map using 'cle.imshow', applying a color map for better visibility.

```
python
cle.imshow(neighbor_count_image, color_map='jet', colorbar=True)
```

3. Exclude Edge Labels: It's important to note that the counts along the image border may not be accurate. To address this, use 'cle.exclude_labels_on_edges' to exclude cells touching the image borders.

```
python
cells_ex_border = cle.exclude_labels_on_edges(cells)
```

4. Correct the Neighbor Count Map: Apply the exclusion to the neighbor count image.

```
python
neighbor_count_image_ex_border = neighbor_count_image * (cells_ex_border != 0)
```

5. Read Neighbor Counts: Finally, extract the neighbor counts into a list or array using 'cle.read_intensities_from_map'.

```
python
neighbor_counts = cle.read_intensities_from_map(cells_ex_border, neighbor_count_image_ex_border)
```

BiA-Bob

- Open-source alternative to OpenAI's Data Analyst
- No need to upload your [image] data
- Extensible through plugins

```
[1]: from bia_bob import bob  
     bob.__version__
```

```
[1]: '0.3.0'
```

```
[ ]:
```



With Christian
Tischer (EMBL
Heidelberg)
@tischitischer

BiA-Bob

- Open-source alternative to OpenAI's Data Analyst
- No need to upload your [image] data
- Extensible through plugins

The screenshot shows a JupyterLab environment with a code cell containing the following instruction:

```
[1]: load demo/blobs.tif,  
segment it and label individual objects,  
measure area and perimeter,  
show the original image, the label image and an area versus perimeter plot in one figure
```

The output of the code cell is a natural language response: "I will load an image named `demo/blobs.tif`, segment it to identify individual objects, label these objects, and then measure their area and perimeter. Finally, I will display the original image, the label image, and a plot of area versus perimeter for each object in a single figure."

Below the text, three visual outputs are displayed:

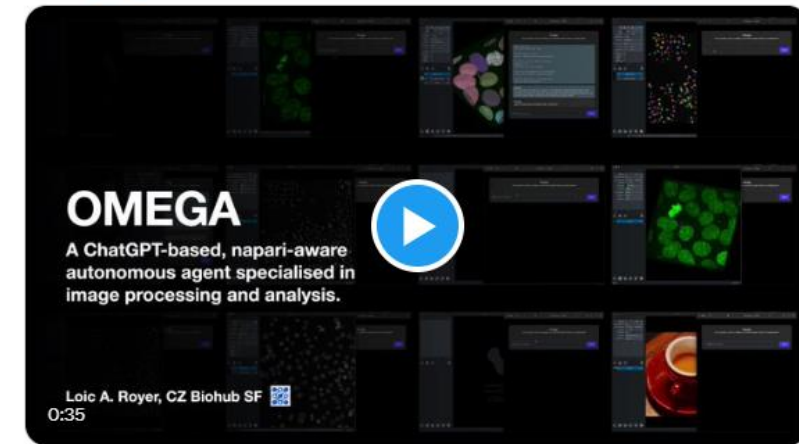
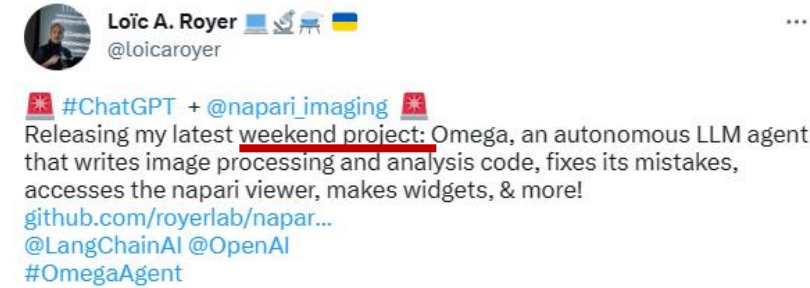
- Original Image:** A grayscale image showing a dense field of small, irregularly shaped objects.
- Labeled Image:** The same image with each object segmented and labeled with a different color.
- Area vs. Perimeter:** A scatter plot showing the relationship between the area and perimeter of the objects. The x-axis is labeled "Area" (ranging from 0 to 800) and the y-axis is labeled "Perimeter" (ranging from 0 to 120). The data points show a strong positive correlation.



