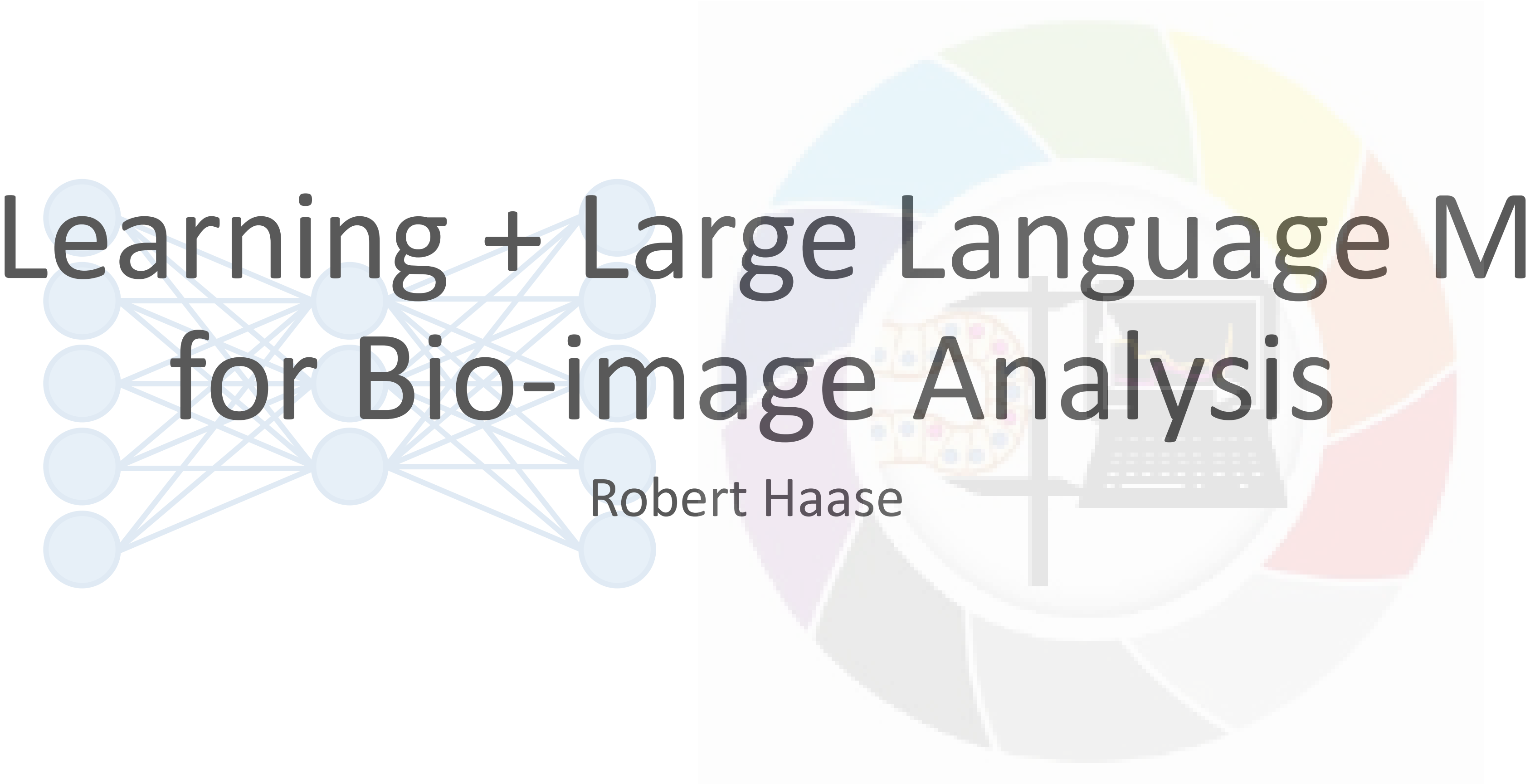


Deep Learning + Large Language Models for Bio-image Analysis

Robert Haase

Reusing Material from

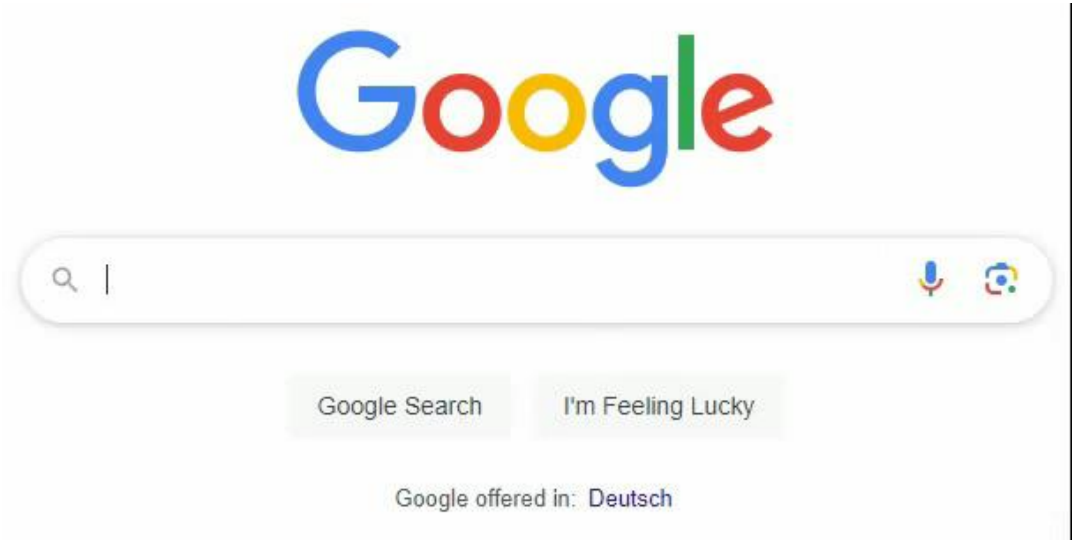
Alex Krull (MPI CBG), Martin Weigert (EPFL Lausanne), Uwe Schmidt (MPI CBG), Johannes Soltwedel, Till Korten (PoL TU Dresden), Loic A. Royer (CZ Biohub), Alexandr Dibrov (CSBD/MPI-CBG Dresden), Aditya Ramesh et al (OpenAI) and Alexandr Khrapichev (University of Oxford), Carsen Stringer et al (HHMI Janelia)

@haesleinhuepf

June 2023

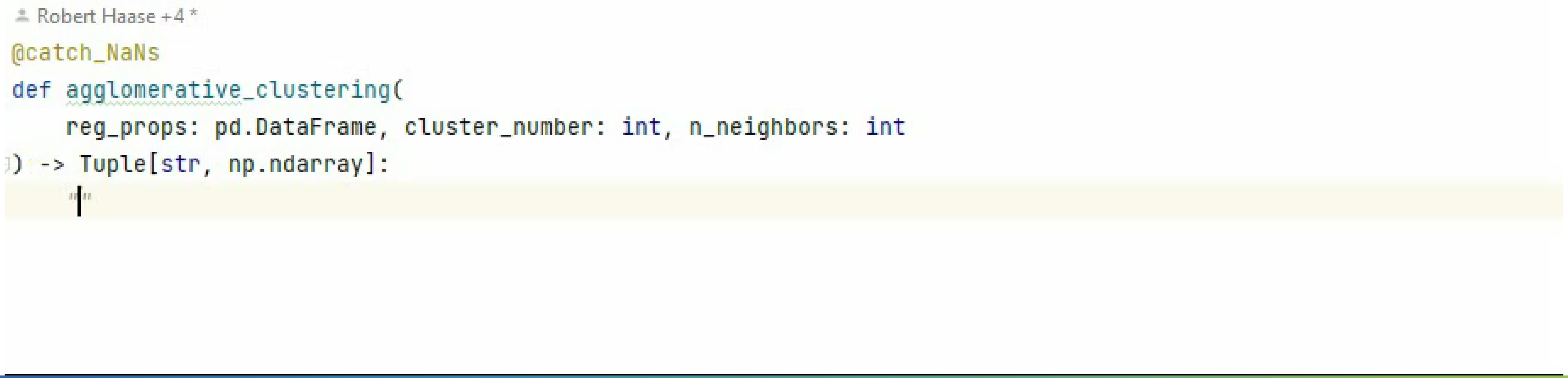
Deep Learning and Large language models

- How it started



- How it's going

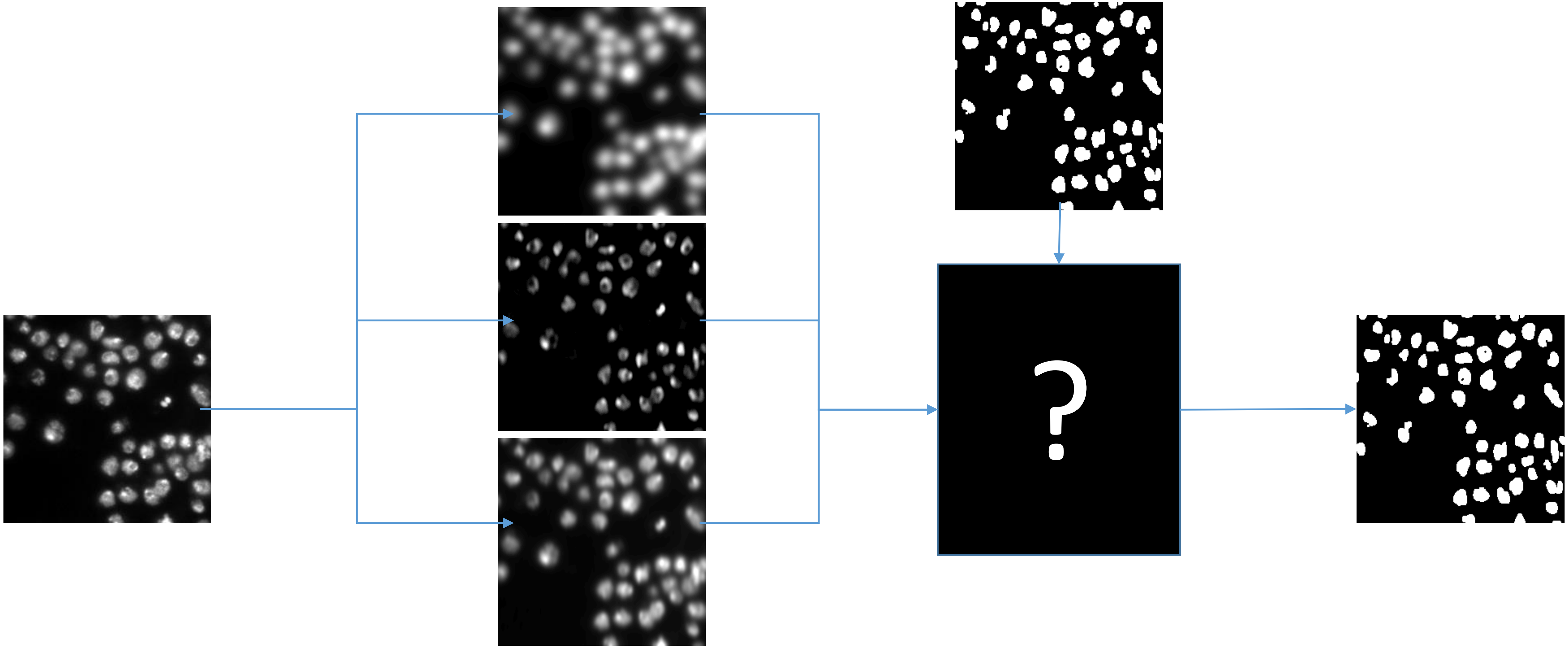
  (GitHub copilot)



```
@catch_NaNs
def agglomerative_clustering(
    reg_props: pd.DataFrame, cluster_number: int, n_neighbors: int
) -> Tuple[str, np.ndarray]:
    """
```

Machine learning for image analysis

- Recap: In classical supervised machine learning, we typically select features for training our classifier

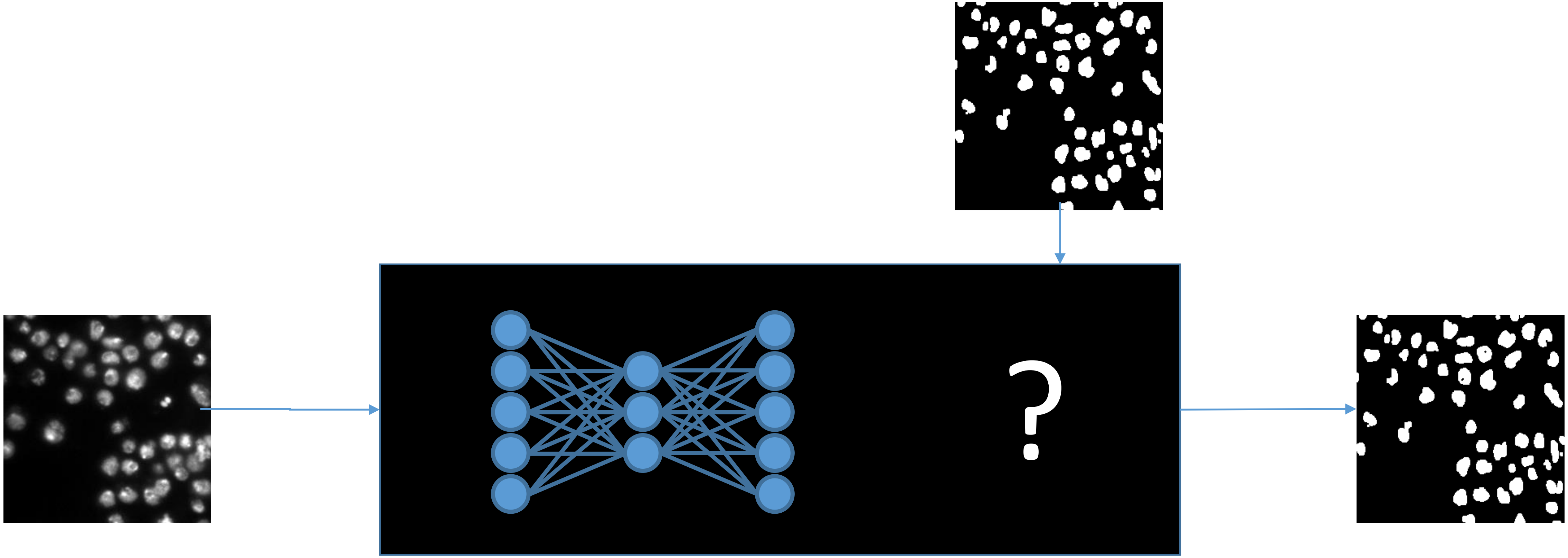


Convolutions

Image data source: BBBC038v1, available from the Broad Bioimage Benchmark Collection (Caicedo et al., Nature Methods, 2019).

Deep learning for image analysis

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

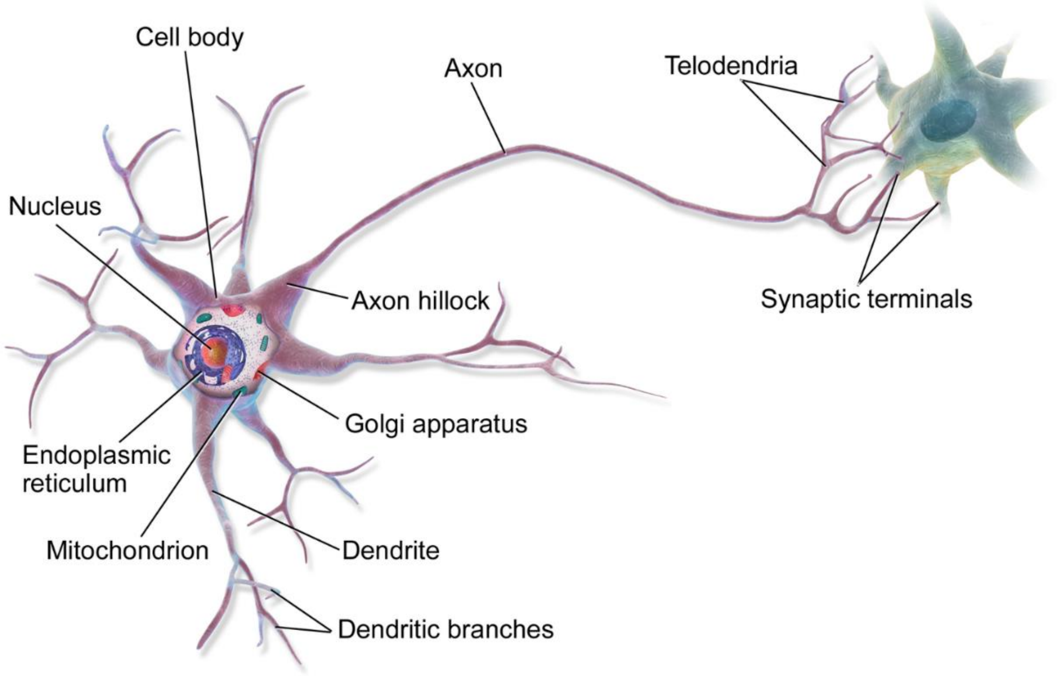


Convolutional neural network

Image data source: BBBC038v1, available from the Broad Bioimage Benchmark Collection (Caicedo et al., Nature Methods, 2019).

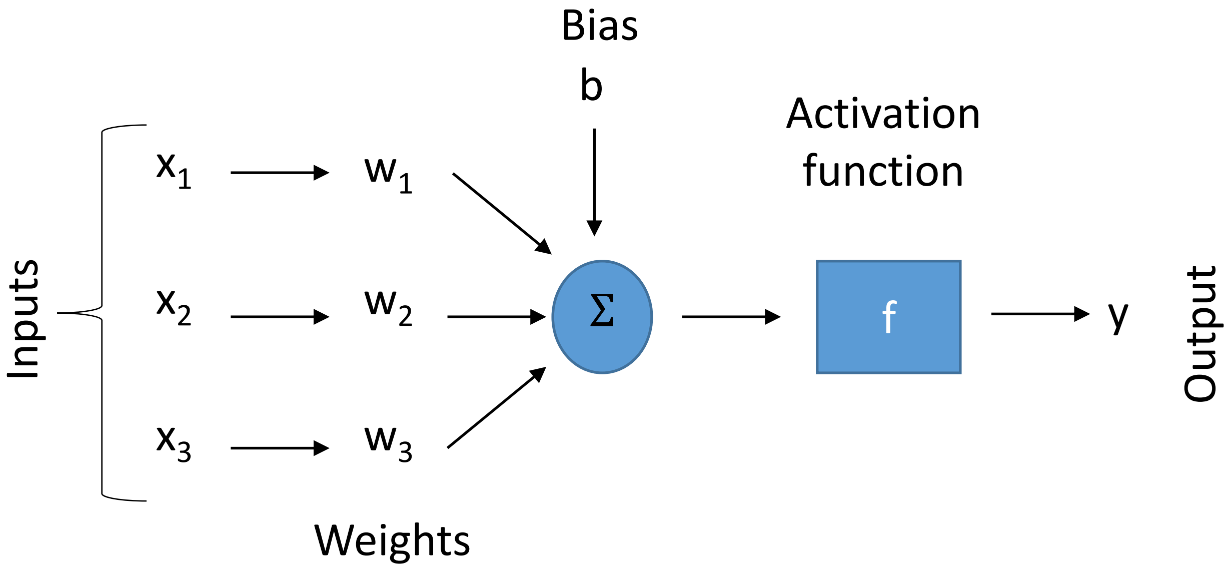
Neural networks

- How biologists see neurons



- How computer scientists see neurons

"perceptron"



$$y = f(w_1 x_1 + w_2 x_2 + w_3 x_3 + b)$$

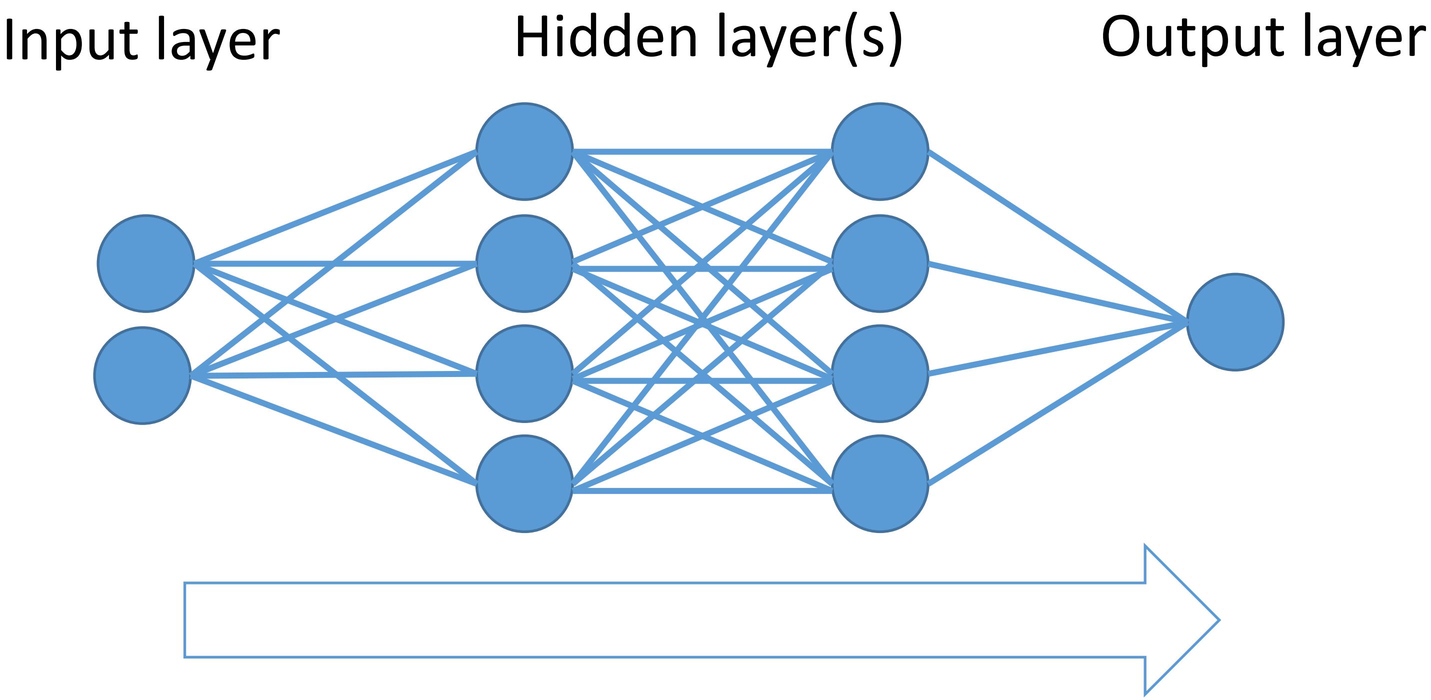
@haesleinhuepf

Neuron image source:
https://commons.wikimedia.org/wiki/File:Blausen_0657_MultipolarNeuron.png
Licensed CC-BY 3.0 by BruceBlaus

Neural Networks

- Early form: “Multilayer Perceptron”
- fully connected class of feedforward artificial neural network

If there are *many* hidden layers, we speak of a *deep* neural network

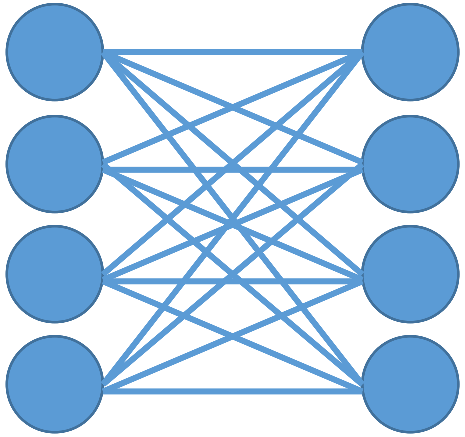


https://en.wikipedia.org/wiki/Multilayer_perceptron

Convolutional neural networks

- Layer types

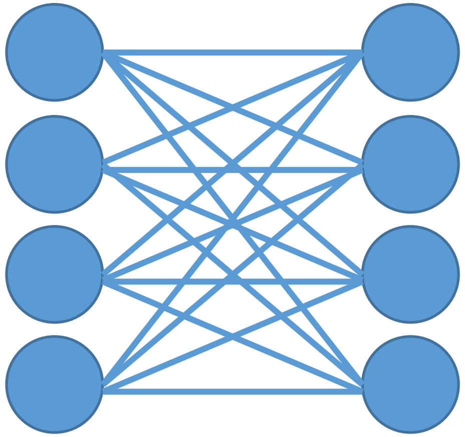
Fully connected layer



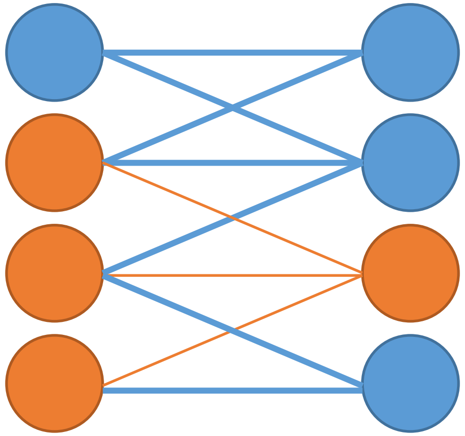
Convolutional neural networks

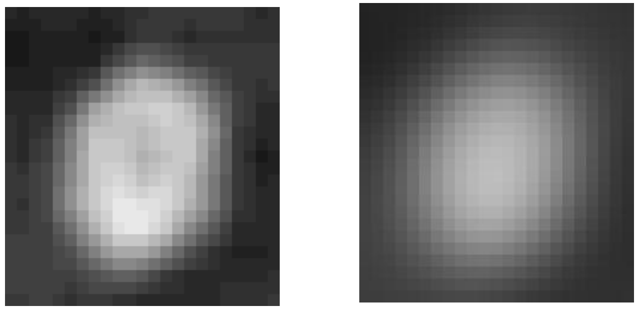
- Layer types

Fully connected layer



Convolutional layer

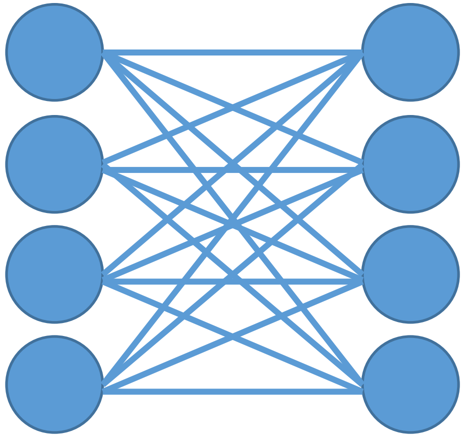




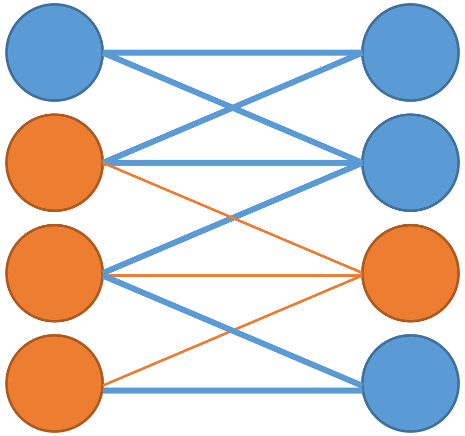
Convolutional neural networks

- Layer types

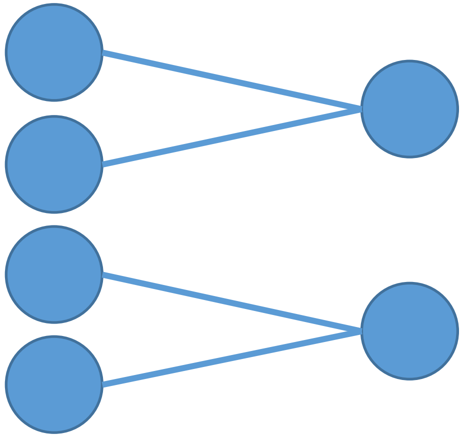
Fully connected layer

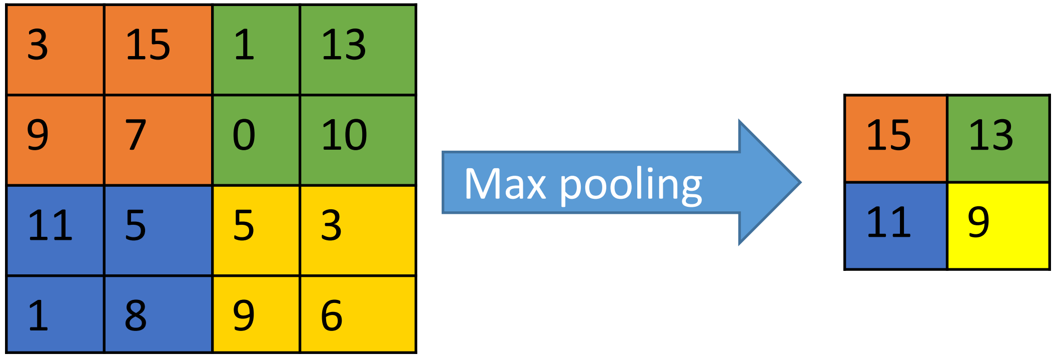


Convolutional layer



Pooling layer
("Max pool", "Average pool")





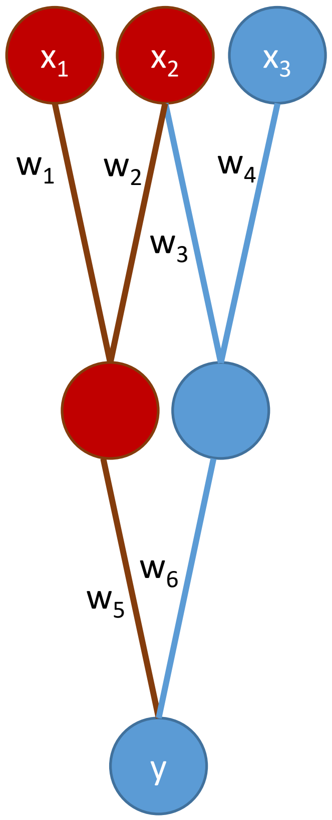
3	15	1	13
9	7	0	10
11	5	5	3
1	8	9	6

Max pooling

15	13
11	9

Convolutional neural networks

- Assuming we had no activation functions in the network layers can be reduced by eliminating brackets!



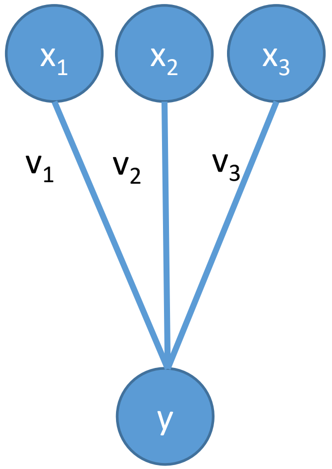
$$y = w_5(w_1x_1 + w_2x_2) + w_6(w_3x_2 + w_4x_3)$$

$$y = w_5w_1x_1 + w_5w_2x_2 + w_6w_3x_2 + w_6w_4x_3$$

$$y = w_5w_1x_1 + (w_5w_2 + w_6w_3)x_2 + w_6w_4x_3$$

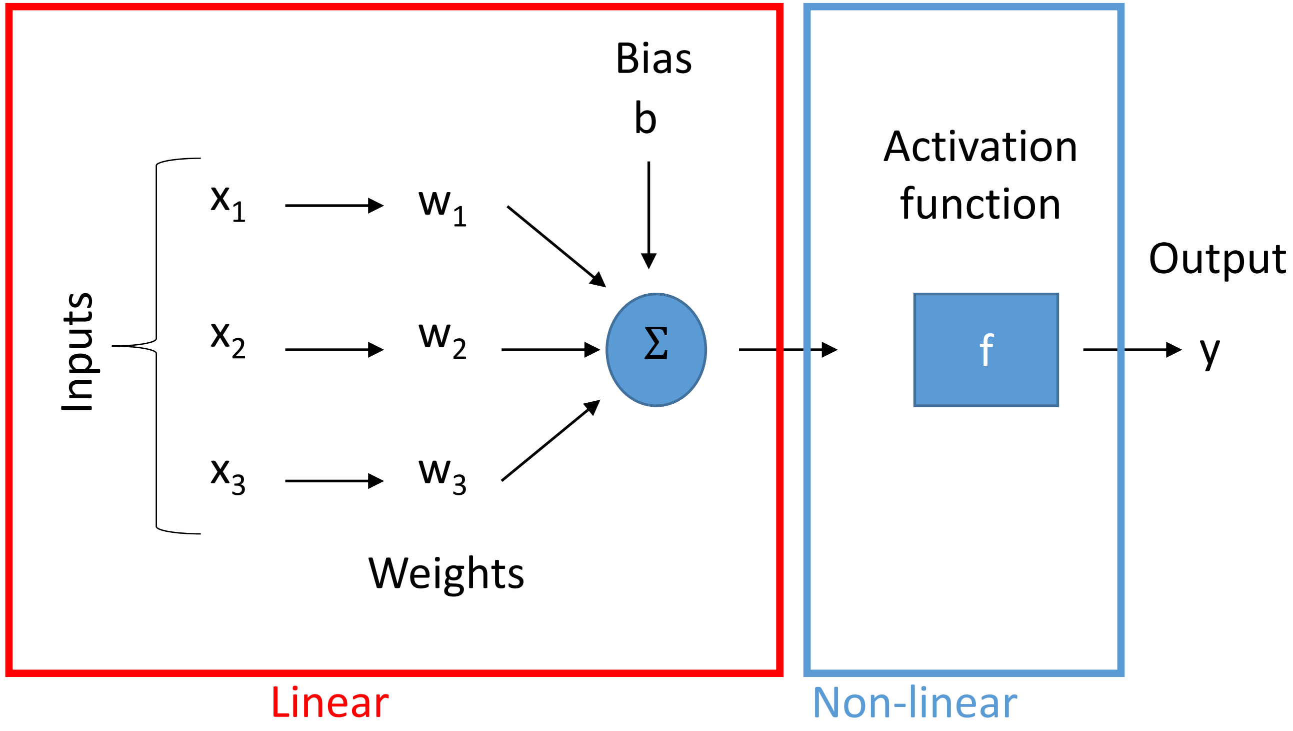
$$v_1 = w_5w_1$$
$$v_2 = w_5w_2 + w_6w_3$$
$$v_3 = w_6w_4$$

$$y = v_1x_1 + v_2x_2 + v_3x_3$$



Activation functions

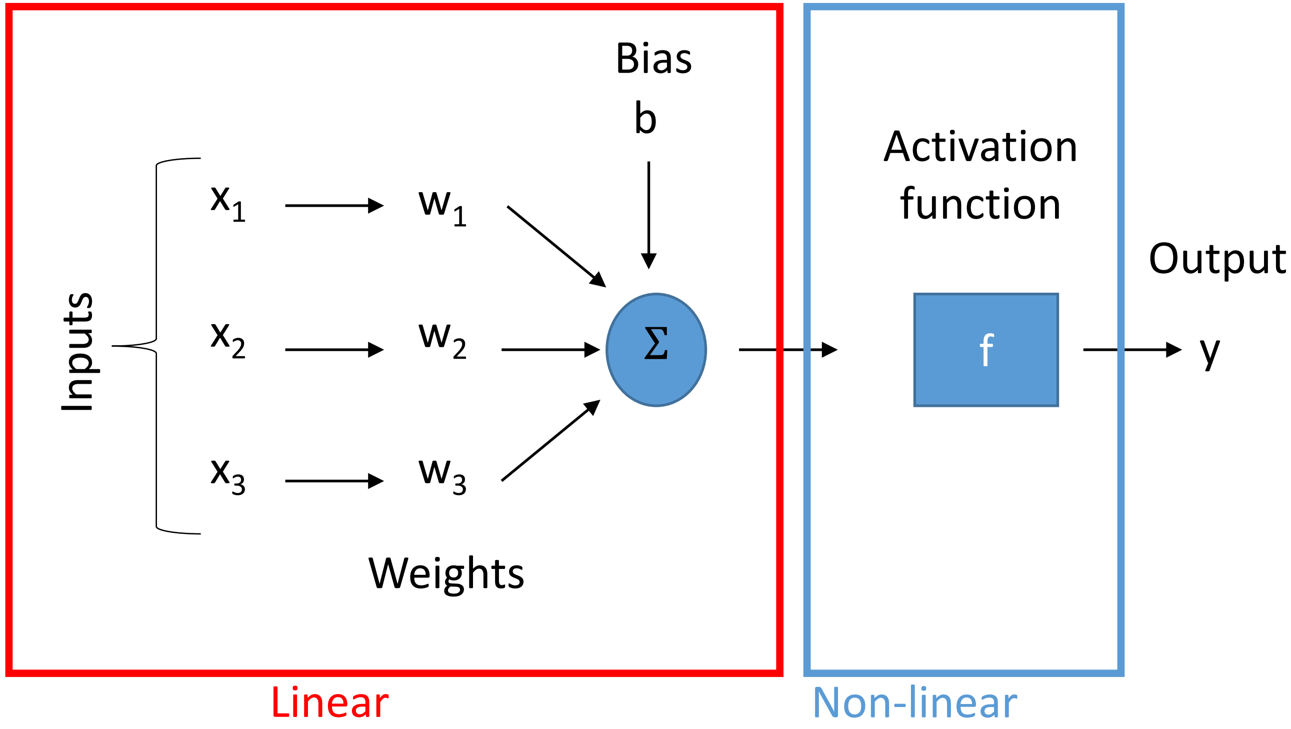
- Introduction of *non-linearity* and *activation functions* enabled what we call *deep-learning* today.

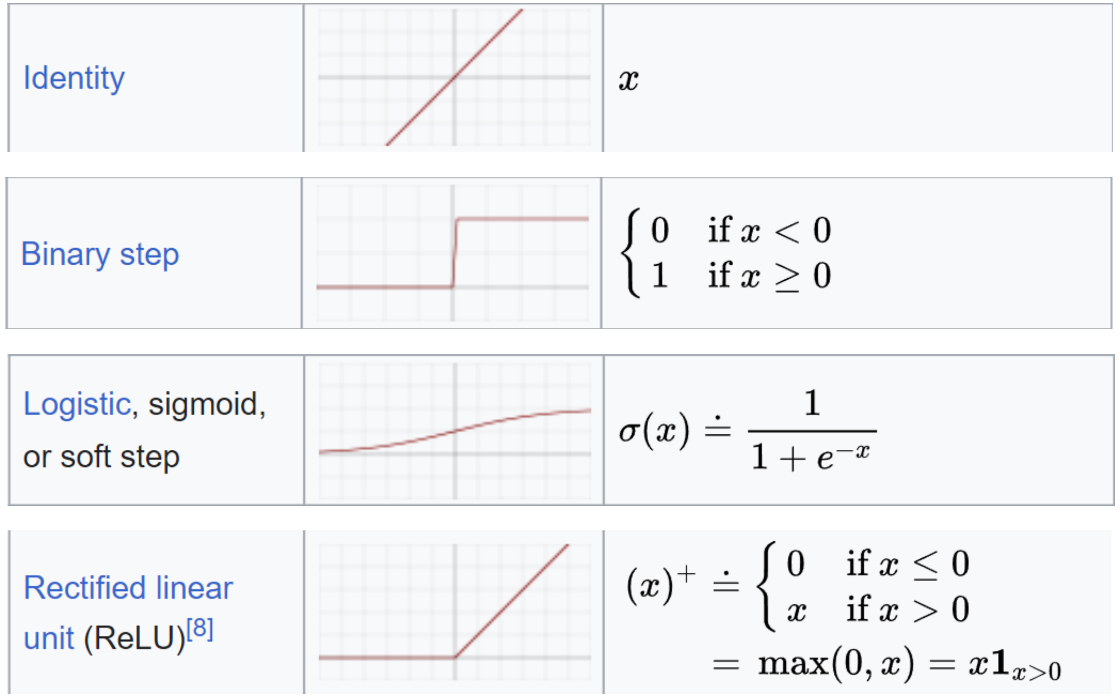


$$y = f(w_1x_1 + w_2x_2 + w_3x_3 + b)$$

Activation functions

- Introduction of *non-linearity* and *activation functions* enabled what we call *deep-learning* today.





Identity		x
Binary step		$\begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}$
Logistic, sigmoid, or soft step		$\sigma(x) \doteq \frac{1}{1+e^{-x}}$
Rectified linear unit (ReLU)[8]		$(x)^{+} \doteq \begin{cases} 0 & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases} = \max(0,x) = x\mathbf{1}_{x>0}$

Source: https://en.wikipedia.org/wiki/Activation_function
Licensed CC-BY-SA 4.0 by Laughsinthestocks

Learning: Back propagation

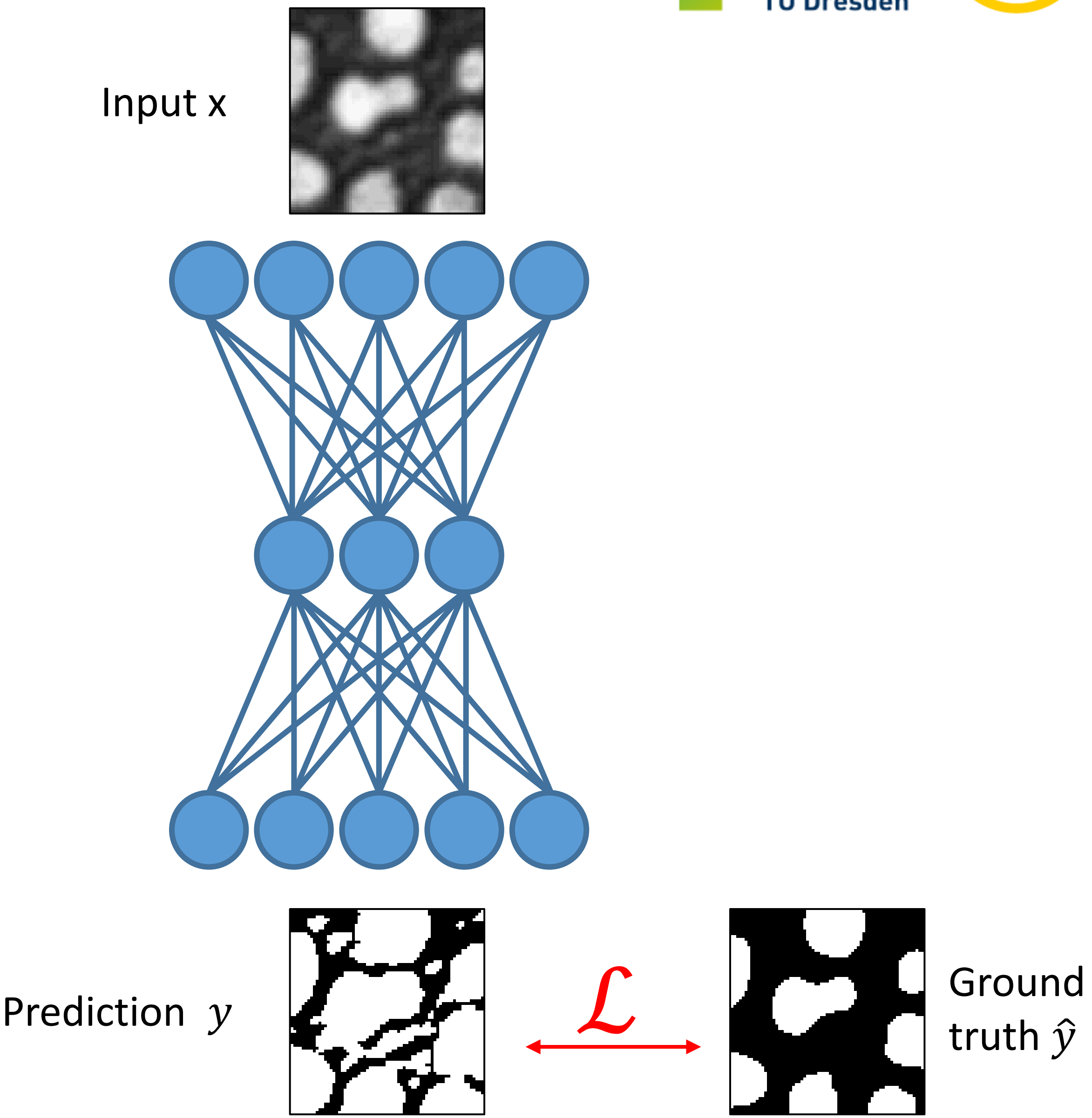
- Learning is an optimization problem

- Step 0: Initialize the network randomly
  - Weights
  - Bias
- Step 1: Forward pass the input through the network, get an initial prediction

- Step 2: Compare the output with the ground truth, compute the error (loss function)
  - The loss function can be freely defined.
  - Example: mean squared error

$$\mathcal{L}(y,\hat{y}) = \frac{1}{M}\sum_{i=1}^{M}(\hat{y}_i - y_i)^2$$

- Step 3: Update weights



Slide adapted from: Mahmood Nazari, TU Dresde

Back-Propagation Algorithm

The loss function can be expanded from

$$\mathcal{L}(y,\hat{y}) = \frac{1}{M}\sum_{i=1}^{M}(\hat{y}_i - y_i)^2$$

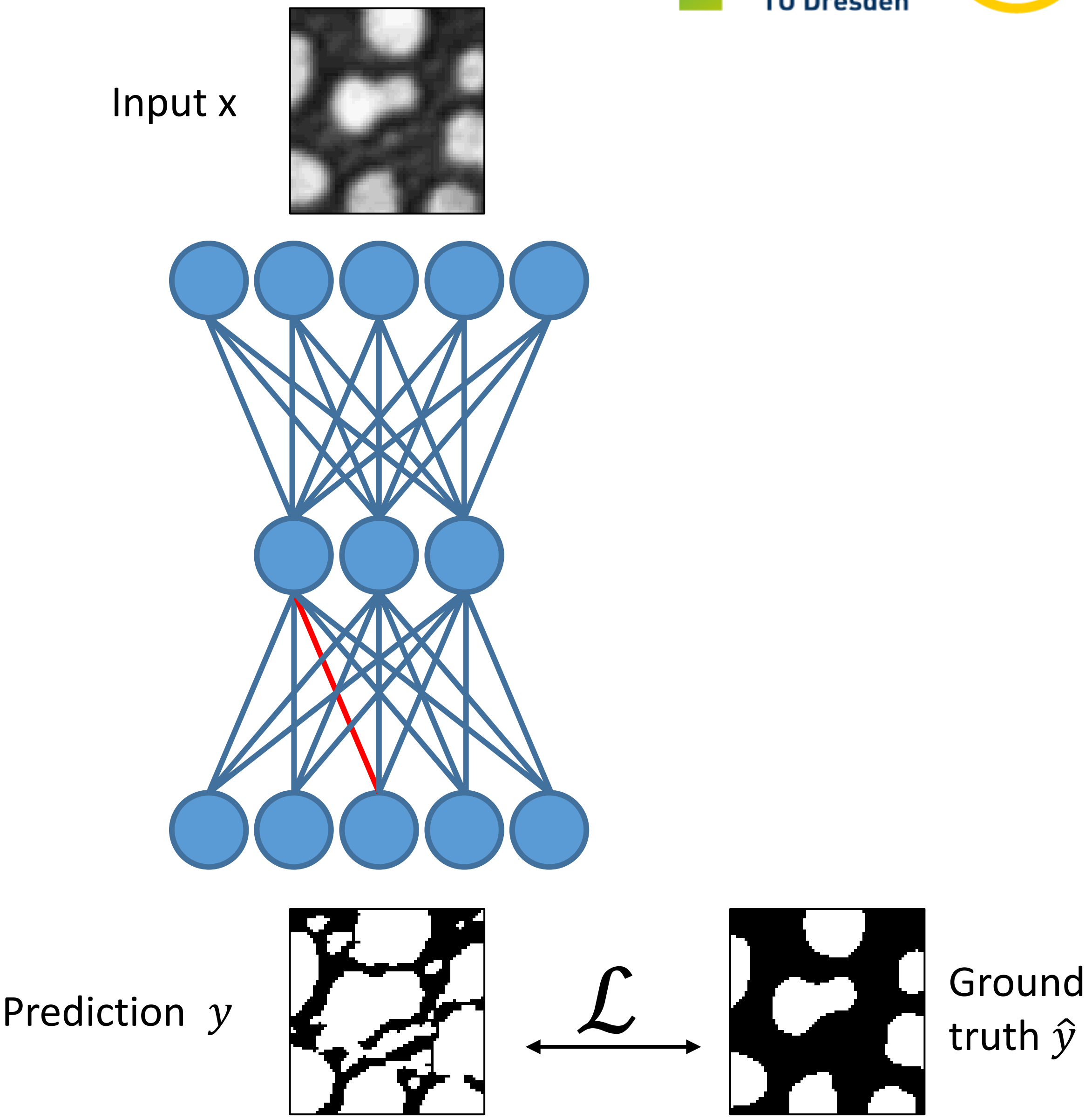
as the prediction depends on inputs x weights w and bias b

$$\mathcal{L}(\hat{y},x,w) = \frac{1}{M}\sum_{i=1}^{M}(\hat{y}_i - (w^T x_i + b))^2$$

Derivatives with respect to w and b:

$$\frac{\partial\mathcal{L}(\hat{y},x,w)}{\partial w} = -\frac{1}{N}\sum_{i=1}^{N}2x_i(\hat{y}_i - (w^T x_i + b))$$

$$\frac{\partial\mathcal{L}(\hat{y},x,w)}{\partial b} = -\frac{1}{N}\sum_{i=1}^{N}2(\hat{y}_i - (w^T x_i + b))$$



@haesleinhuepf

Slide adapted from : Mahmood Nazari, TU Dresd

Back-Propagation Algorithm

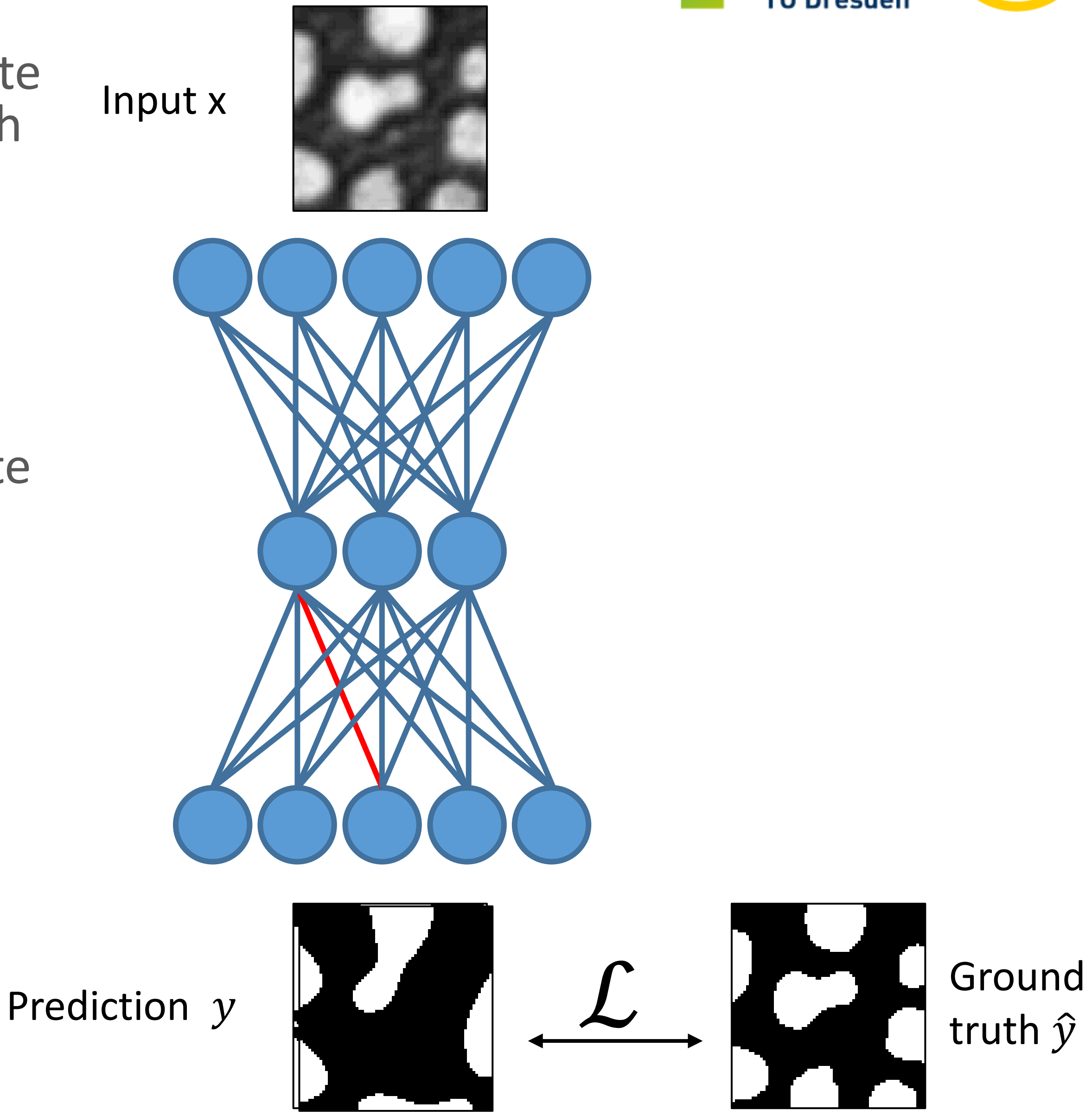
- The principle of the BackProp algorithm is to calculate the gradient of the loss function with respect to each trainable parameters of the network, i.e.