With Christian Tischer (EMBL Heidelberg) @tischitischer

napari-chatGPT

Napari-chatGPT can automate programming plugins / “widgets”.



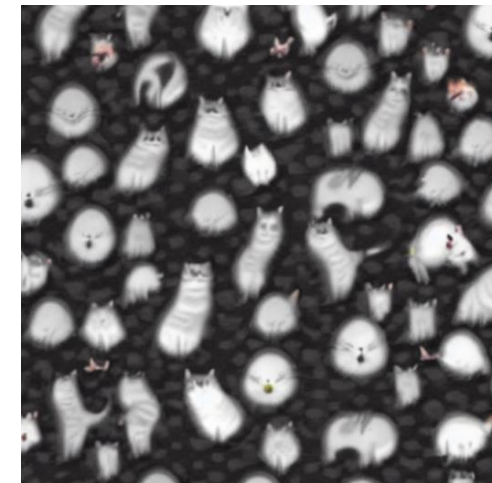
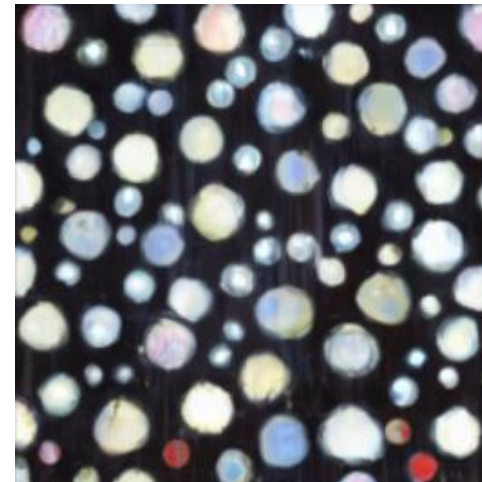
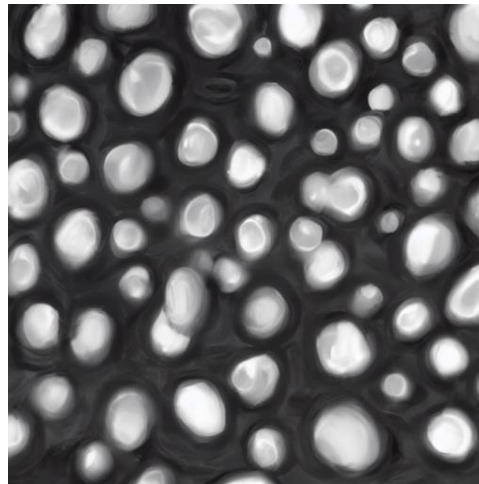
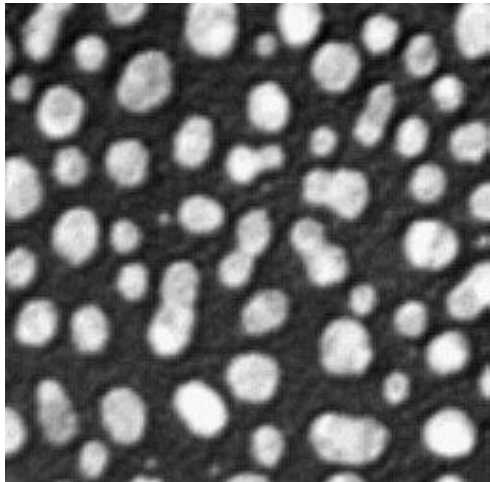
5:19 AM · May 3, 2023 · 183.6K Views

208 Retweets 44 Quotes 770 Likes 424 Bookmarks

What else can [multi-modal] LLMs do?