$$\frac{\partial \mathcal{L}}{\partial w_{ij}^{k}}$$

- where w_{ij}^{k} is the i:th weight of node j in layer k, which will allow the optimization algorithm to update the weights step by step using stochastic gradient descent

$$w_{ij}^{k} = w_{ij}^{k} + \eta \frac{\partial \mathcal{L}}{\partial w_{ij}^{k}}$$

- where η is the step length, in this context known as the **learning rate**.
  η can be varied during training (e.g. from epoc to epoc).
- *Epoc*: Updating all parameters considering all input/ground-truth pairs



@haesleinhuepf

Slide adapted from : Mahmood Nazari, TU Dresd

Model validation

Train dataset (e.g. 80% of the data)

- Used for training directly

Validation dataset (10% of the data)

- After every iteration see if the model overfits

Test dataset (10% of the data)

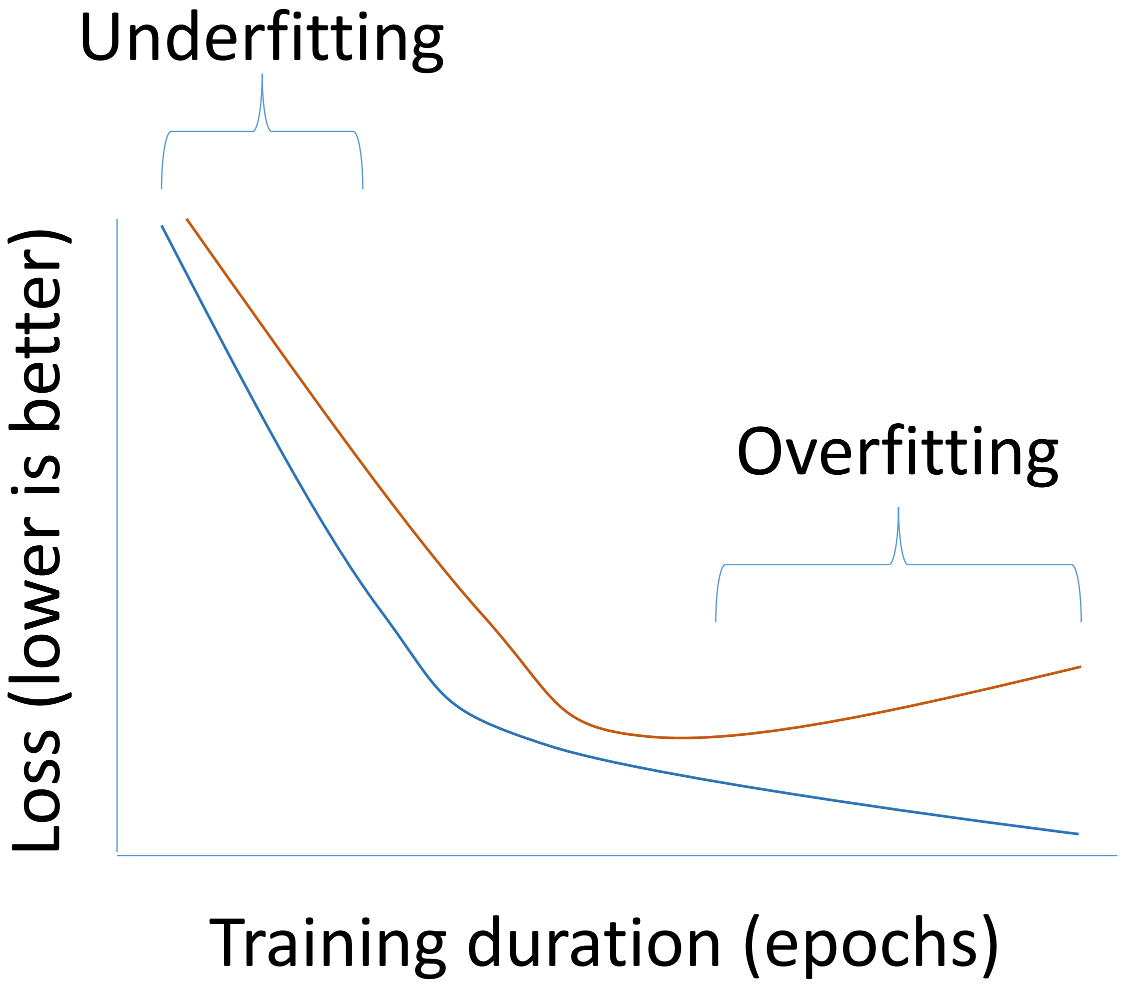
- Final evaluation after training is finished (once)

Underfitting

- A trained model that is not even able to properly process the data it was trained on

Overfitting

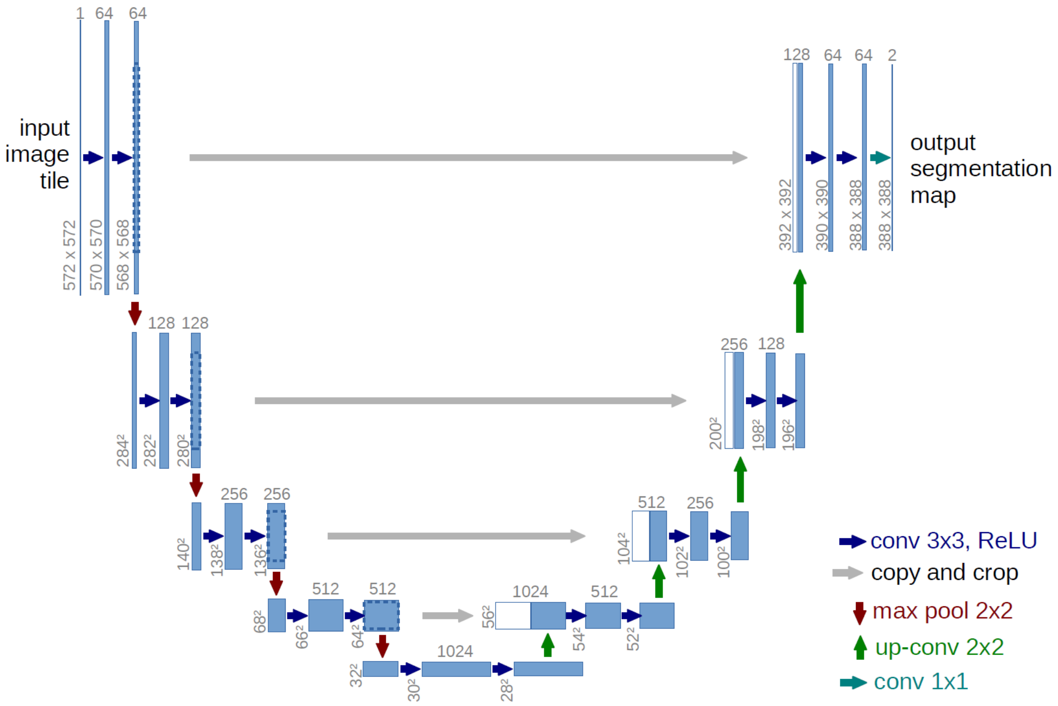
- A model that is able to process data it was trained on well
- It processes other data poorly



https://towardsdatascience.com/how-to-split-data-into-three-sets-train-validation-and-test-and-why-e50d22d3e54c

The U-net

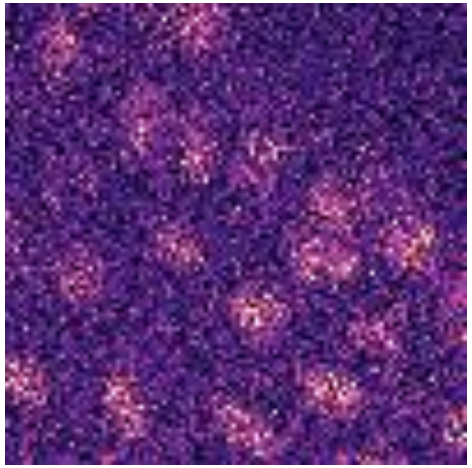
- The U-net is a very common DNN architecture in biological image processing.
  - Contraction: Increase the “What”, decrease the “Where”
  - Expansion: Increase the “Where”, decrease the “What”

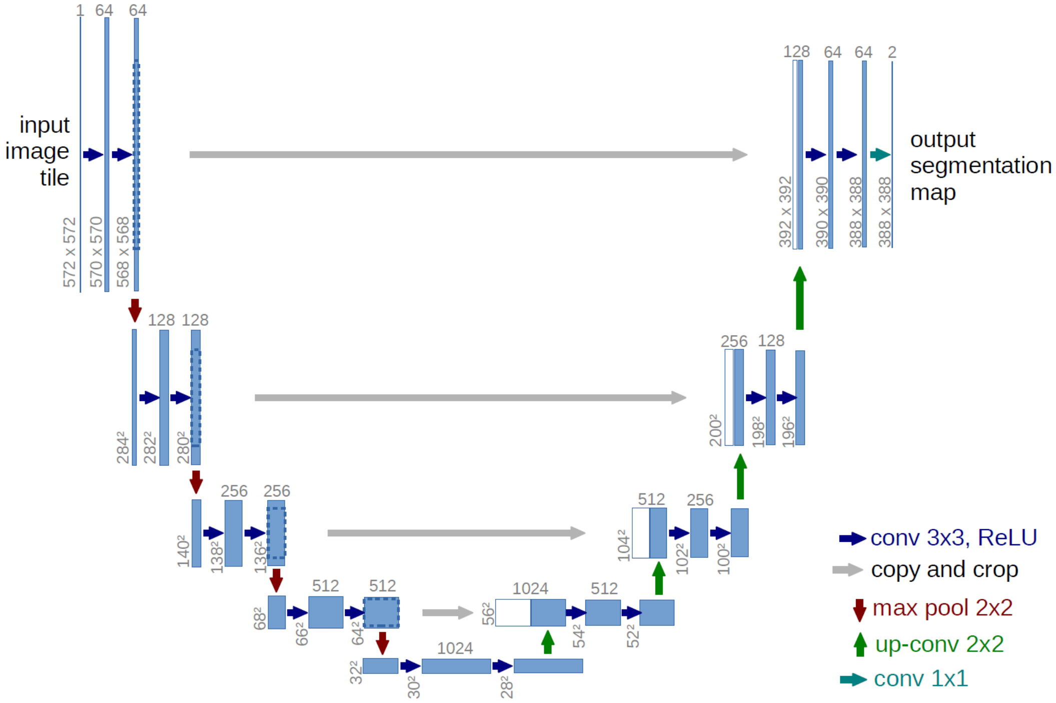


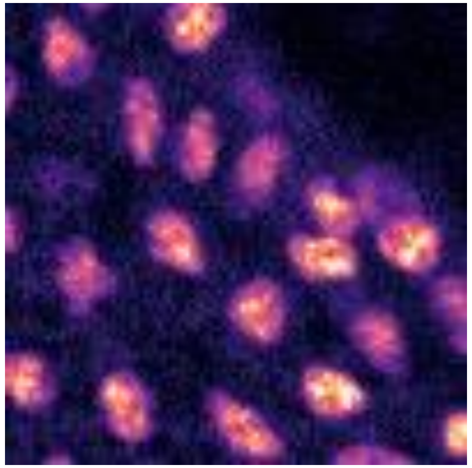
Sources: Ronneberger (2015) https://arxiv.org/pdf/1505.04597.pdf
Johannes Soltwedel, OncoRay, TU Dresden

Image denoising: CARE

- Content aware image restoration (CARE)
- Image acquisition of pairs of images: A high-quality and a low-quality image.
- Problem: shot noise, biology moves!







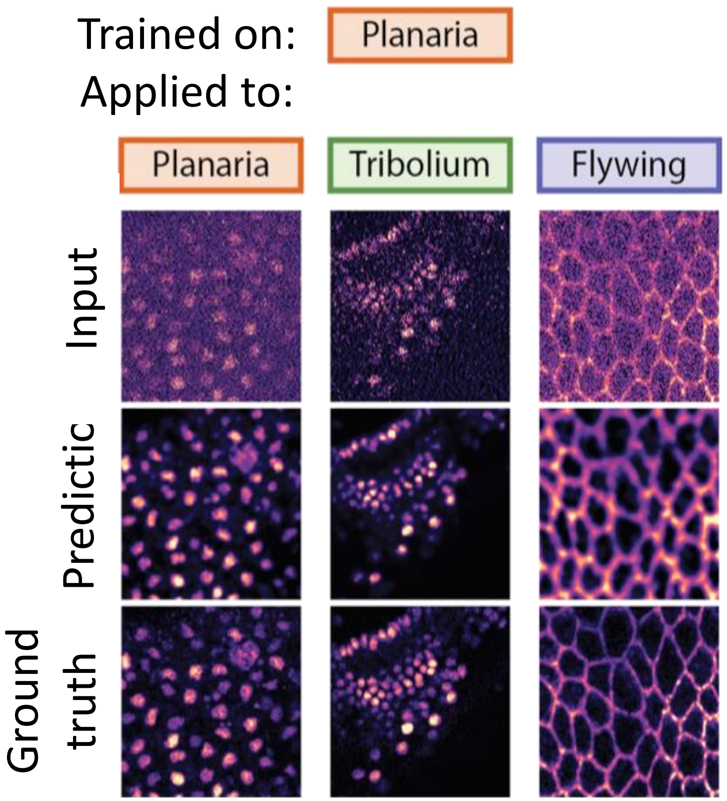
@haesleinhuepf

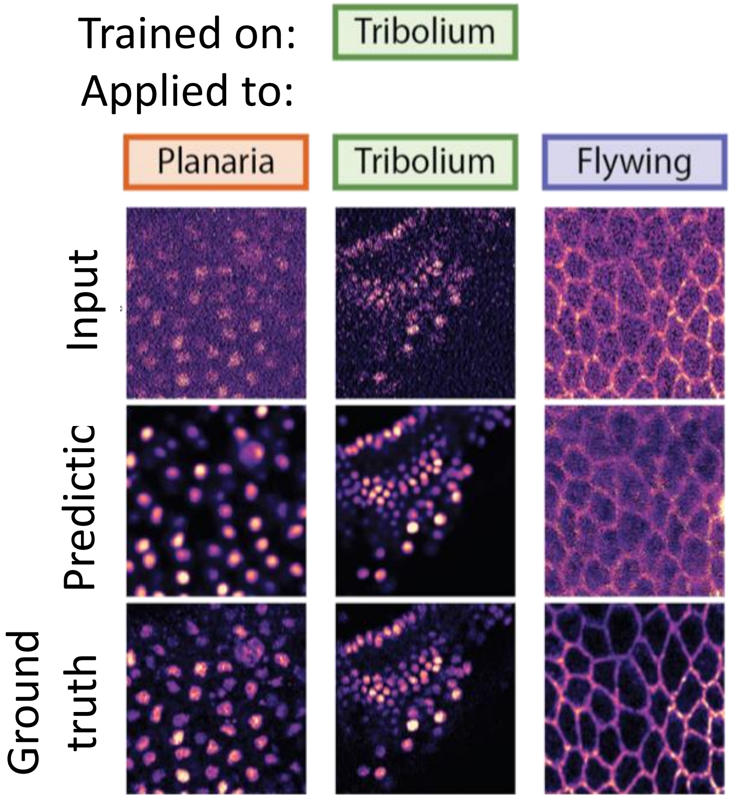
Source: Ronneberger (2015) https://arxiv.org/pdf/1505.04597.pdf

Slide adapted from: Martin Weigert, EPFL Lausanne

Image denoising: CARE

- Important to use on the same conditions/structures/staining that the networks were trained on!

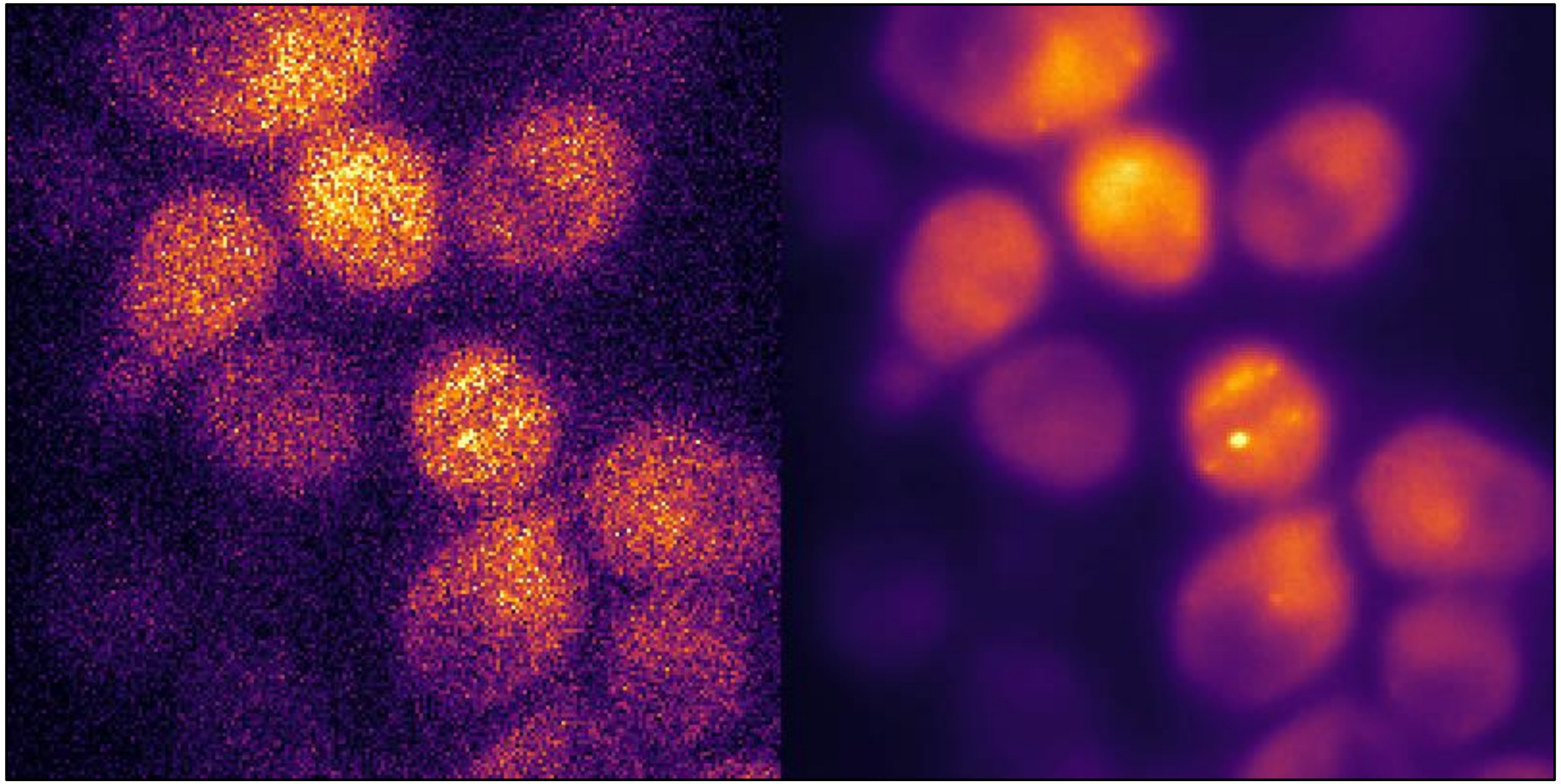




Slide adapted from: Martin Weigert, EPFL Lausanne

Image denoising

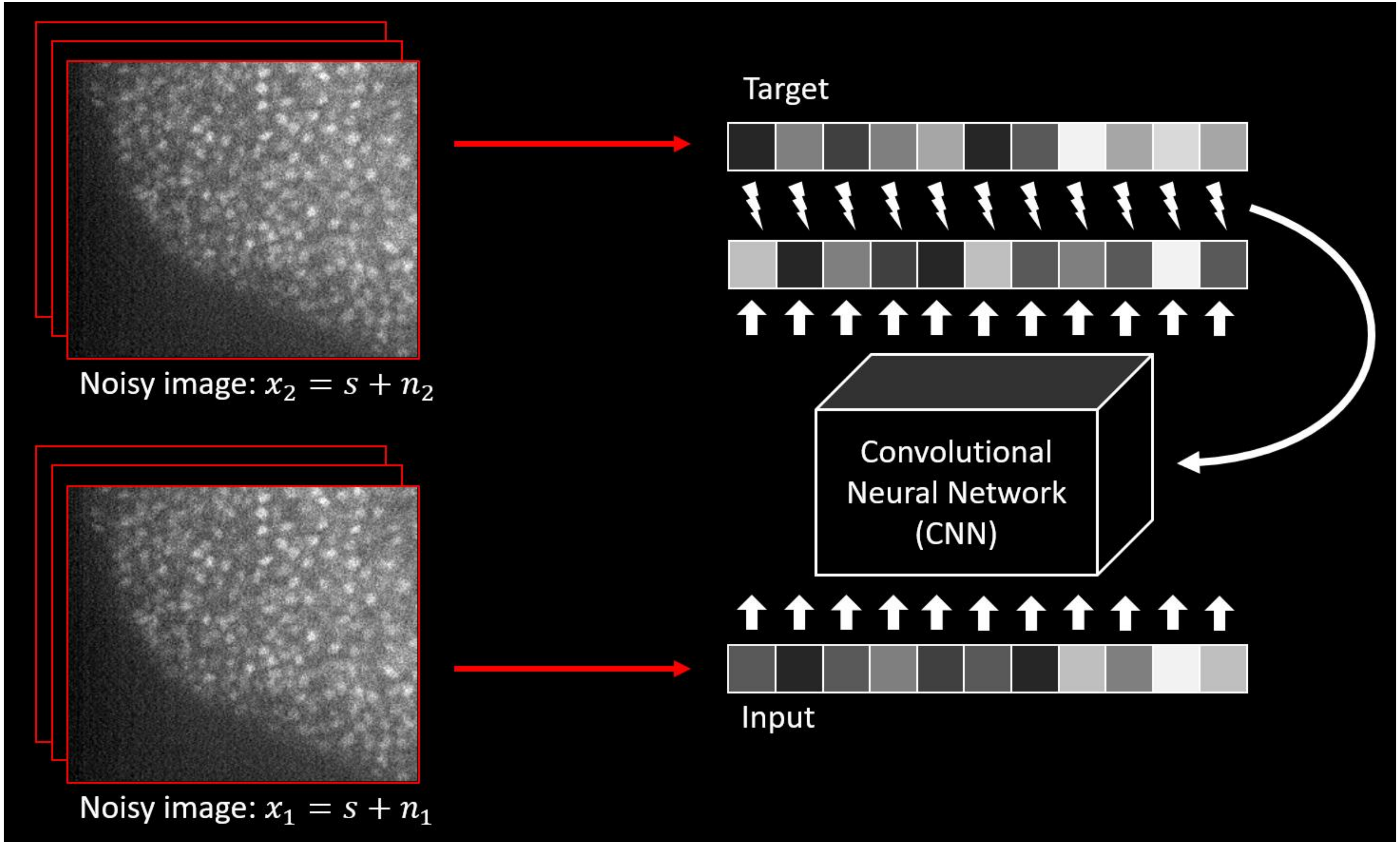
- Noise2Void



Data by Stephanie Heinrich
Slide adapted from: Alexander Krull, MPI CBG

Image denoising

- Noise2Noise by NVidia (Lehtinen 2018) https://arxiv.org/pdf/1803.04189.pdf



Noisy image: $x_2 = s + n_2$

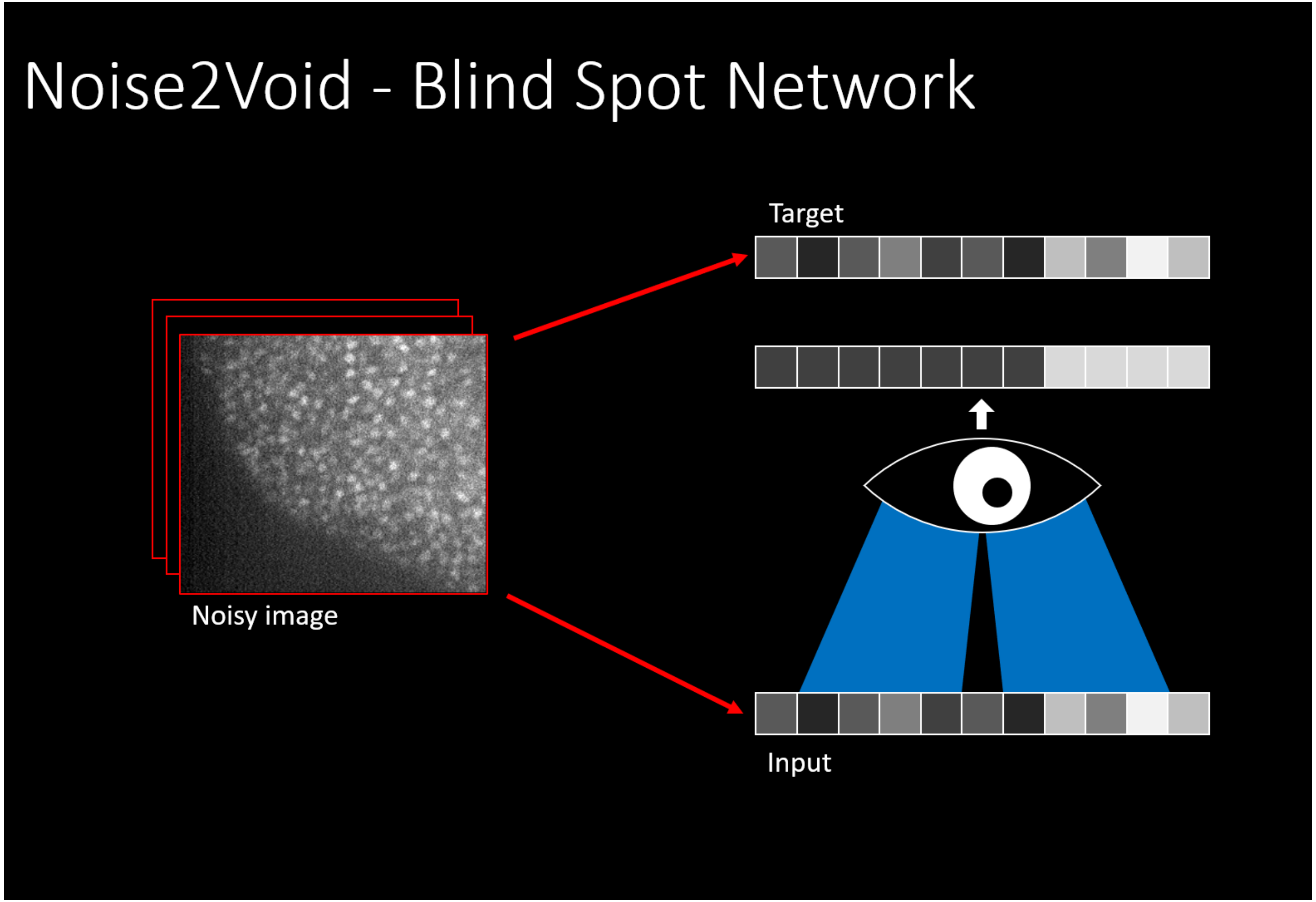
Noisy image: $x_1 = s + n_1$

@haesleinhuepf

Slide adapted from Alexander Krull (MPI CBG, now at U. Birmingham)

Image denoising

- Noise2void: Krull et al (2019) https://arxiv.org/abs/1811.10980
- Noise2self: Batson and Royer (2019) https://arxiv.org/abs/1901.11365

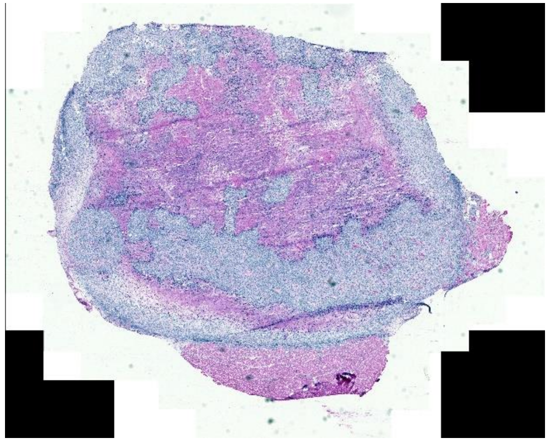

Noise2Void - Blind Spot Network

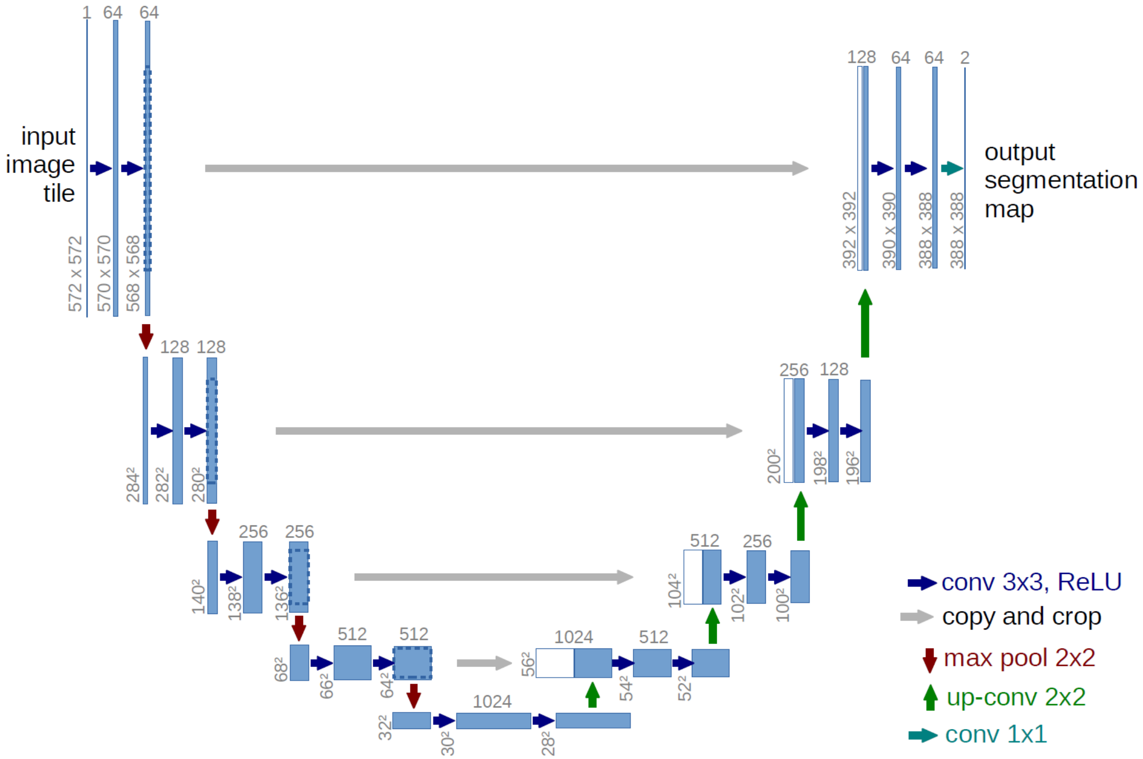
@haesleinhuepf

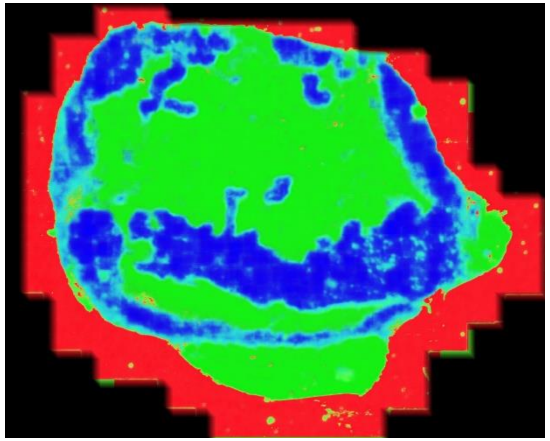
Slide adapted from Alexander Krull (MPI CBG, now at U. Birmingham)

Image segmentation

- U-net for pixel classification
  -> Semantic segmentation
  -> Probability maps





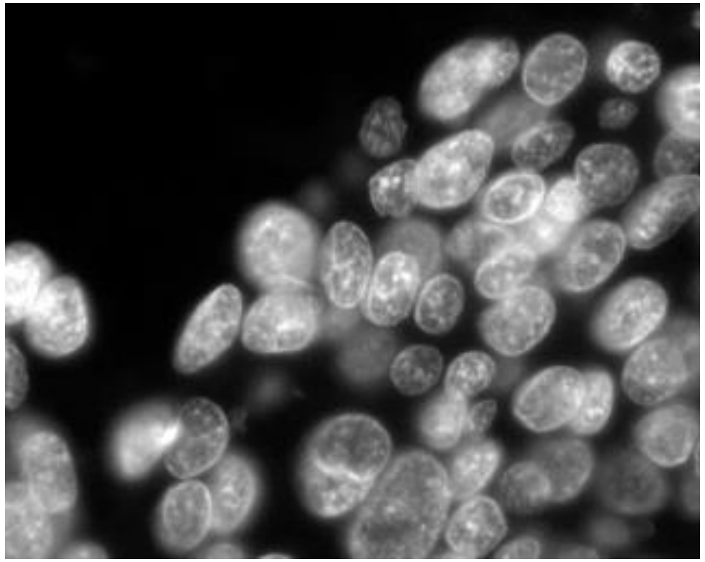


@haesleinhuepf

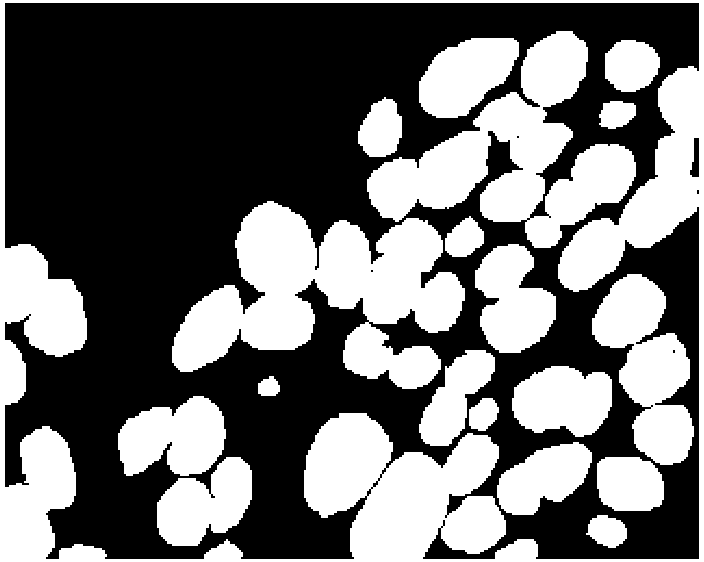
Sources: Ronneberger (2015) https://arxiv.org/pdf/1505.04597.pdf
Johannes Soltwedel, OncoRay, TU Dresden

Nuclei/cell segmentation

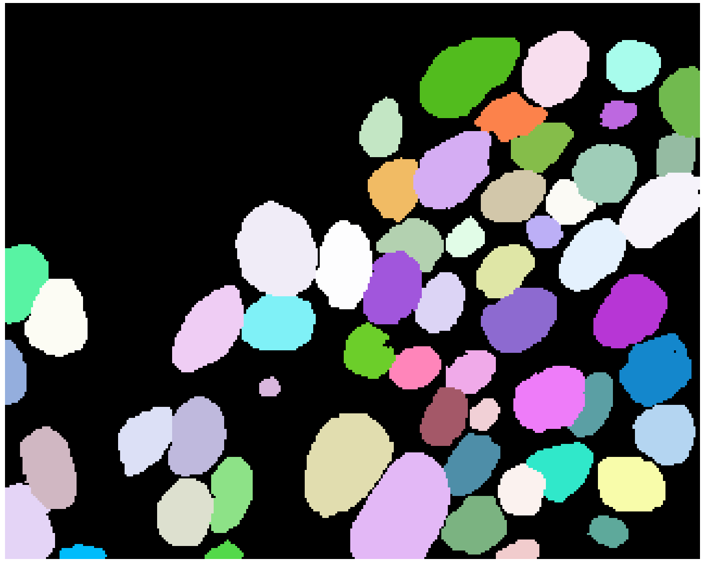
- Recap: Terminology



Input Image



Semantic Segmentation
(foreground/background)



Instance Segmentation
(individual cells)

Slide adapted from: Martin Weigert, EPFL Lausanne

CellPose

- Cell/Nuclei – segmentation based on flow-fields
- Technically similar to Watershed, but with a deep-learning based altitude-image

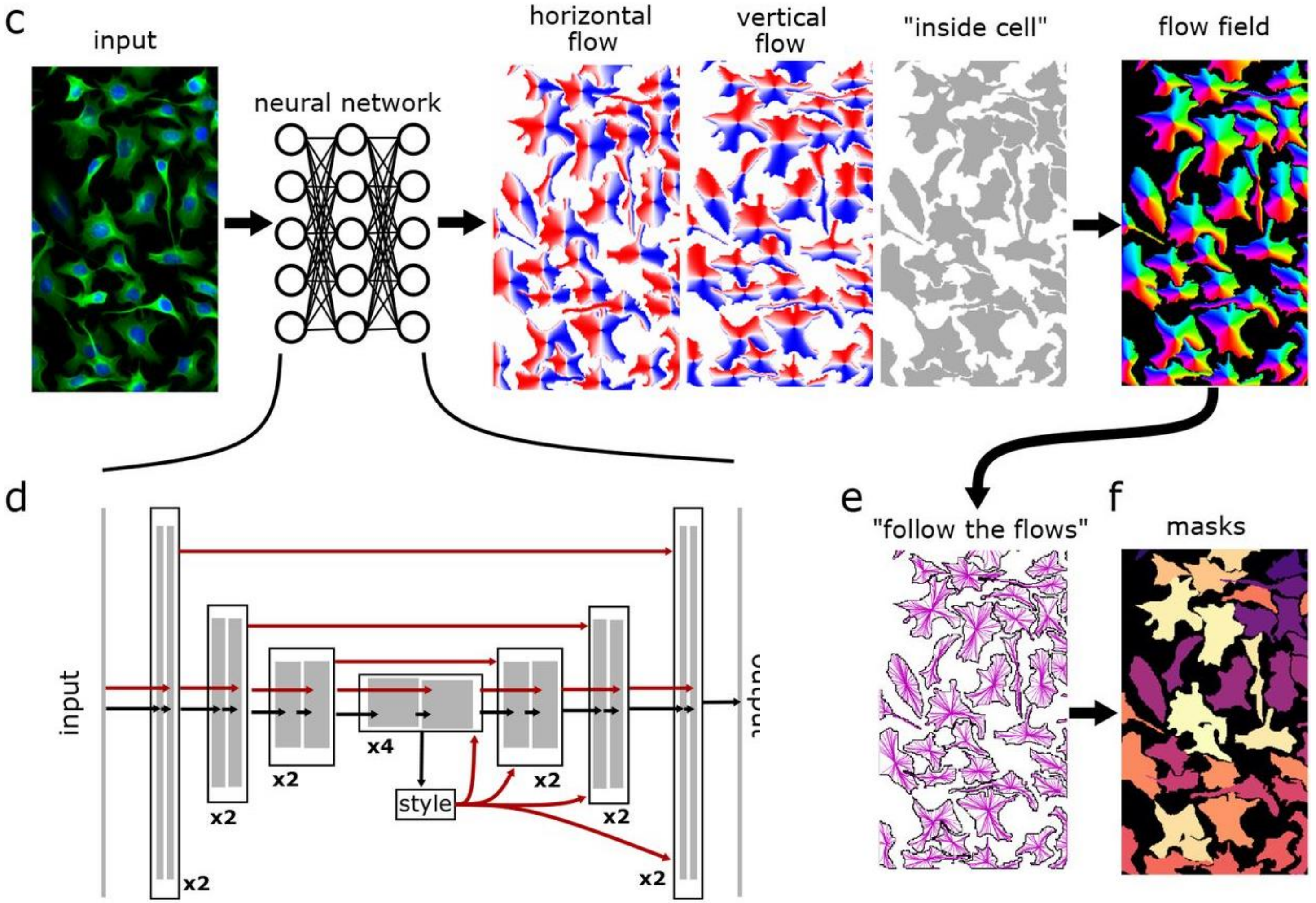
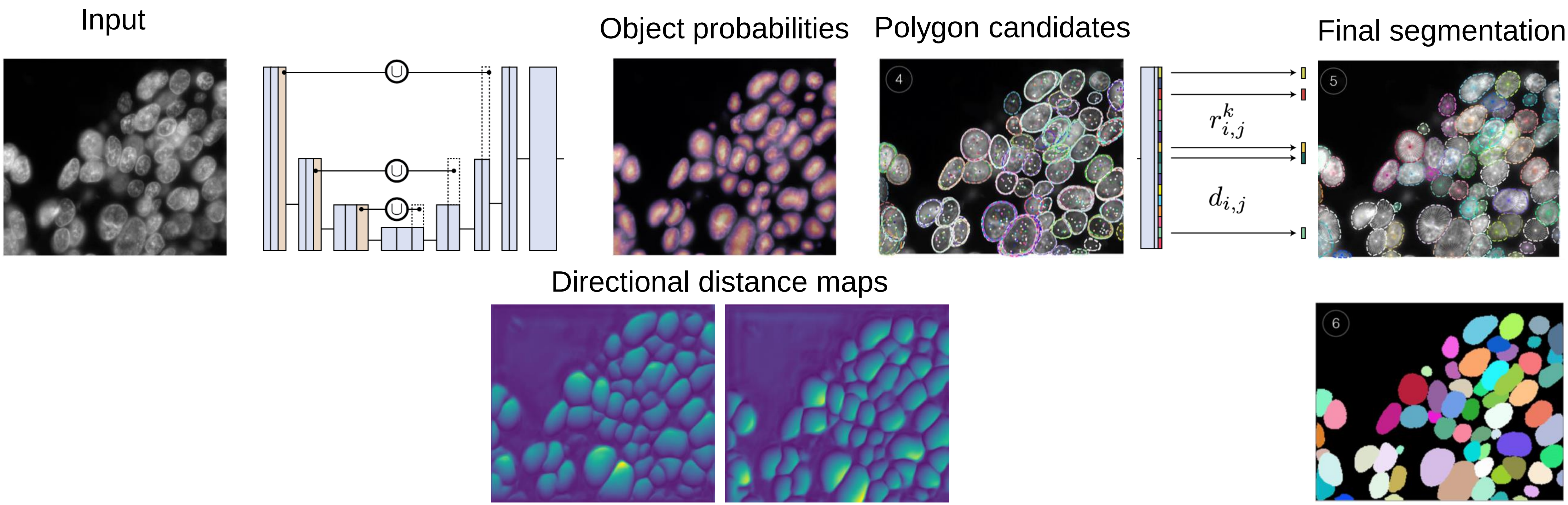


Image source: Modified from Stringer et al (2020) licensed [CC-BY-NC 4.0](https://creativecommons.org/licenses/by-nc/4.0/)
https://www.biorxiv.org/content/10.1101/2020.02.02.931238v2.full

StarDist

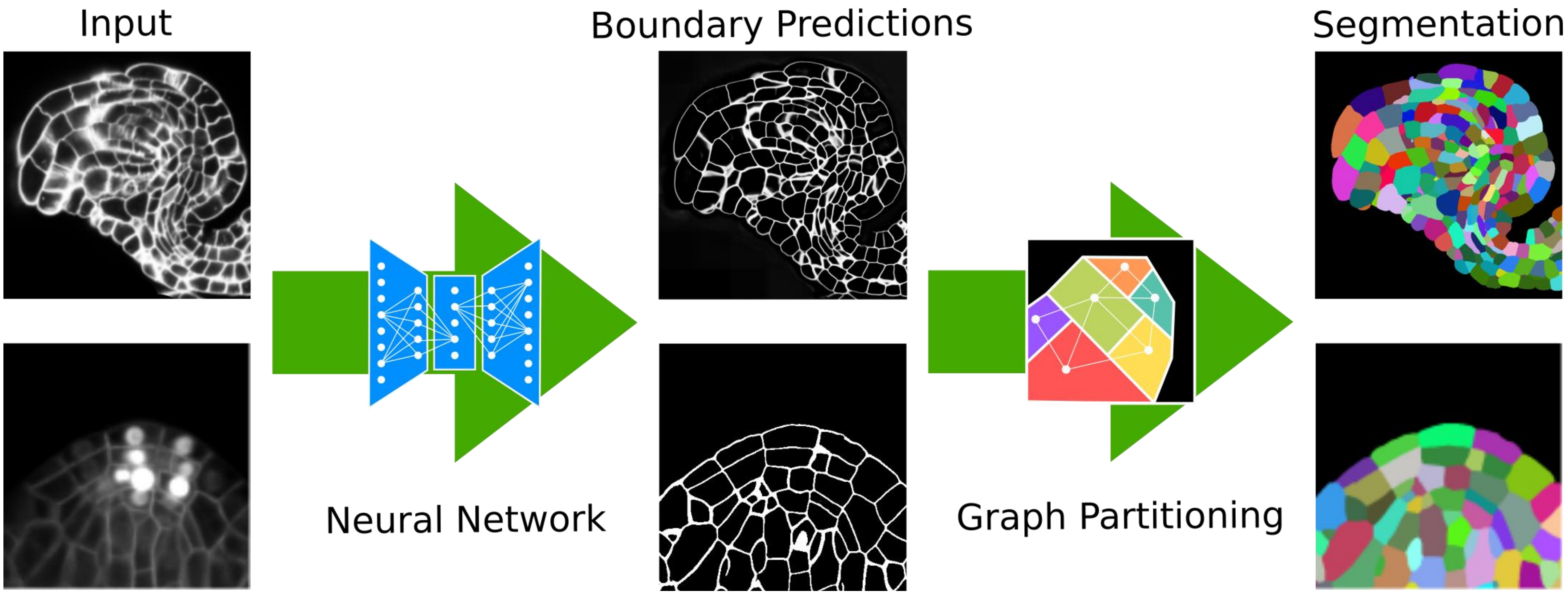
- Prediction of probable object centers + polygon outlines
- Non-maximum-suppression of less likely polygons



Slide adapted from: Martin Weigert, EPFL Lausanne and from https://github.com/stardist/stardist (license: BSD3)

PlantSeg

- Combination of neural networks + graph partitioning



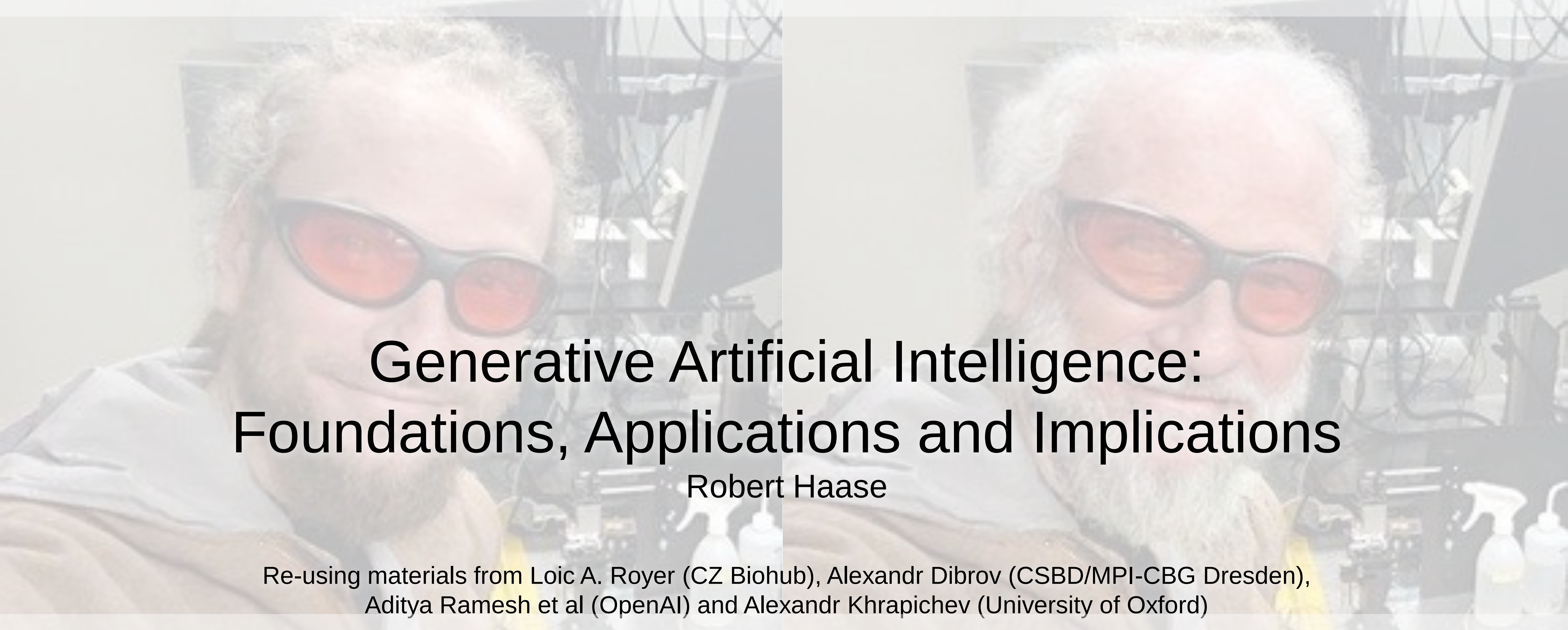
Source: Modified from https://github.com/hci-unihd/plant-seg

Summary: Deep Learning for Bio-image Analysis

- [Convolutional] Neural Networks is a decade old technology that enabled breakthroughs in various fields during the last decade. Examples in Microscopy:
  - Image Denoising
  - Image Segmentation
- Common scheme: Smart algorithms for processing input/output of neural networks
  - ~~Image in, segmentation out~~

- Deep-Learning based often represent state-of-the-art techniques with respect to result quality
- Training these models is
  - computationally expensive,
  - needs large amounts of training data (~~single images~~),
  - requires a certain level of expertise

- If Otsu-Thresholding + Connected Component Labeling does the job, don’t dive into deep learning!