Generate images, e.g. for augmenting data

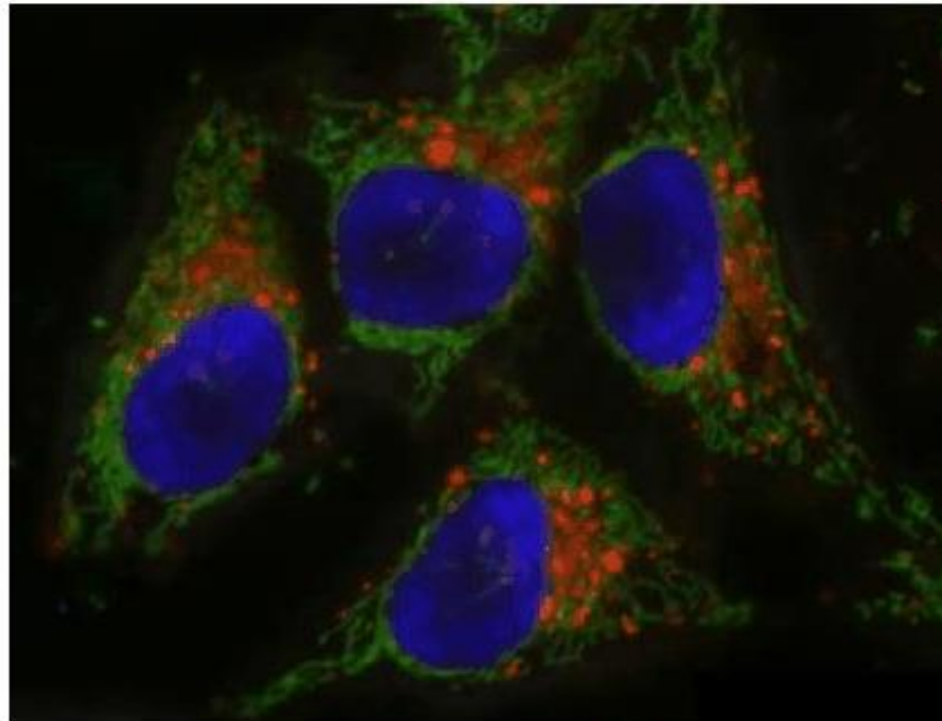
Potentially useful to
make algorithms
more robust



New technologies bring new risks...

If you can generate images,
you can also generate parts of images....

[6]:



Interesting
challenges for our
community ahead

Generative Artificial Intelligence

Challenges

- Data safety / security / integrity
- Computational cost of training of neural networks
 - CO2-footprint/climate change
 - Accessibility
- Bias: “a nice photo of a human”
- Hallucinations
- False information / fake news
- Glitch tokens
- Debugging is hard



Model: Default (GPT-3.5)