Generative Artificial Intelligence: Foundations, Applications and Implications

Robert Haase

Re-using materials from Loic A. Royer (CZ Biohub), Alexandr Dibrov (CSBD/MPI-CBG Dresden), Aditya Ramesh et al (OpenAI) and Alexandr Khrapichev (University of Oxford)

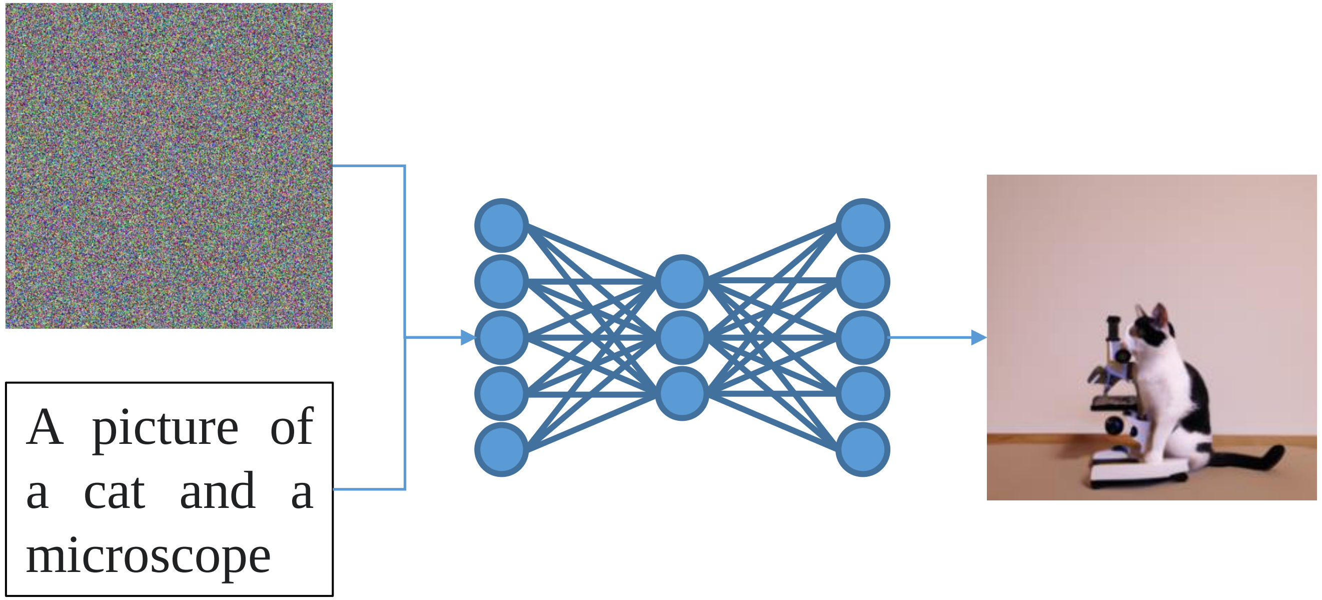
@haesleinhuepf

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



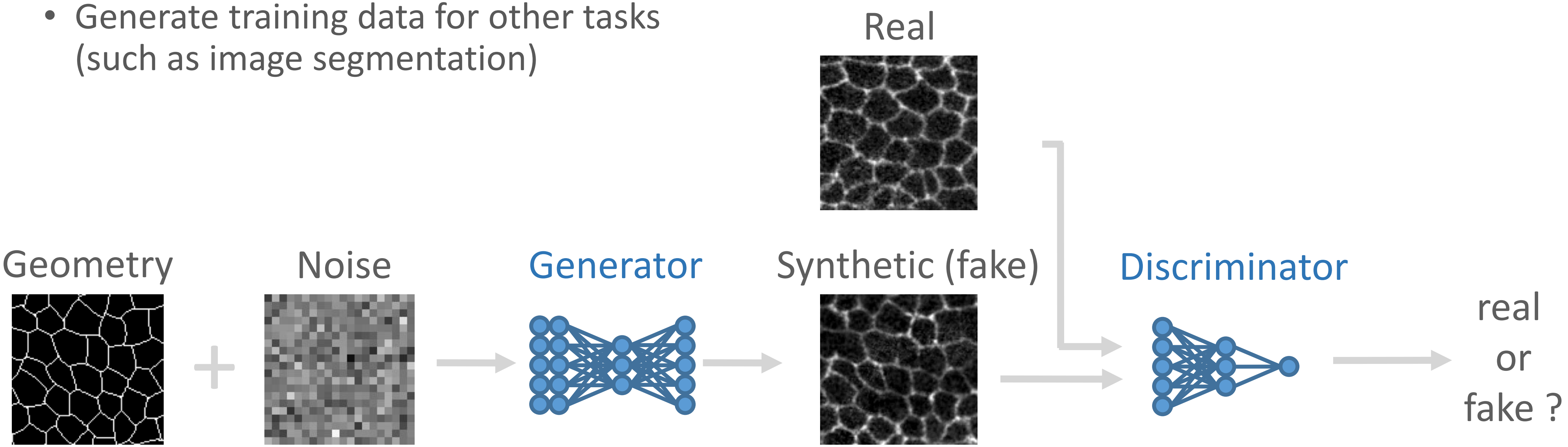
A picture of a cat and a microscope

---

[1] source: https://en.wikipedia.org/wiki/Generative_artificial_intelligence

How does it work?

- Combination of neural networks
- Example: Generative Adversarial Networks (GANs)
- Use-cases:
  - Generate training data for other tasks (such as image segmentation)



Real

Geometry

Noise

Generator

Synthetic (fake)

Discriminator

real

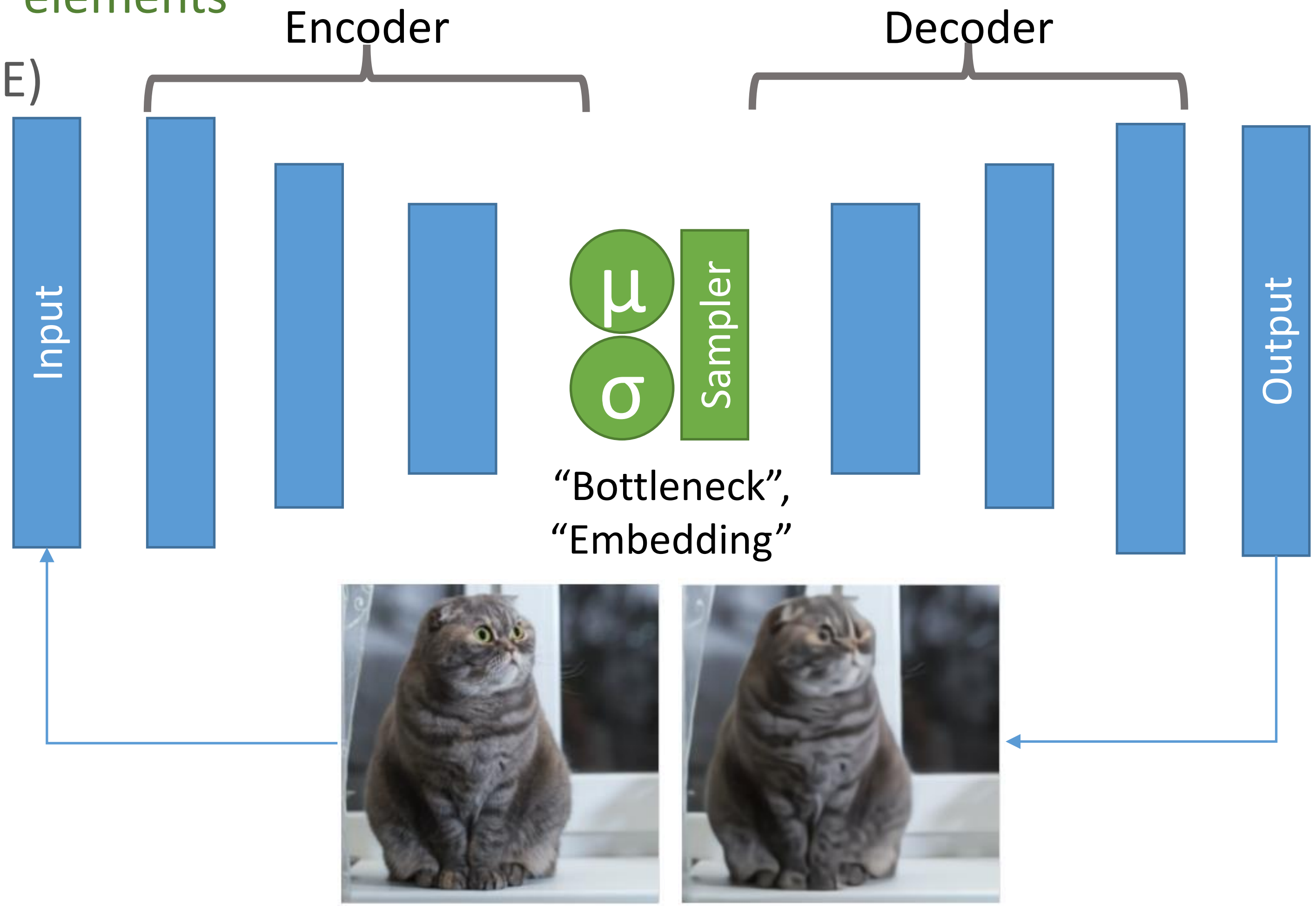
or

fake ?

Modified from Alexander Dibrov (Myers lab, CSBD / MPI-CBG)

How does it work?

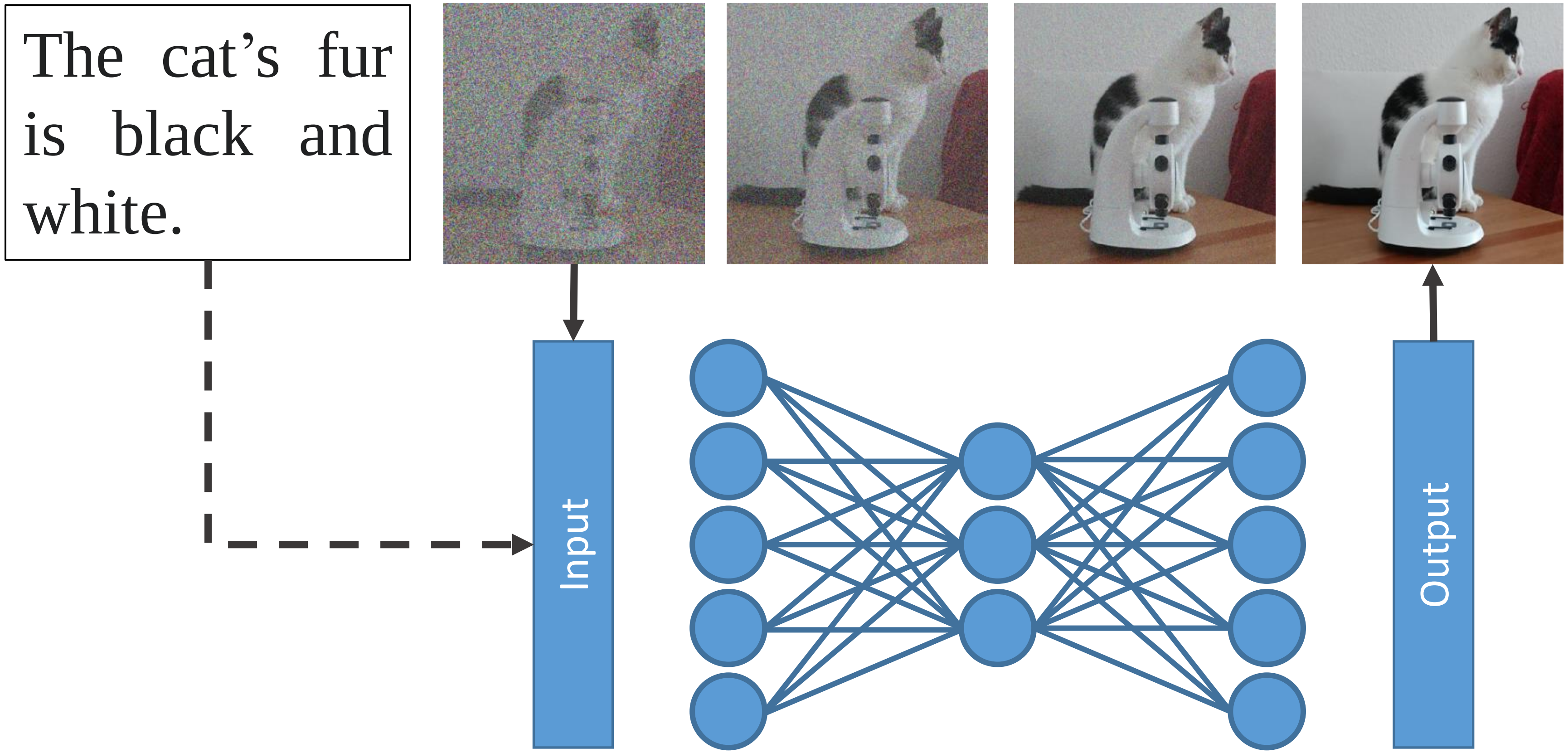
- Combination of neural networks + other elements
- Example: Variational Auto-Encoders (VAE)
- Use-cases:
  - Combine images (cat, microscope)



Cat picture source: Ramesh et al., licensed CC-BY 4.0
https://arxiv.org/pdf/2102.12092.pdf

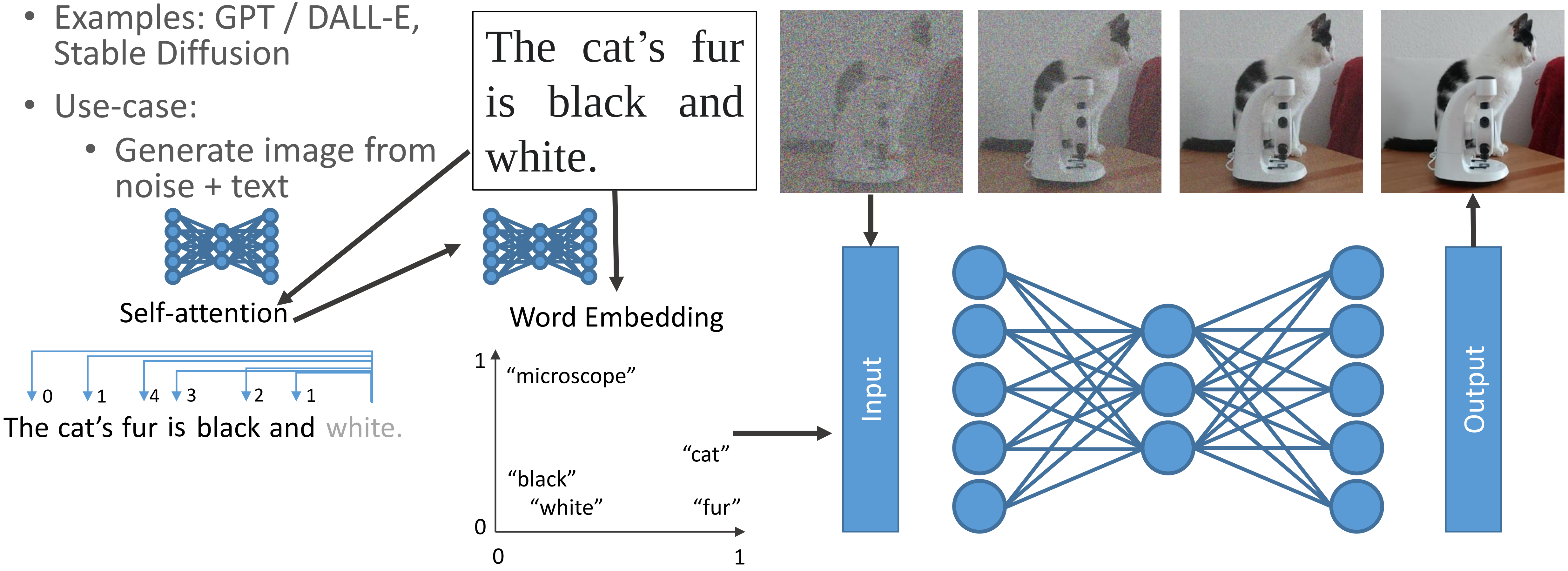
How does it work?

- Combination of neural networks + other elements + various data sources
- Examples: GPT / DALL-E, Stable Diffusion
- Use-case:
  - Generate image from noise + text



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

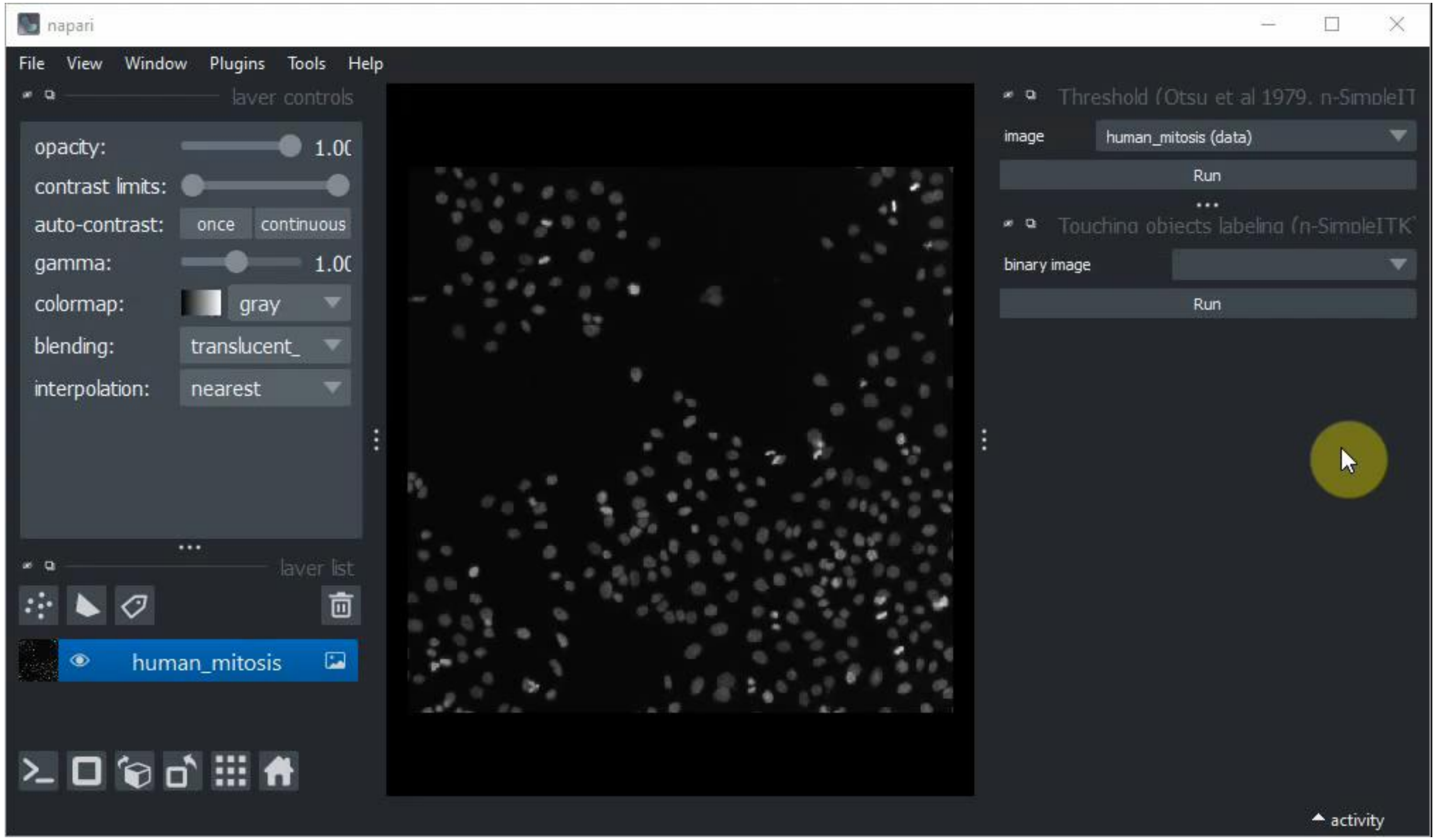


Deconstruction

- Deconstruction is a method in software engineering to
  - understand how existing software works
  - prevent reinventing the wheel
  - identify limitations
  - identify bottlenecks
- Related methods
  - Reverse engineering
  - Code review
  - Pair programming
- In the age of computers writing code, reading code is a key skill.

Deconstruction of napari-chatGPT

- Napari is a Python-based image viewer, extensible via plugins / “widgets”.



opacity: 1.0

contrast limits:

auto-contrast: once continuous

gamma: 1.0

colormap: gray

blending: translucent_

interpolation: nearest

human_mitosis

Threshold (Otsu et al 1979, n-SimpleIT

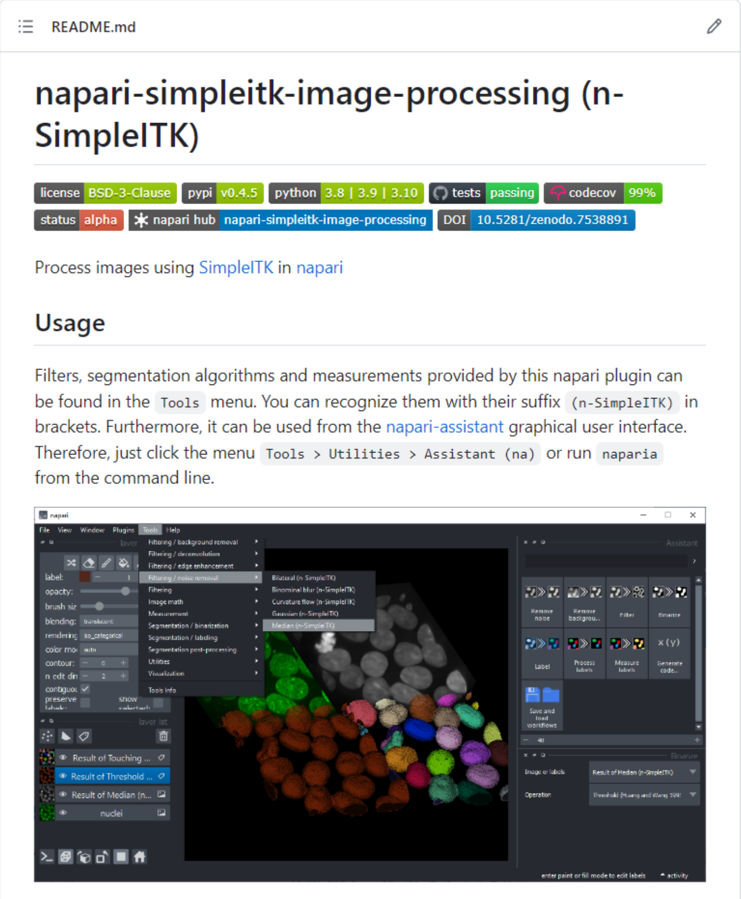
image: human_mitosis (data)

Run

Touching objects labeling (n-SimpleITK

binary image:

Run



napari-simpleitk-image-processing (n-SimpleITK)

license BSD-3-Clause | pypi v0.4.5 | python 3.8 | 3.9 | 3.10 | tests passing | codecov 99%
status alpha | napari hub napari-simpleitk-image-processing | DOI 10.5281/zenodo.7538891

Process images using SimpleITK in napari

Usage

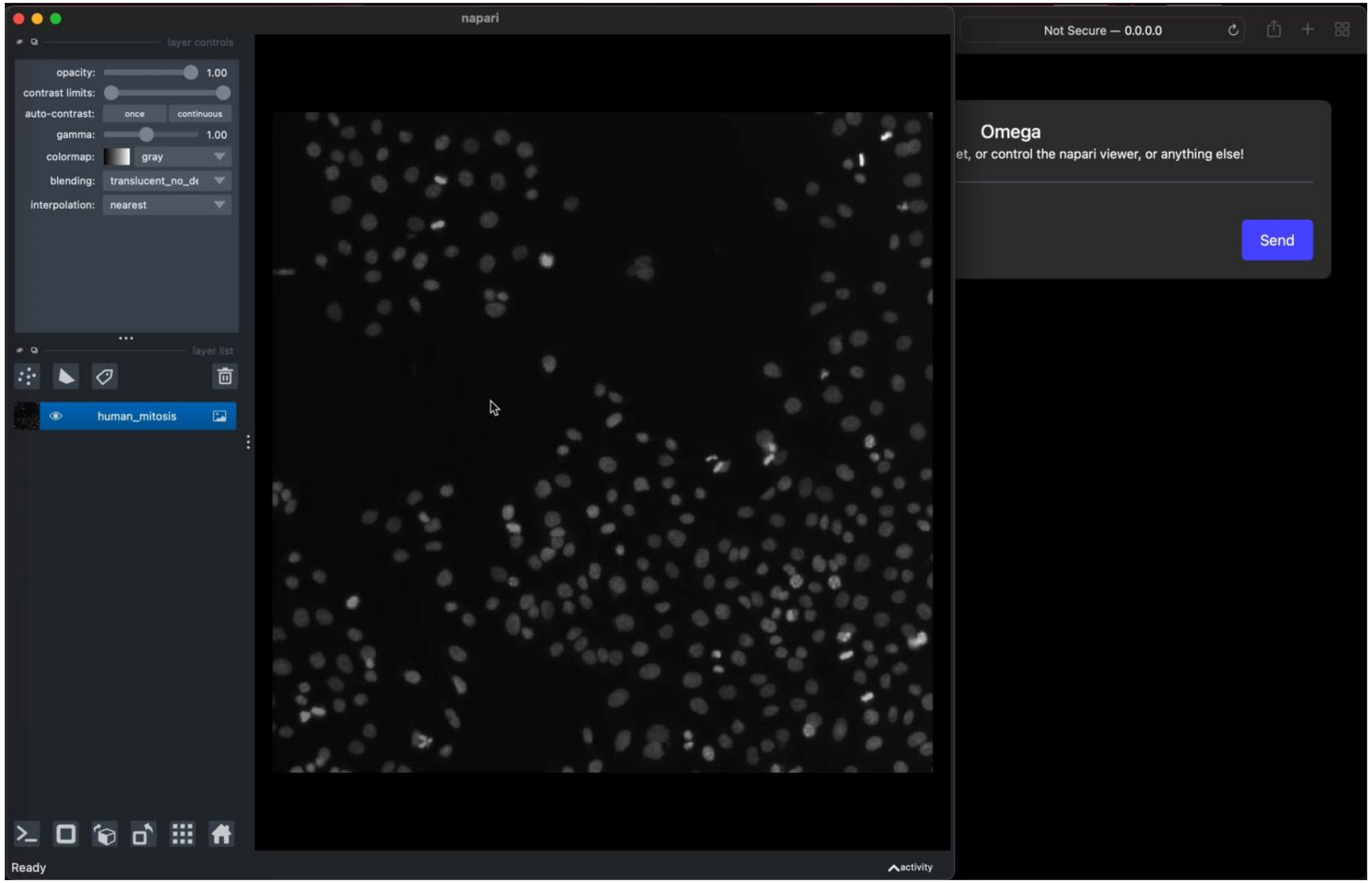
Filters, segmentation algorithms and measurements provided by this napari plugin can be found in the `Tools` menu. You can recognize them with their suffix `(n-SimpleITK)` in brackets. Furthermore, it can be used from the napari-assistant graphical user interface. Therefore, just click the menu `Tools > Utilities > Assistant (na)` or run `naparia` from the command line.

@haesleinhuepf

https://napari.org/
https://github.com/haesleinhuepf/napari-simpleitk-image-processing

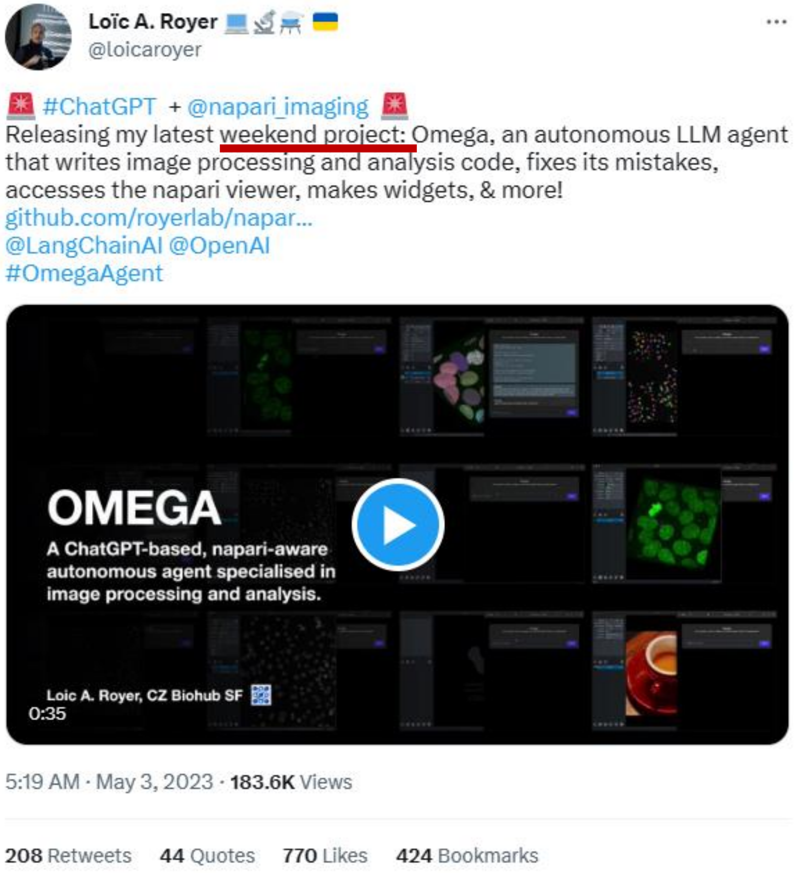
Deconstruction of napari-chatGPT

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



opacity: 1.00
contrast limits:
auto-contrast: once continuous
gamma: 1.00
colormap: gray
blending: translucent_no_de
interpolation: nearest

Omega
et, or control the napari viewer, or anything else!