R Who developed Gradoop? ChatGPT 4

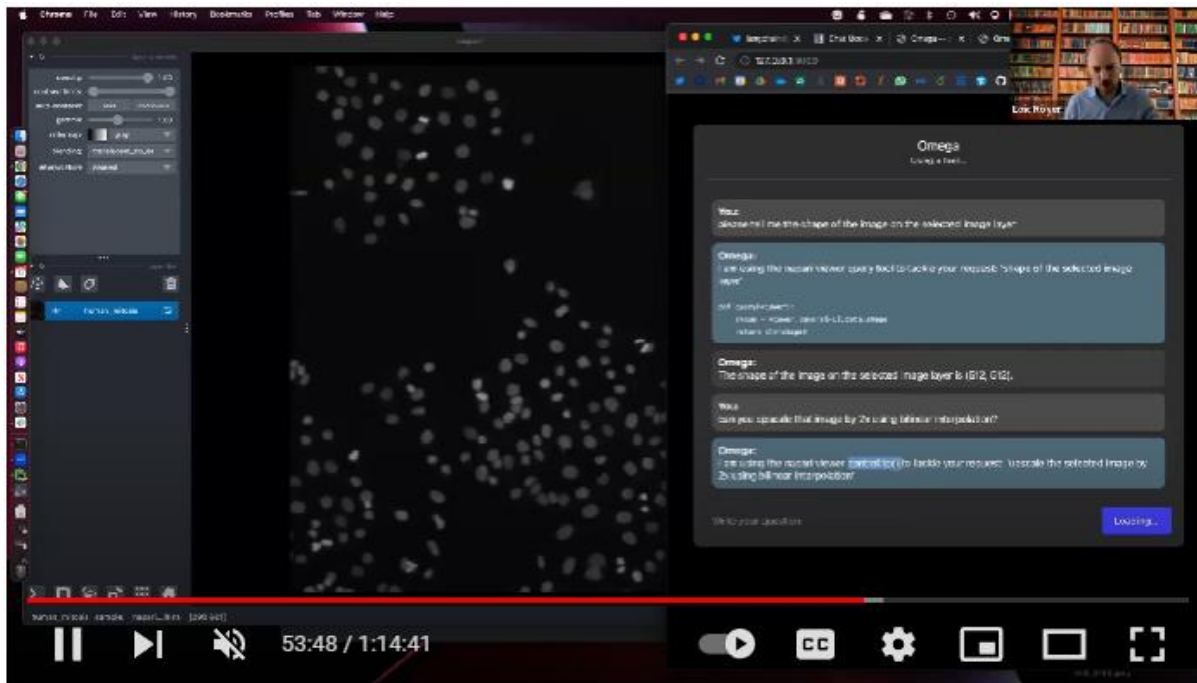
You Who developed Gradoop?

ChatGPT Gradoop is a scalable graph data management and analytics framework developed by the Database Systems Group at Leipzig University, Germany. It was specifically designed to support the efficient analysis of large-scale graph data, providing capabilities for graph storage, graph transformations, pattern matching, and analytical data processing. The framework is built on top of Apache Flink, which is a stream-processing and batch-processing framework, allowing Gradoop to leverage Flink's scalable data processing capabilities.

May 2023

March 2024

Further reading / watching



Code review of Napari-ChatGPT by Loic Royer (CZI Biohub)



Scientific Computing ...
122 subscribers



https://www.youtube.com/watch?v=JMo6Sn-L_j4



DigitalSreeni

@DigitalSreeni 71.3K subscribers 380 videos

This channel walks you through the entire process of learning to code in P... >



308 - An introduction to language models with focu...



311 - Fine tuning GPT2 using custom documents



309 - Training your own Chatbot using GPT

<https://www.youtube.com/c/digitalsreeni>

Acknowledgements

Communities & platforms



NFDI4
BIOIMAGE



GloBIAS
global bioimage analysts' society



neubias
network of european
bioimage analysts



FocalPlane
Where biology meets microscopy



BiAPoL team

- Marcelo Zoccoler
 - Johannes Soltwedel
 - Maleeha Hassan
 - Stefan Hahmann
 - Somashekhar Kulkarni
- Former lab members:
- Ryan George Savill
 - Laura Zigutyte
 - Mara Lampert
 - Allyson Ryan
 - Conni Wetzker
 - Till Korten



Funding



Bundesministerium
für Bildung
und Forschung

SACHSEN



Diese Maßnahme wird gefördert durch die Bundesregierung aufgrund eines Beschlusses des Deutschen Bundestages. Diese Maßnahme wird mitfinanziert durch Steuermittel auf der Grundlage des von den Abgeordneten des Sächsischen Landtags beschlossenen Haushaltes.

Chan
Zuckerberg
Initiative



Deutsche
Forschungsgemeinschaft

Some Figures were generated using



DALL·E