Loïc A. Royer 💻🔬🪑🇺🇦
@loicaroyer

🚨 #ChatGPT + @napari_imaging 🚨
Releasing my latest weekend project: Omega, an autonomous LLM agent that writes image processing and analysis code, fixes its mistakes, accesses the napari viewer, makes widgets, & more!
github.com/royerlab/napar...
@LangChainAI @OpenAI
#OmegaAgent

OMEGA
A ChatGPT-based, napari-aware autonomous agent specialised in image processing and analysis.
Loic A. Royer, CZ Biohub SF
0:35

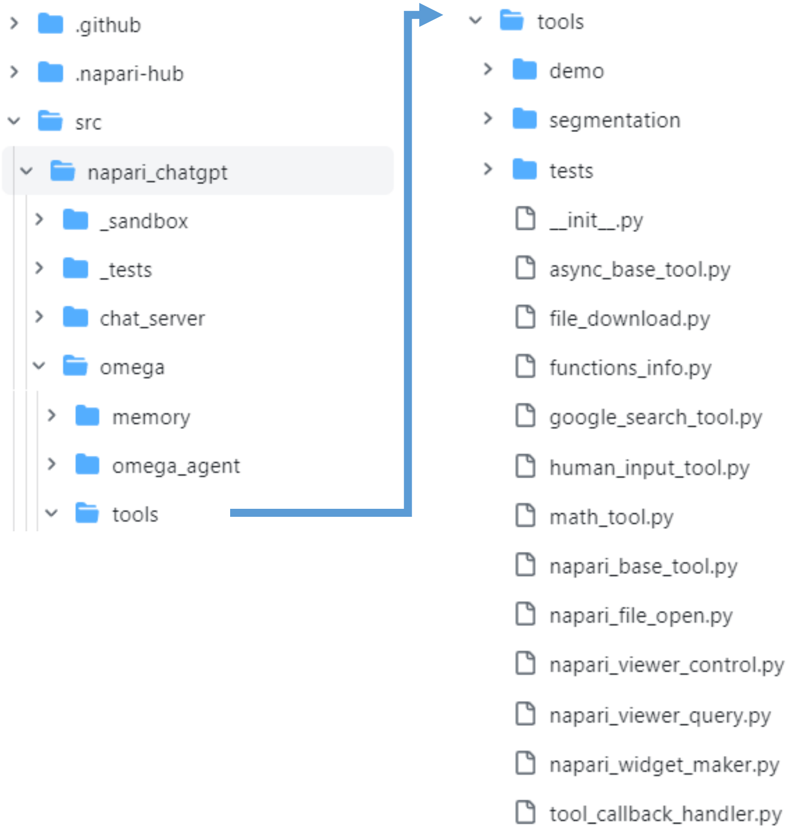
5:19 AM · May 3, 2023 · **183.6K** Views

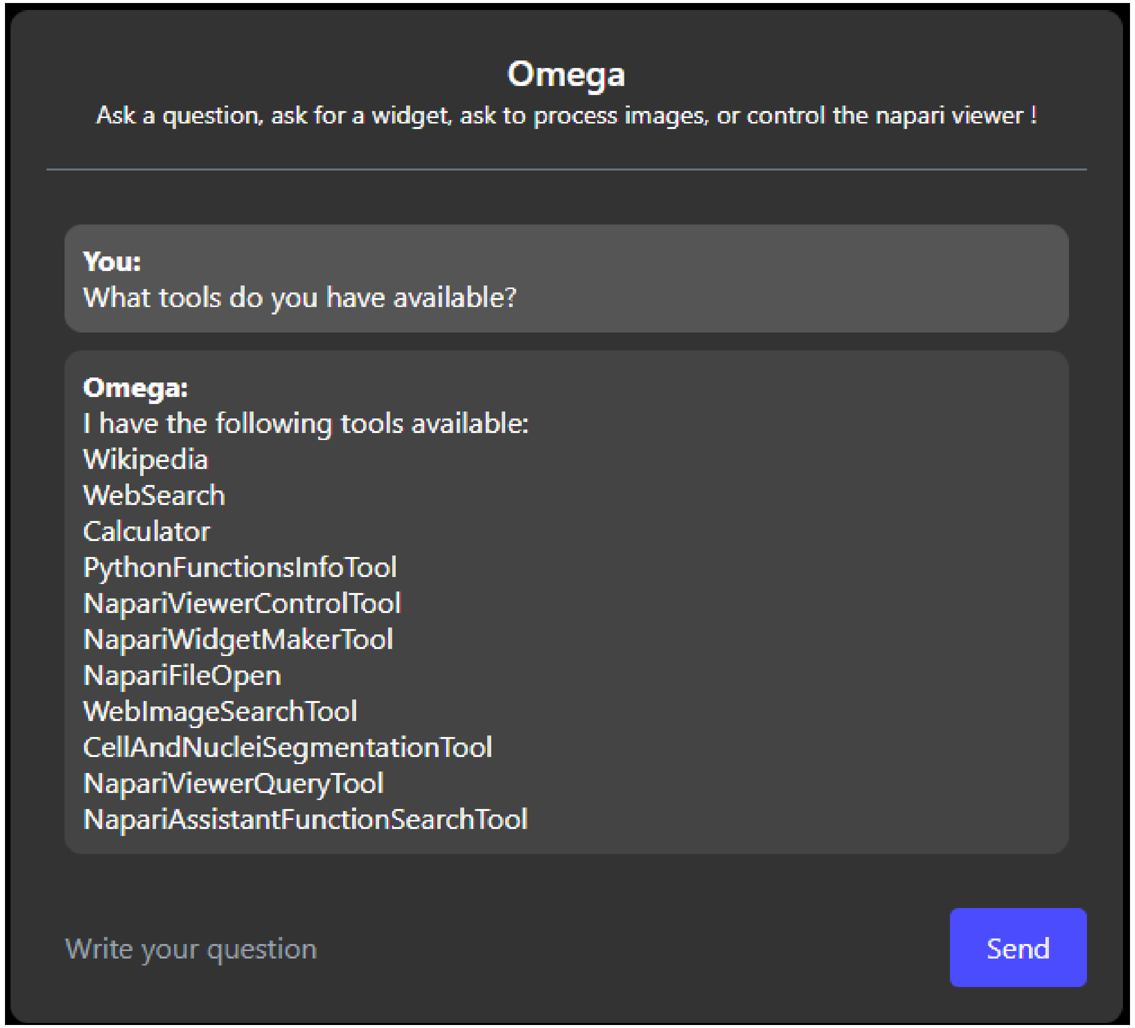
208 Retweets **44** Quotes **770** Likes **424** Bookmarks

Source: https://github.com/royerlab/napari-chatgpt (BSD3 license)
https://twitter.com/loicaroyer/status/1653600252807757824

Deconstruction of napari-chatGPT

- Napari-chatGPT defines a list of "tools"




Omega
Ask a question, ask for a widget, ask to process images, or control the napari viewer !

You:
What tools do you have available?

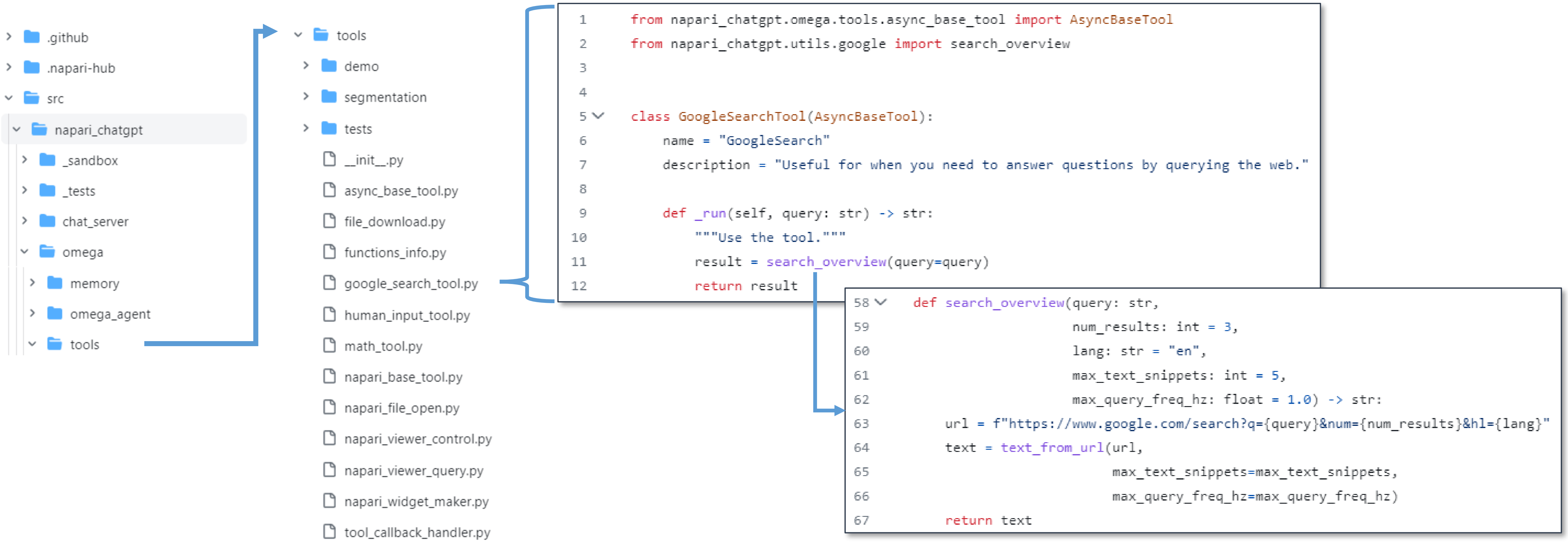
Omega:
I have the following tools available:
Wikipedia
WebSearch
Calculator
PythonFunctionsInfoTool
NapariViewerControlTool
NapariWidgetMakerTool
NapariFileOpen
WebImageSearchTool
CellAndNucleiSegmentationTool
NapariViewerQueryTool
NapariAssistantFunctionSearchTool

@haesleinhuepf

https://github.com/royerlab/napari-chatgpt/blob/aa00b48d0523da588ece5dc8cdb006c895c4b473/src/napari_chatgpt/omega/tools/google_search_tool.py
https://github.com/royerlab/napari-chatgpt/blob/aa00b48d0523da588ece5dc8cdb006c895c4b473/src/napari_chatgpt/utils/google.py#L58
Code origin: https://github.com/Nv7-GitHub/googlesearch

Deconstruction of napari-chatGPT

- Napari-chatGPT defines a list of “tools”



```
> .github
> .napari-hub
v src
  v napari_chatgpt
    > _sandbox
    > _tests
    > chat_server
    v omega
      > memory
      > omega_agent
      v tools
```

```
v tools
  > demo
  > segmentation
  > tests
    __init__.py
    async_base_tool.py
    file_download.py
    functions_info.py
    google_search_tool.py
    human_input_tool.py
    math_tool.py
    napari_base_tool.py
    napari_file_open.py
    napari_viewer_control.py
    napari_viewer_query.py
    napari_widget_maker.py
    tool_callback_handler.py
```

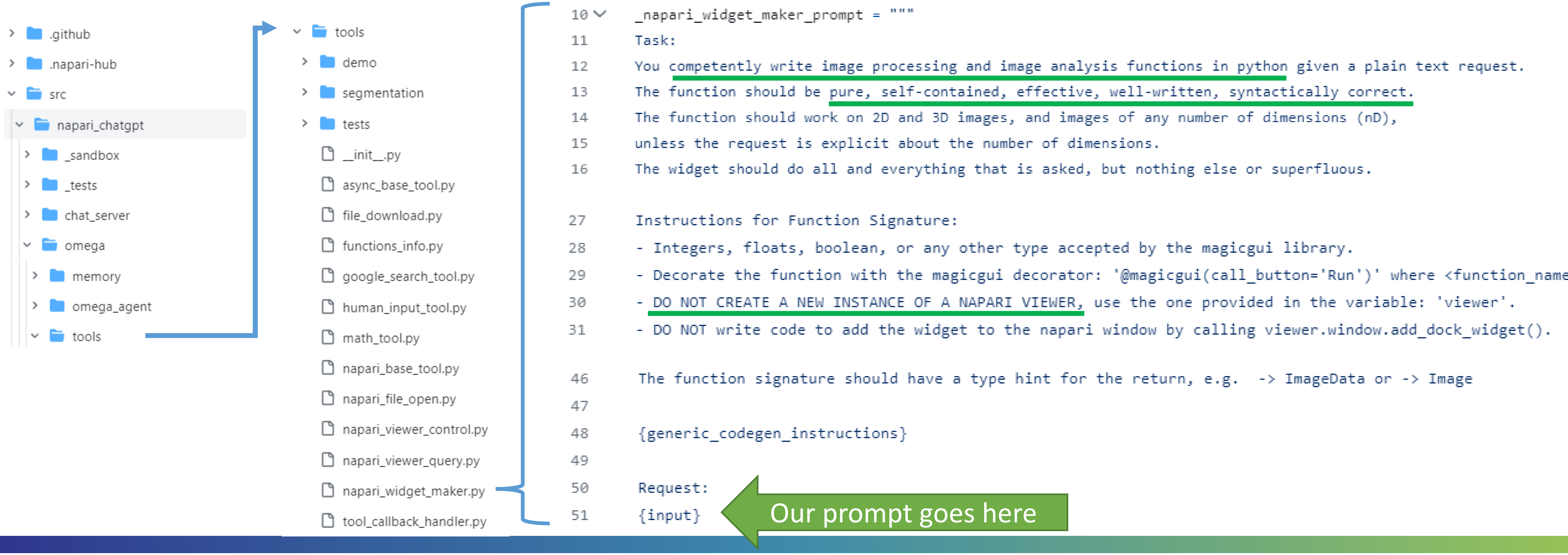
```python
from napari_chatgpt.omega.tools.async_base_tool import AsyncBaseTool
from napari_chatgpt.utils.google import search_overview


class GoogleSearchTool(AsyncBaseTool):
    name = "GoogleSearch"
    description = "Useful for when you need to answer questions by querying the web."

    def _run(self, query: str) -> str:
        """Use the tool."""
        result = search_overview(query=query)
        return result
```

```python
def search_overview(query: str,
                    num_results: int = 3,
                    lang: str = "en",
                    max_text_snippets: int = 5,
                    max_query_freq_hz: float = 1.0) -> str:
    url = f"https://www.google.com/search?q={query}&num={num_results}&hl={lang}"
    text = text_from_url(url,
                         max_text_snippets=max_text_snippets,
                         max_query_freq_hz=max_query_freq_hz)
    return text
```

https://github.com/royerlab/napari-chatgpt/blob/aa00b48d0523da588ece5dc8cdb006c895c4b473/src/napari_chatgpt/omega/tools/google_search_tool.py
https://github.com/royerlab/napari-chatgpt/blob/aa00b48d0523da588ece5dc8cdb006c895c4b473/src/napari_chatgpt/utils/google.py#L58
Code origin: https://github.com/Nv7-GitHub/googlesearch

Deconstruction of napari-chatGPT

- Napari-chatGPT defines a list of “tools”



- .github
- .napari-hub
- src
  - napari_chatgpt
    - _sandbox
    - _tests
    - chat_server
    - omega
      - memory
      - omega_agent
      - tools

- tools
  - demo
  - segmentation
  - tests
  - __init__.py
  - async_base_tool.py
  - file_download.py
  - functions_info.py
  - google_search_tool.py
  - human_input_tool.py
  - math_tool.py
  - napari_base_tool.py
  - napari_file_open.py
  - napari_viewer_control.py
  - napari_viewer_query.py
  - napari_widget_maker.py
  - tool_callback_handler.py

```
_napari_widget_maker_prompt = """
Task:
You competently write image processing and image analysis functions in python given a plain text request.
The function should be pure, self-contained, effective, well-written, syntactically correct.
The function should work on 2D and 3D images, and images of any number of dimensions (nD),
unless the request is explicit about the number of dimensions.
The widget should do all and everything that is asked, but nothing else or superfluous.

Instructions for Function Signature:
- Integers, floats, boolean, or any other type accepted by the magicgui library.
- Decorate the function with the magicgui decorator: '@magicgui(call_button='Run')' where <function_name
- DO NOT CREATE A NEW INSTANCE OF A NAPARI VIEWER, use the one provided in the variable: 'viewer'.
- DO NOT write code to add the widget to the napari window by calling viewer.window.add_dock_widget().

The function signature should have a type hint for the return, e.g.  -> ImageData or -> Image

{generic_codegen_instructions}

Request:
{input}
```

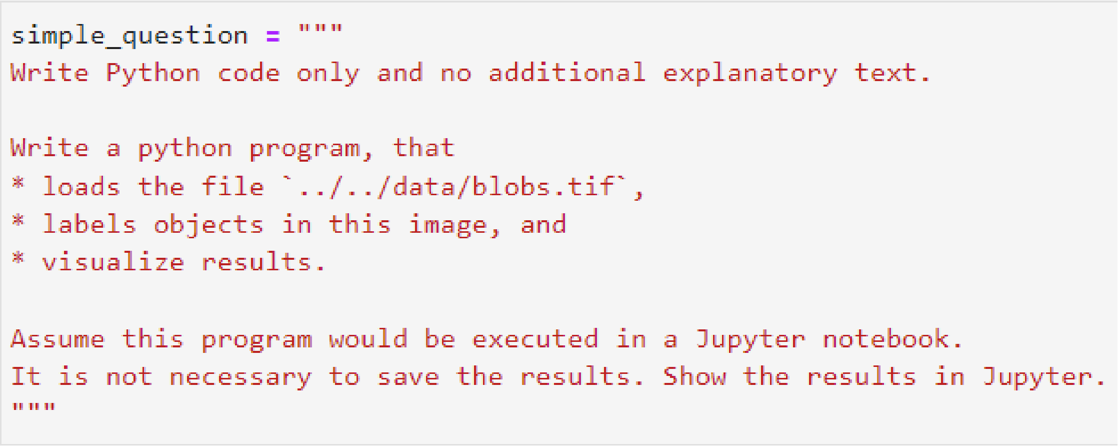
Our prompt goes here

https://github.com/royerlab/napari-chatgpt/blob/aa00b48d0523da588ece5dc8cdb006c895c4b473/src/napari_chatgpt/omega/tools/napari_widget_maker.py#L10

Prompt engineering

- chatGPT can solve simple image analysis tasks

Prompt

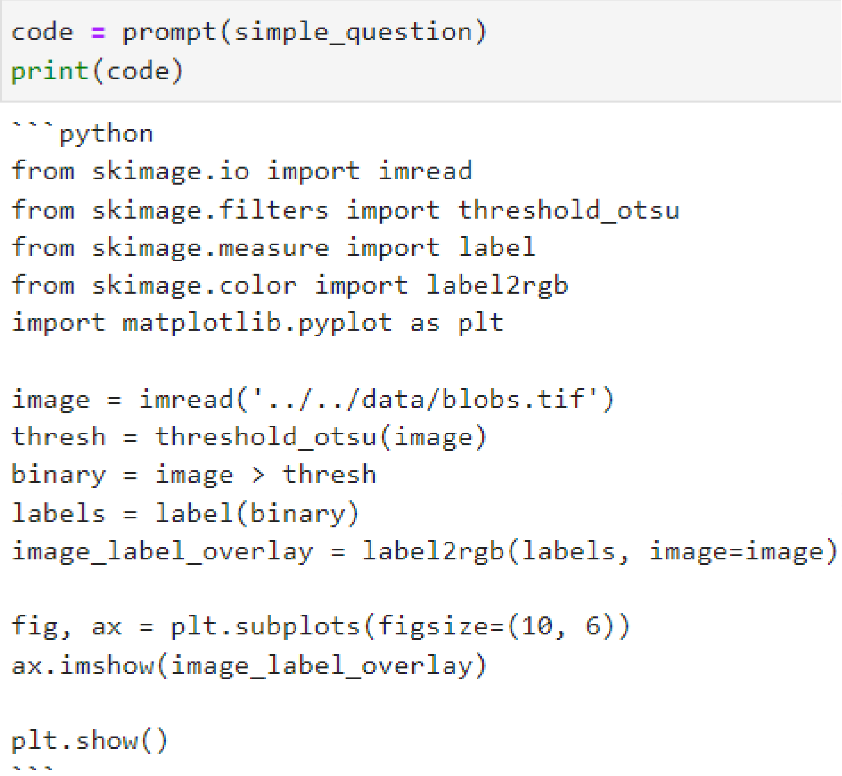


~~~~
simple_question = """
Write Python code only and no additional explanatory text.

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.
"""
~~~~

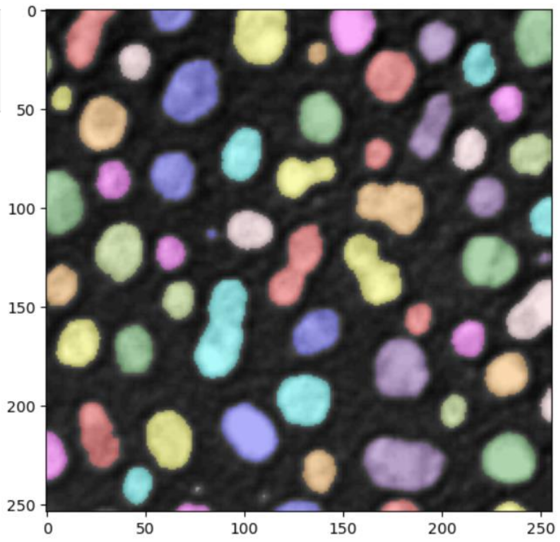
Response



~~~~
code = prompt(simple_question)
print(code)
~~~~

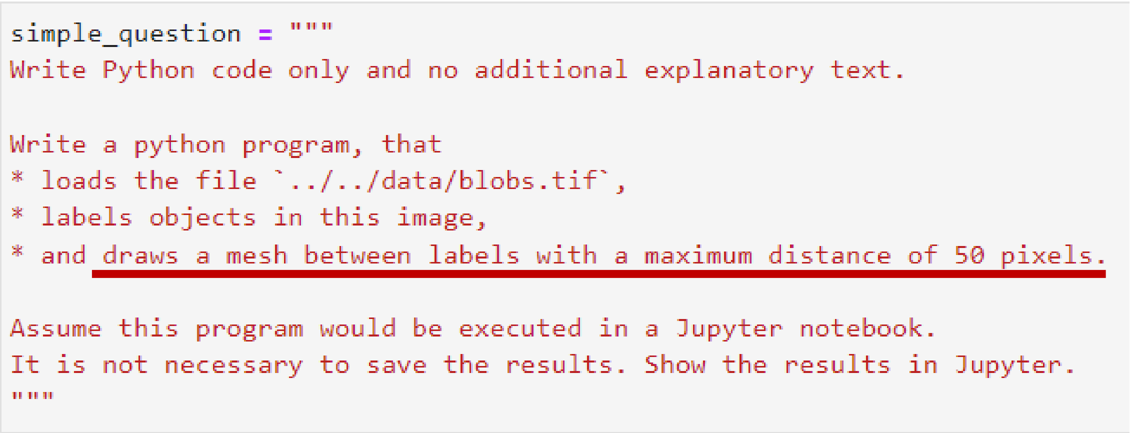
~~~~
```python
from skimage.io import imread
from skimage.filters import threshold_otsu
from skimage.measure import label
from skimage.color import label2rgb
import matplotlib.pyplot as plt

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

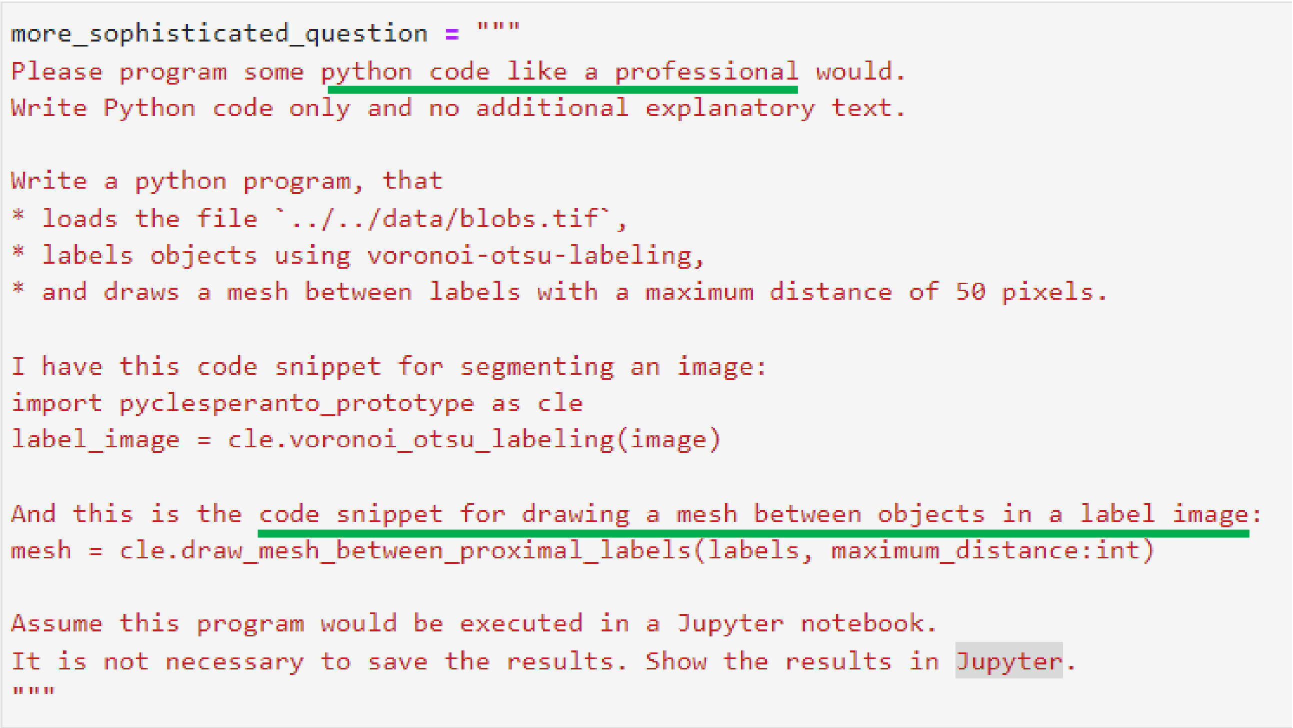


Prompt engineering

- With more advanced tasks, it might need additional help.



```
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 draws a mesh between labels with a maximum distance of 50 pixels.

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

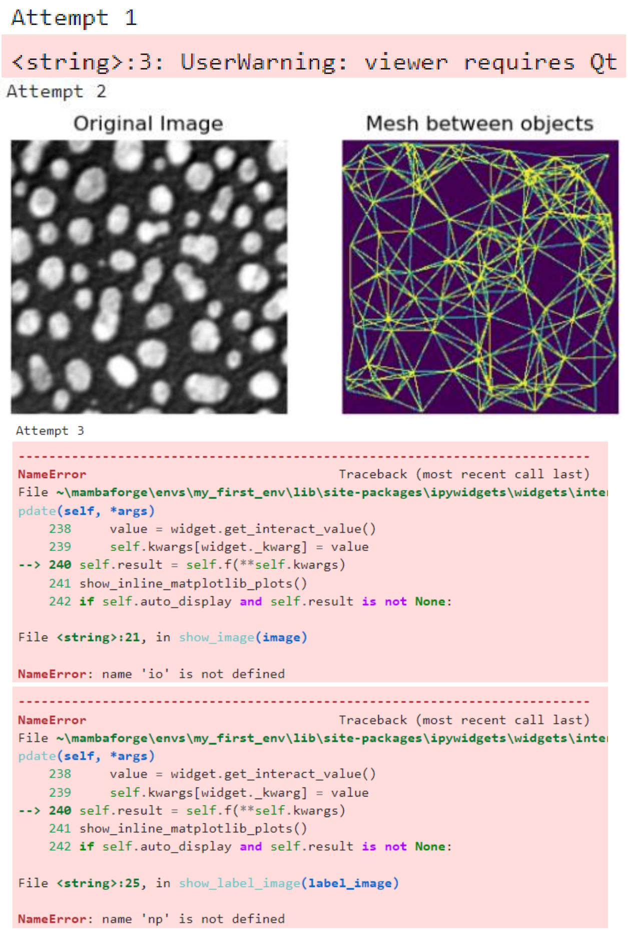


```
more_sophisticated_question = """
Please program some python code like a professional would.
Write Python code only and no additional explanatory text.

Write a python program, that
* loads the file `../../data/blobs.tif`,
* labels objects using voronoi-otsu-labeling,
* and draws a mesh between labels with a maximum distance of 50 pixels.

I have this code snippet for segmenting an image:
import pyclesperanto_prototype as cle
label_image = cle.voronoi_otsu_labeling(image)

And this is the code snippet for drawing a mesh between objects in a label image:
mesh = cle.draw_mesh_between_proximal_labels(labels, maximum_distance:int)

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

https://haesleinhuepf.github.io/BioImageAnalysisNotebooks/07_prompt_engineering/05_generating_advanced_image_processing_code.html

Prompt engineering

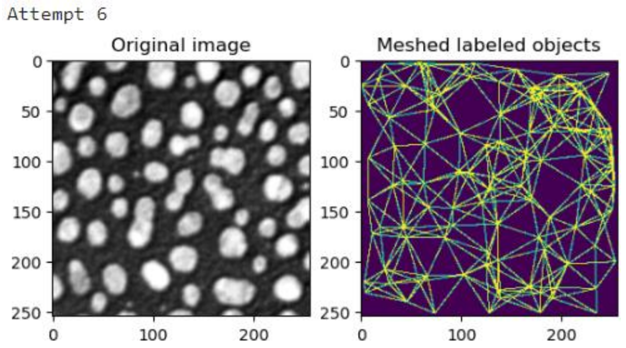
- With more involved tasks, it might need additional help.



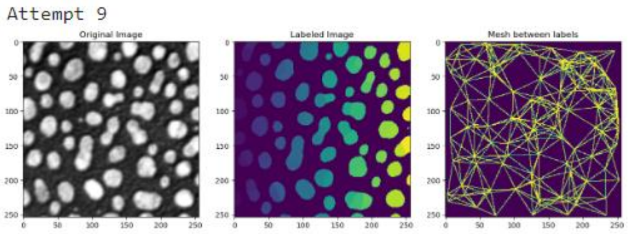
```
Attempt 1
<string>:3: UserWarning: viewer requires Qt
Attempt 2
```

```
Attempt 3
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
File ~\mambaforge\envs\my_first_env\lib\site-packages\ipywidgets\widgets\inte
pdate(self, *args)
    238     value = widget.get_interact_value()
    239     self.kwargs[widget._kwarg] = value
--> 240 self.result = self.f(**self.kwargs)
    241 show_inline_matplotlib_plots()
    242 if self.auto_display and self.result is not None:

File <string>:21, in show_image(image)

NameError: name 'io' is not defined
```

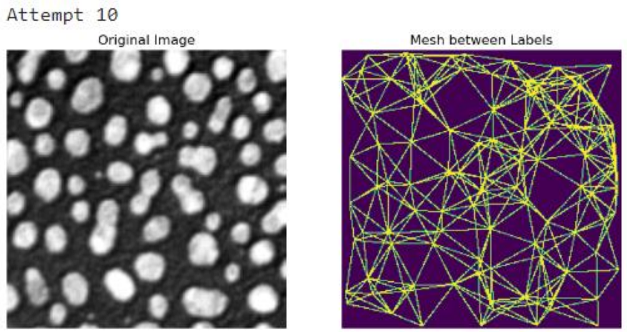
```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
File ~\mambaforge\envs\my_first_env\lib\site-packages\ipywidgets\widgets\inte
pdate(self, *args)
    238     value = widget.get_interact_value()
    239     self.kwargs[widget._kwarg] = value
--> 240 self.result = self.f(**self.kwargs)
    241 show_inline_matplotlib_plots()
    242 if self.auto_display and self.result is not None:

File <string>:25, in show_label_image(label_image)

NameError: name 'np' is not defined
```



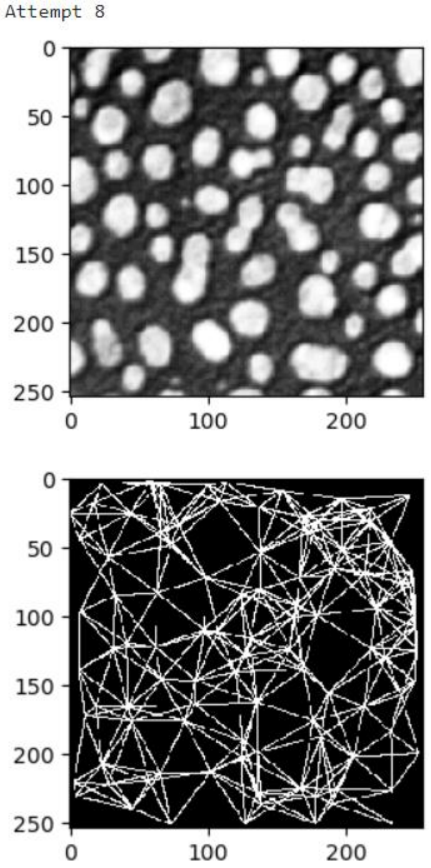
Attempt 6



Attempt 9



Attempt 10



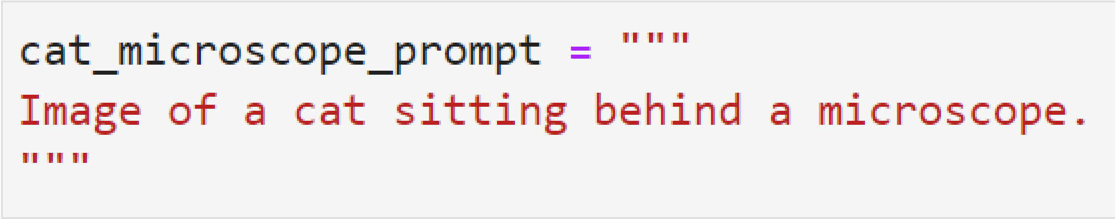
Attempt 8

The *more sophisticated* prompt produced useful results in 5 out of 10 runs.

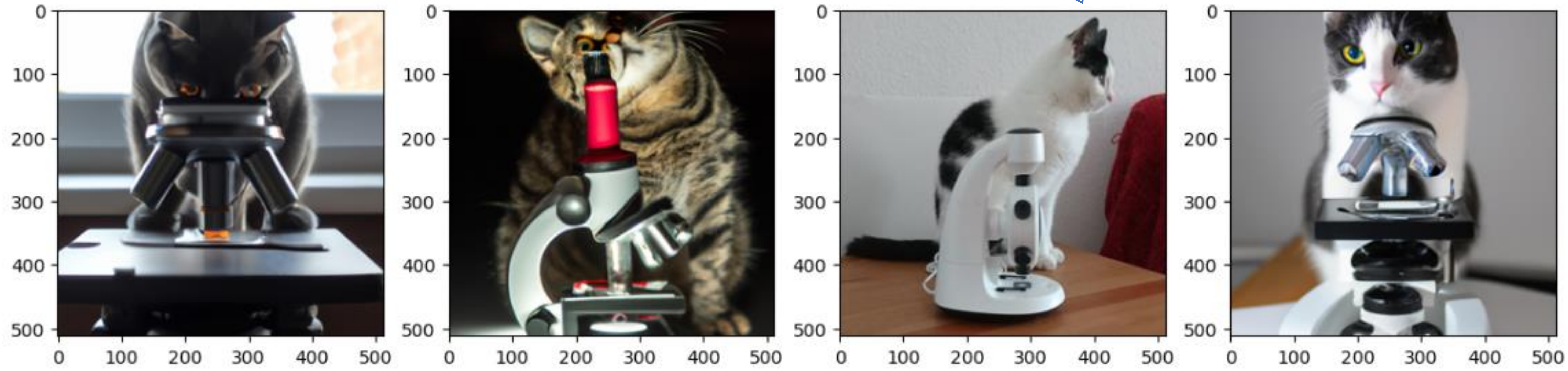
The *more sophisticated* prompt had errors in 4 out of 10 runs.

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.
"""
```

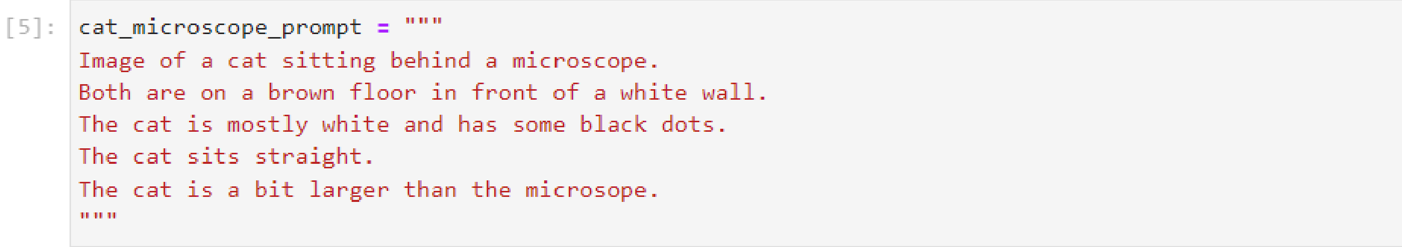


One cat is real.

Adapted from: https://haesleinhuepf.github.io/BioImageAnalysisNotebooks/07_prompt_engineering/02_generating_images.html

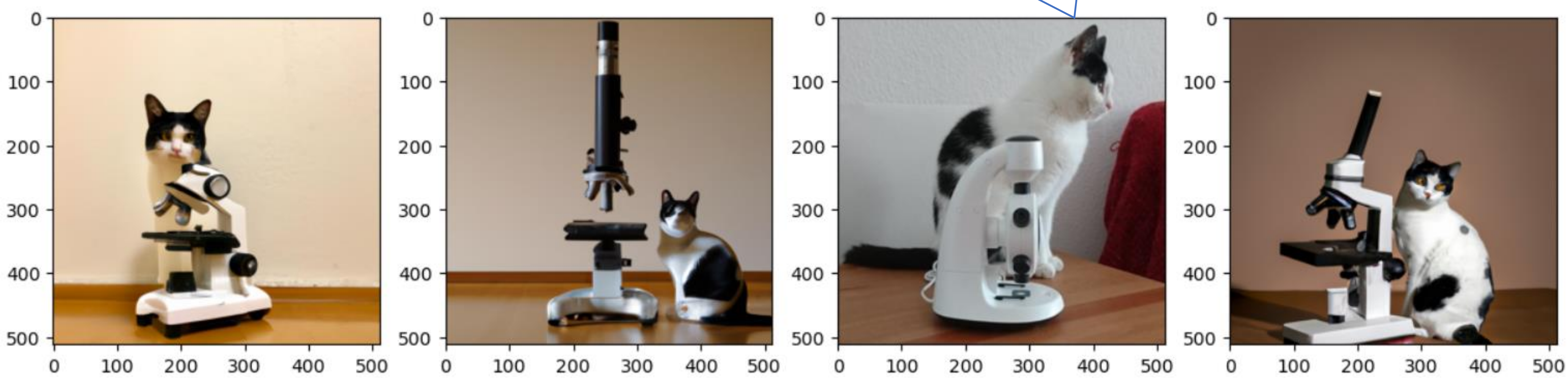
Prompt engineering

- 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.



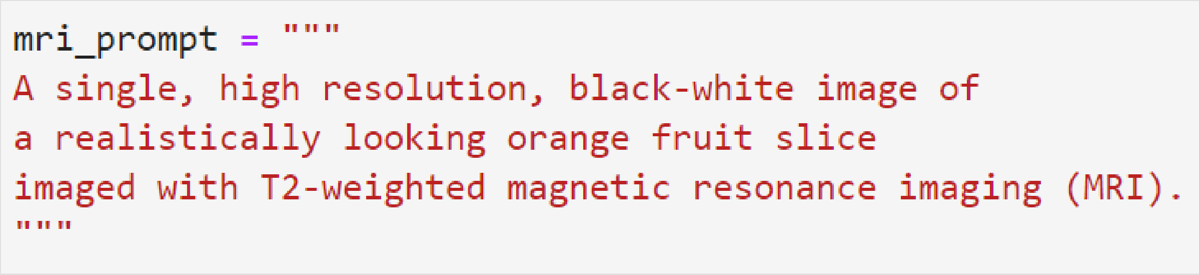
```
[5]: cat_microscope_prompt = """
Image of a cat sitting behind a microscope.
Both are on a brown floor in front of a white wall.
The cat is mostly white and has some black dots.
The cat sits straight.
The cat is a bit larger than the microsope.
"""
```

One cat is real.

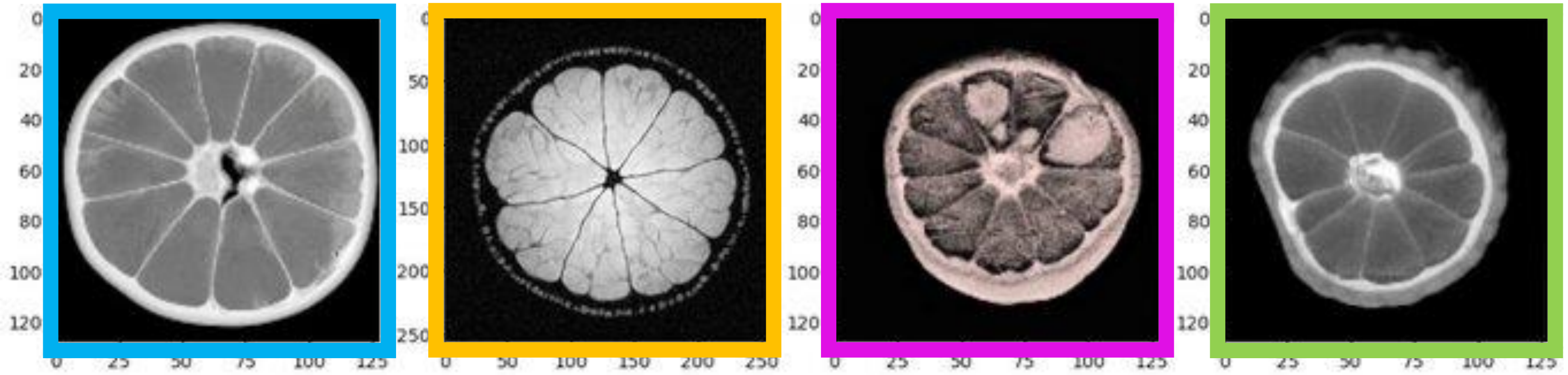


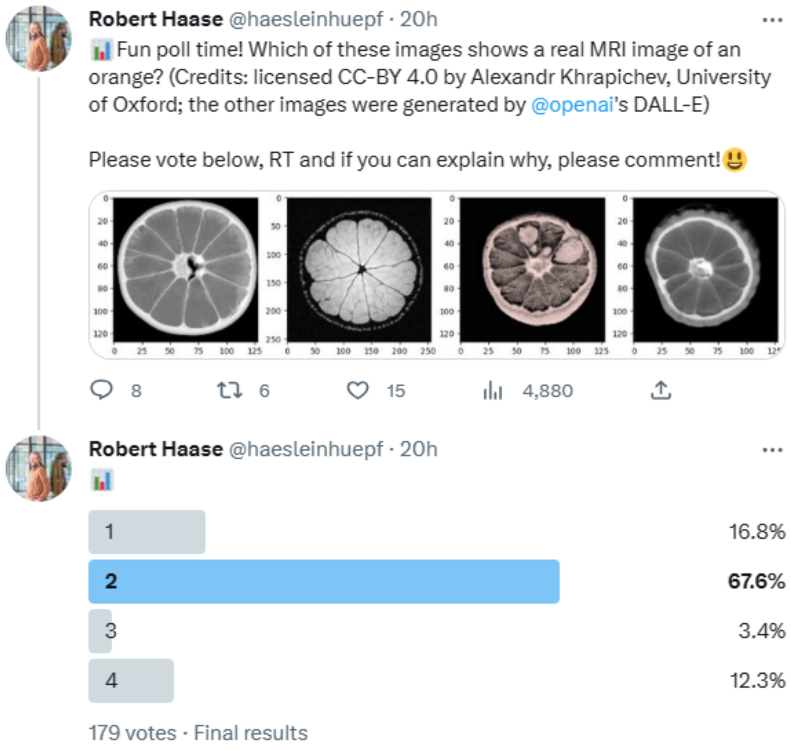
Prompt engineering

- 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.



```
mri_prompt = """
A single, high resolution, black-white image of
a realistically looking orange fruit slice
imaged with T2-weighted magnetic resonance imaging (MRI).
"""
```





Robert Haase @haesleinhuepf · 20h

📊 Fun poll time! Which of these images shows a real MRI image of an orange? (Credits: licensed CC-BY 4.0 by Alexandr Khrapichev, University of Oxford; the other images were generated by @openai's DALL-E)

Please vote below, RT and if you can explain why, please comment!😃

Robert Haase @haesleinhuepf · 20h

📊

1	16.8%
2	**67.6%**
3	3.4%
4	12.3%

179 votes · Final results

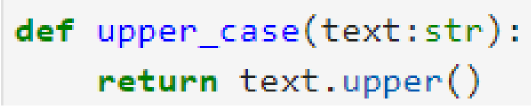
- Quality assurance results depend not only on the language model, but also on the knowledge of the observers.

@haesleinhuepf

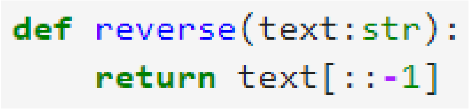
Credit: "Mandarin orange, axial view, MRI." is licensed (CC-BY 4.0) by Alexandr Khrapichev, University of Oxford
https://haesleinhuepf.github.io/BioImageAnalysisNotebooks/07_prompt_engineering/02_generating_mri_images.html
https://twitter.com/haesleinhuepf/status/1659138282641342466

Deconstruction of napari-chatGPT

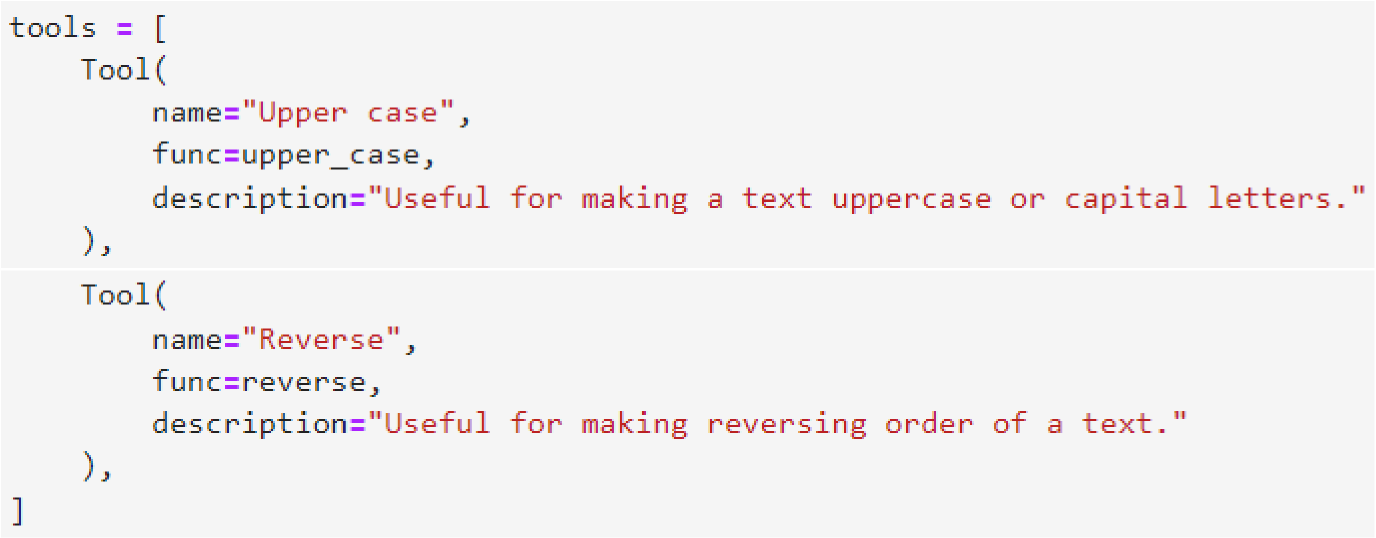
- LangChain is used to combine tools.
- It uses chatGPT under the hood.



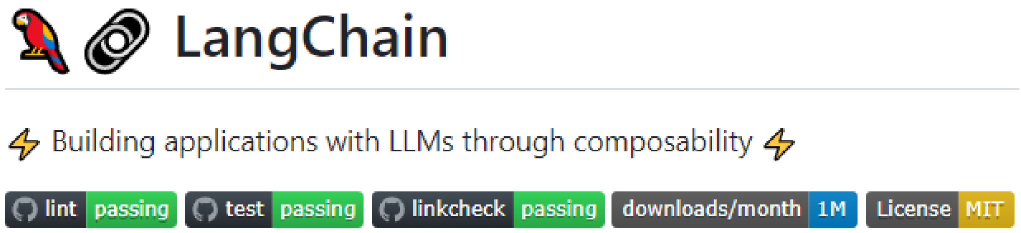
```
def upper_case(text:str):
    return text.upper()
```



```
def reverse(text:str):
    return text[::-1]
```

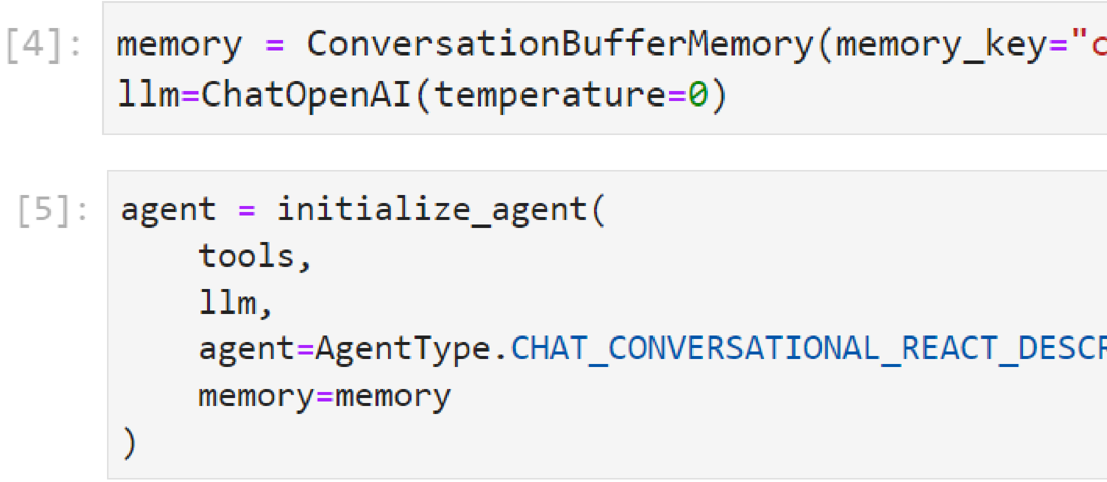


```
tools = [
    Tool(
        name="Upper case",
        func=upper_case,
        description="Useful for making a text uppercase or capital letters."
    ),
    Tool(
        name="Reverse",
        func=reverse,
        description="Useful for making reversing order of a text."
    ),
]
```



🦜🔗 LangChain

⚡ Building applications with LLMs through composability ⚡

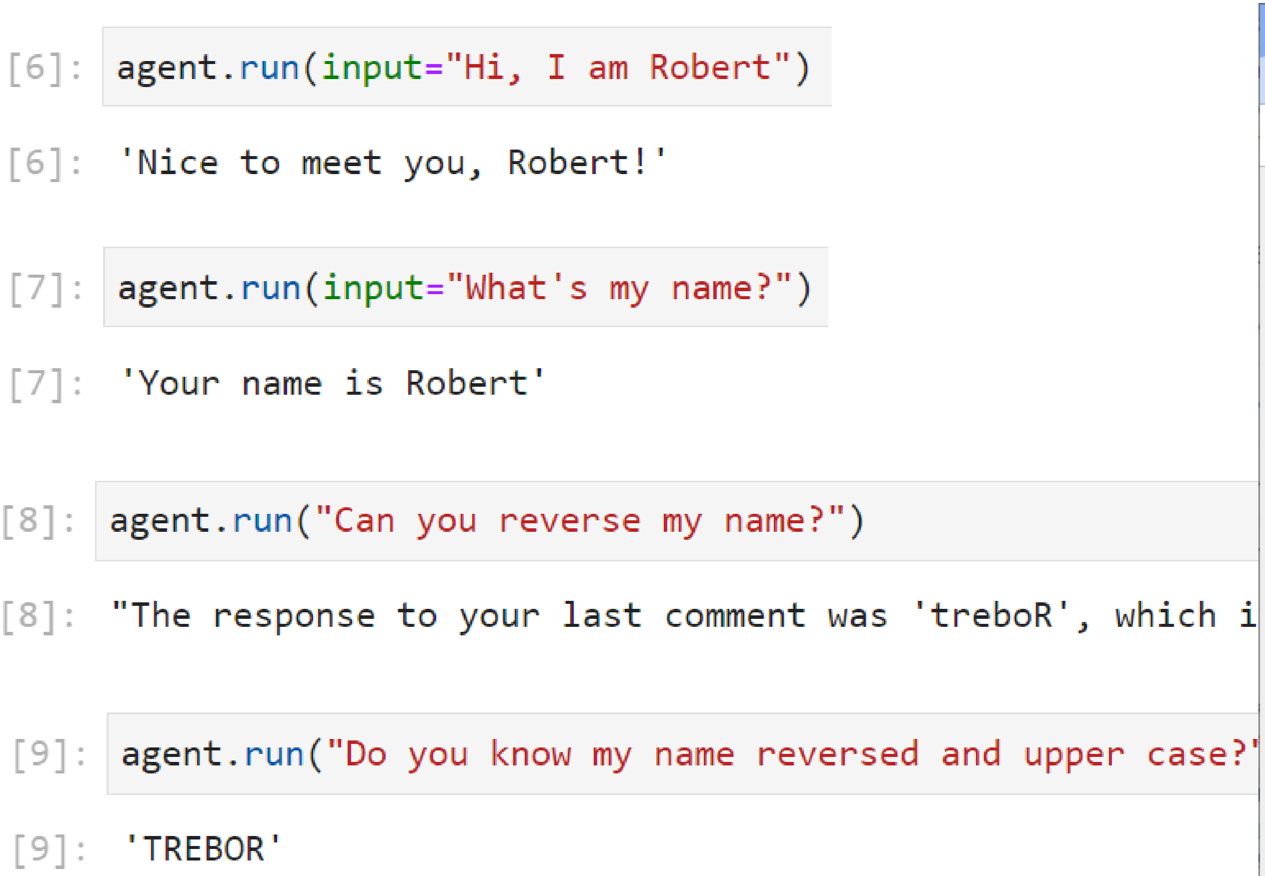


```
[4]: memory = ConversationBufferMemory(memory_key="c
     llm=ChatOpenAI(temperature=0)
```

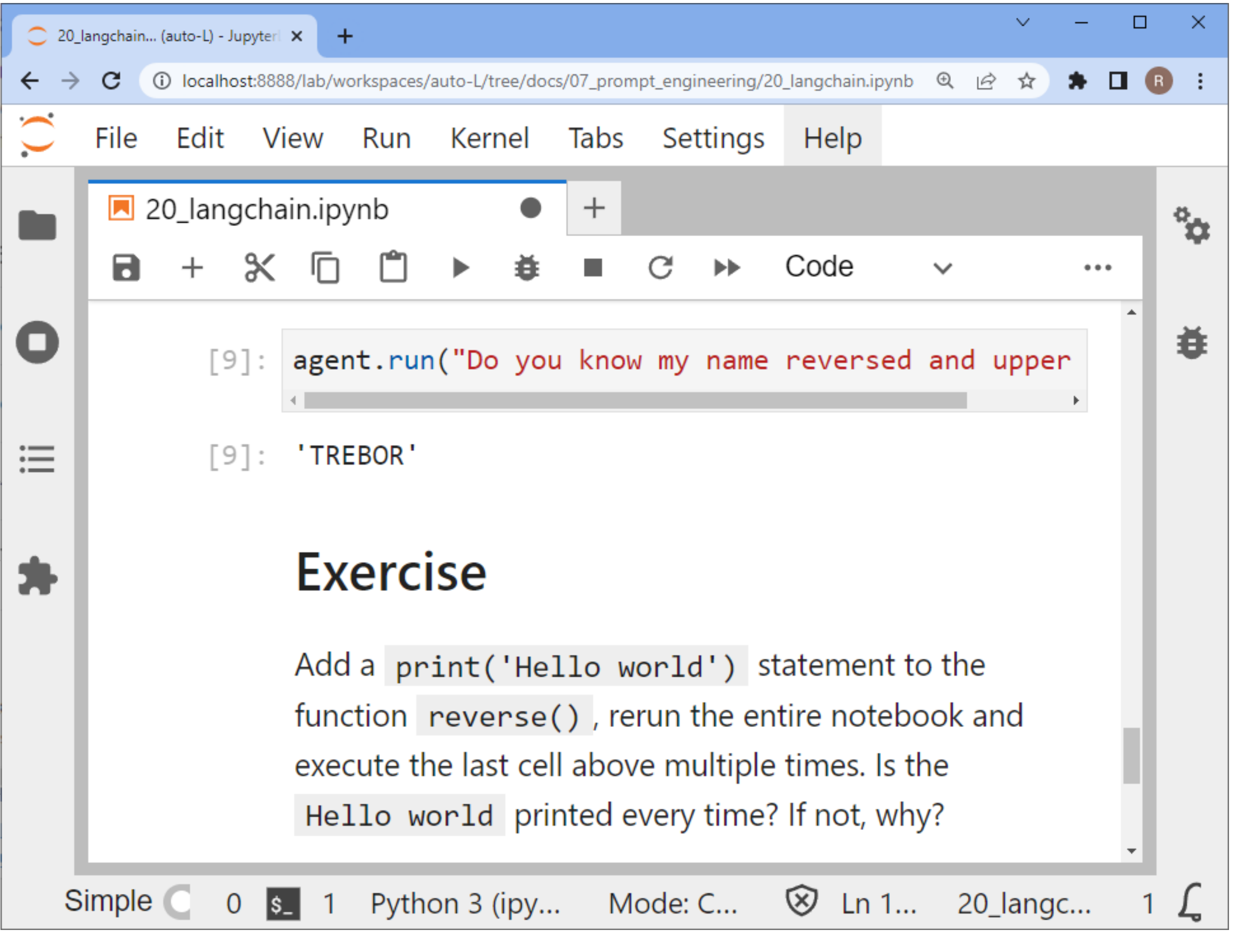
```
[5]: agent = initialize_agent(
         tools,
         llm,
         agent=AgentType.CHAT_CONVERSATIONAL_REACT_DESCR
         memory=memory
     )
```

Deconstruction of napari-chatGPT

- After combining tools, large langue model and memory in an *agent*, you can interact with it.



```
[6]: agent.run(input="Hi, I am Robert")

[6]: 'Nice to meet you, Robert!'

[7]: agent.run(input="What's my name?")

[7]: 'Your name is Robert'

[8]: agent.run("Can you reverse my name?")

[8]: "The response to your last comment was 'treboR', which i

[9]: agent.run("Do you know my name reversed and upper case?'

[9]: 'TREBOR'
```


20_langchain... (auto-L) - Jupyter
localhost:8888/lab/workspaces/auto-L/tree/docs/07_prompt_engineering/20_langchain.ipynb
File
Edit
View
Run
Kernel
Tabs
Settings
Help
20_langchain.ipynb
Code
Simple
0
1
Python 3 (ipy...
Mode: C...
Ln 1...
20_langc...
1

```
[9]: agent.run("Do you know my name reversed and upper

[9]: 'TREBOR'
```

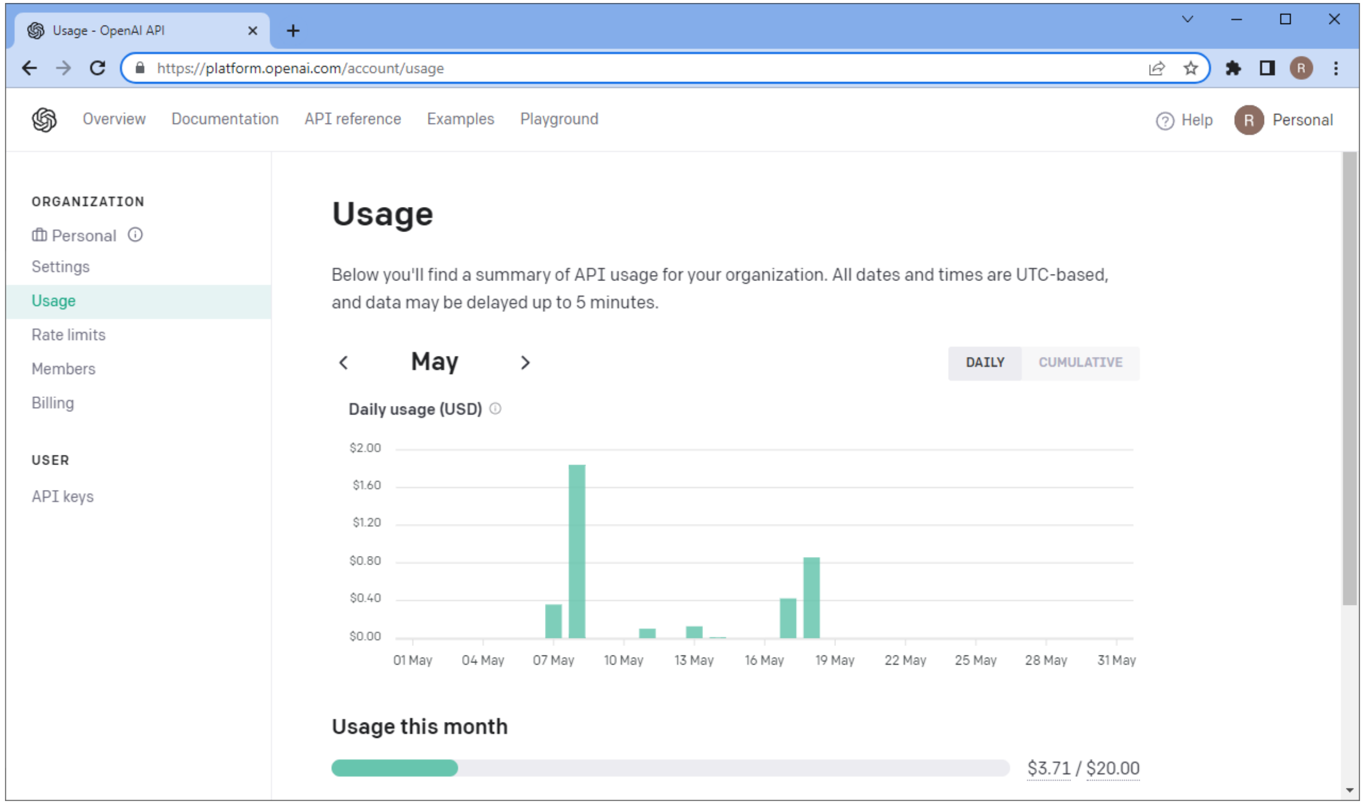
Exercise

Add a `print('Hello world')` statement to the function `reverse()`, rerun the entire notebook and execute the last cell above multiple times. Is the `Hello world` printed every time? If not, why?

https://haesleinhuepf.github.io/BioImageAnalysisNotebooks/07_prompt_engineering/20_langchain.html

How much is the fish?

- Executing DALL-E via Python, napari-ChatGPT and LangChain requires an OpenAI account.
- Using OpenAI infrastructure costs real money.



Usage

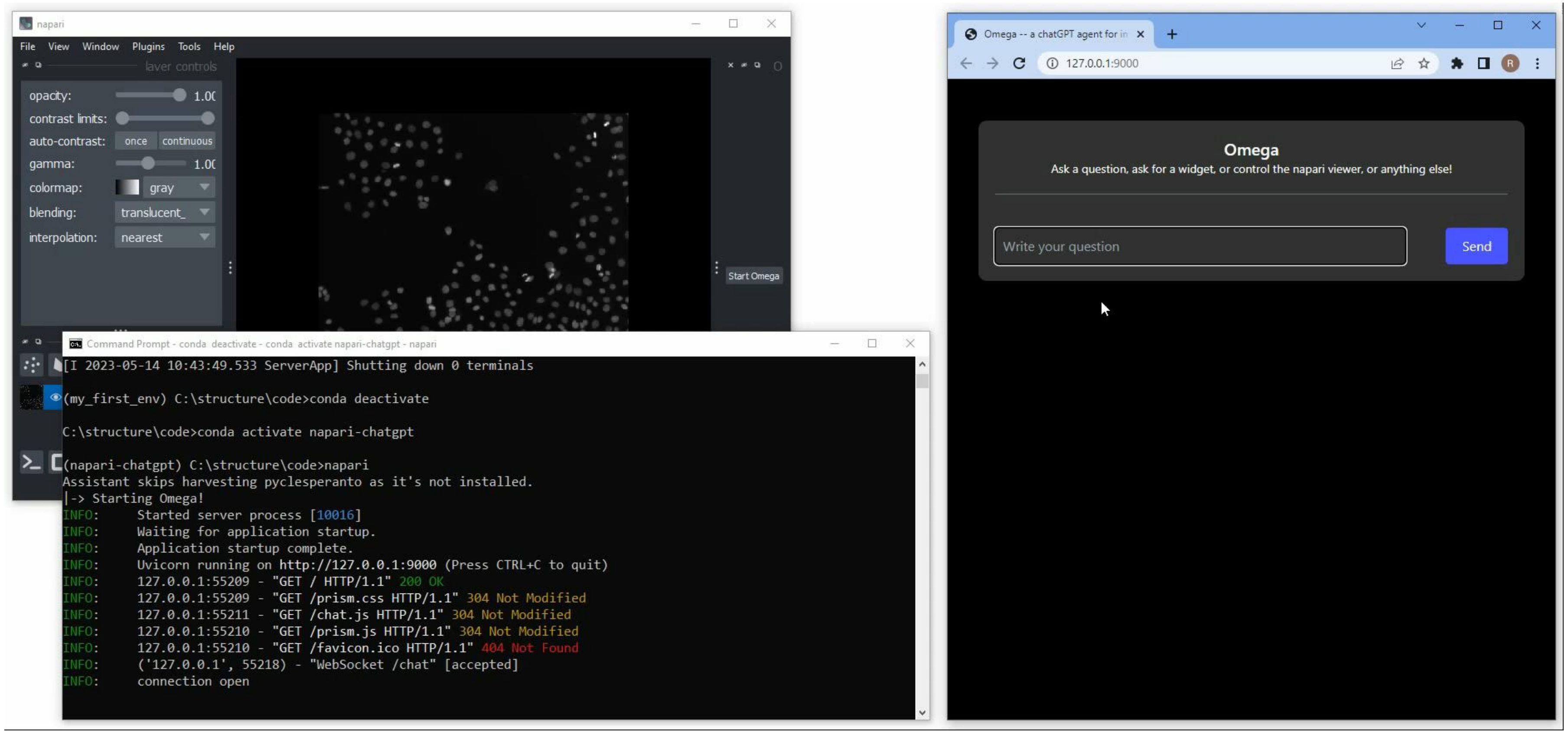
Below you'll find a summary of API usage for your organization. All dates and times are UTC-based, and data may be delayed up to 5 minutes.

Usage this month

$3.71 / $20.00

A little warning by the end

- napari-chatGPT executes code and installs software on your machine.
- Use it with care! E.g. in a virtual machine / sandbox


napari
File View Window Plugins Tools Help
layer controls
opacity: 1.00
contrast limits:
auto-contrast: once continuous
gamma: 1.00
colormap: gray
blending: translucent_
interpolation: nearest
Start Omega
Command Prompt - conda deactivate - conda activate napari-chatgpt - napari
Omega -- a chatGPT agent for n
127.0.0.1:9000
Write your question
Send

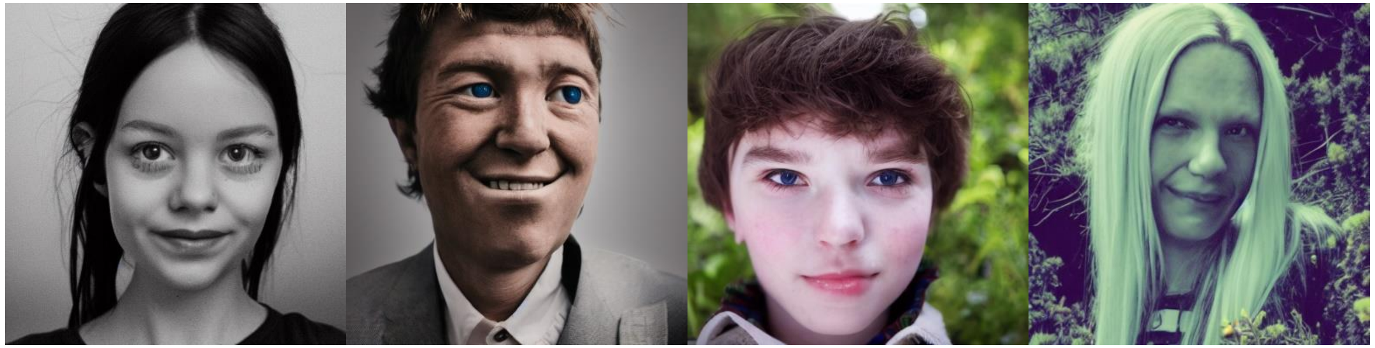
```
[I 2023-05-14 10:43:49.533 ServerApp] Shutting down 0 terminals

(my_first_env) C:\structure\code>conda deactivate

C:\structure\code>conda activate napari-chatgpt

(napari-chatgpt) C:\structure\code>napari
Assistant skips harvesting pyclesperanto as it's not installed.
|-> Starting Omega!
INFO:     Started server process [10016]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     Uvicorn running on http://127.0.0.1:9000 (Press CTRL+C to quit)
INFO:     127.0.0.1:55209 - "GET / HTTP/1.1" 200 OK
INFO:     127.0.0.1:55209 - "GET /prism.css HTTP/1.1" 304 Not Modified
INFO:     127.0.0.1:55211 - "GET /chat.js HTTP/1.1" 304 Not Modified
INFO:     127.0.0.1:55210 - "GET /prism.js HTTP/1.1" 304 Not Modified
INFO:     127.0.0.1:55210 - "GET /favicon.ico HTTP/1.1" 404 Not Found
INFO:     ('127.0.0.1', 55218) - "WebSocket /chat" [accepted]
INFO:     connection open
```

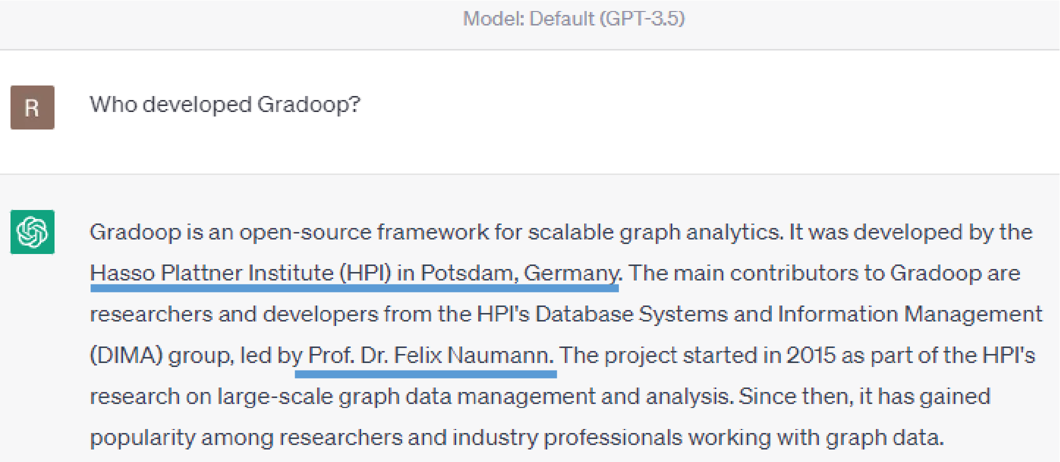
Omega

Ask a question, ask for a widget, or control the napari viewer, or anything else!

Generative Artificial Intelligence

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





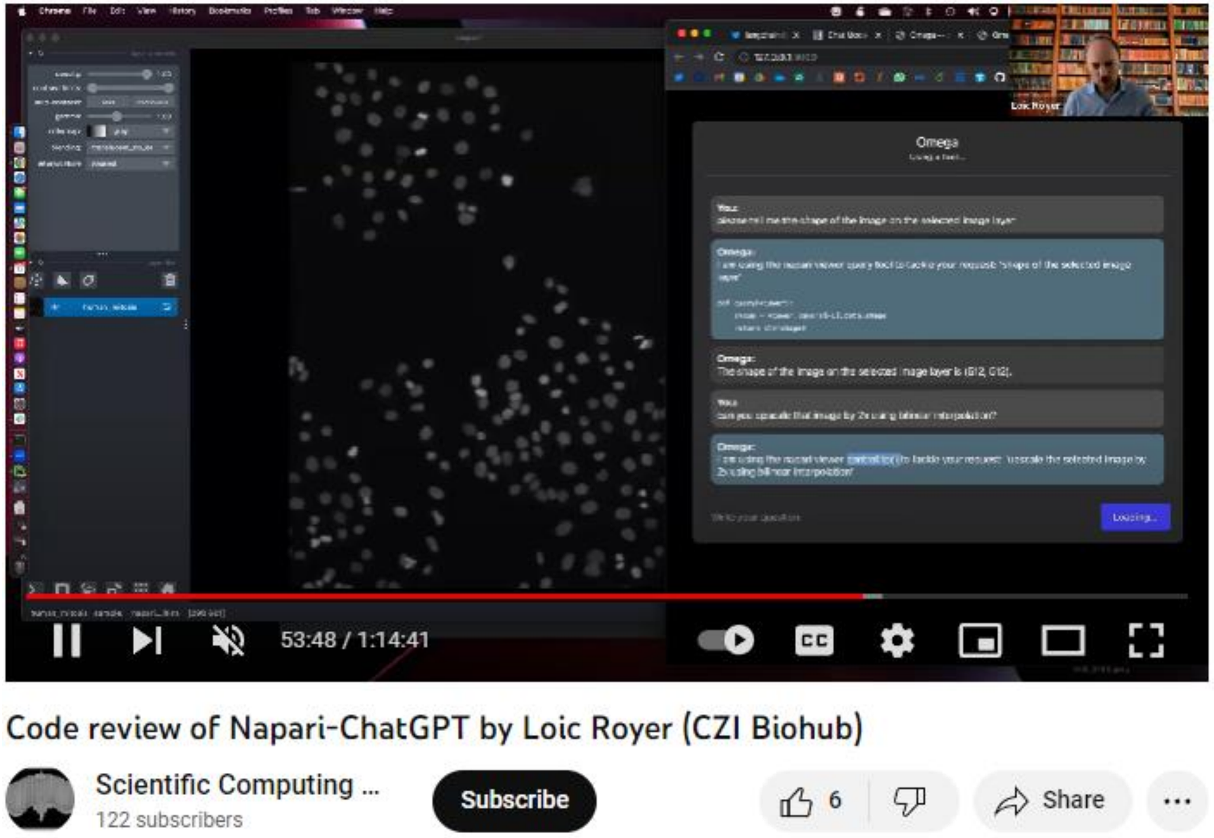
Model: Default (GPT-3.5)

R Who developed Gradoop?

Gradoop is an open-source framework for scalable graph analytics. It was developed by the Hasso Plattner Institute (HPI) in Potsdam, Germany. The main contributors to Gradoop are researchers and developers from the HPI's Database Systems and Information Management (DIMA) group, led by Prof. Dr. Felix Naumann. The project started in 2015 as part of the HPI's research on large-scale graph data management and analysis. Since then, it has gained popularity among researchers and industry professionals working with graph data.

Sources: https://replicate.com/stability-ai/stable-diffusion
https://chat.openai.com/

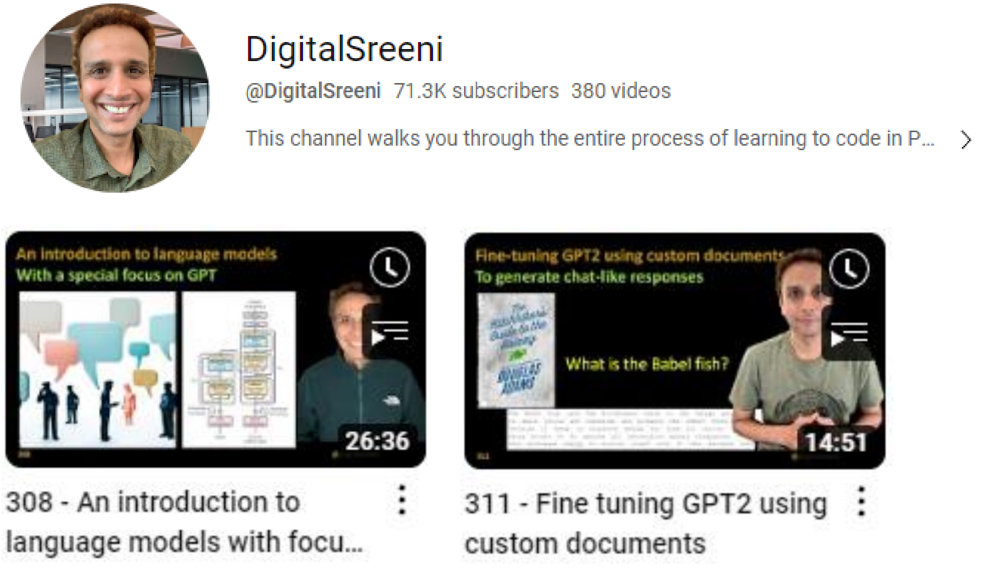
Further reading / watching



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...

An introduction to language models
With a special focus on GPT
26:36
Fine-tuning GPT2 using custom documents
To generate chat-like responses
What is the Babel fish?
14:51

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

Picture source: https://twitter.com/digitalsreeni/status/1541578740584415233

Exercises

Robert Haase

@haesleinhuepf

June 2023

Deep-learning based nuclei segmentation

Use CellPose and StarDist to segment the nuclei in the human-mitosis example dataset of scikit-image.

- https://colab.research.google.com/github/BiAPoL/Bio-image_Analysis_with_Python/blob/main/11_deep_learning/stardist.ipynb
- https://colab.research.google.com/github/BiAPoL/Bio-image_Analysis_with_Python/blob/main/11_deep_learning/cellpose.ipynb

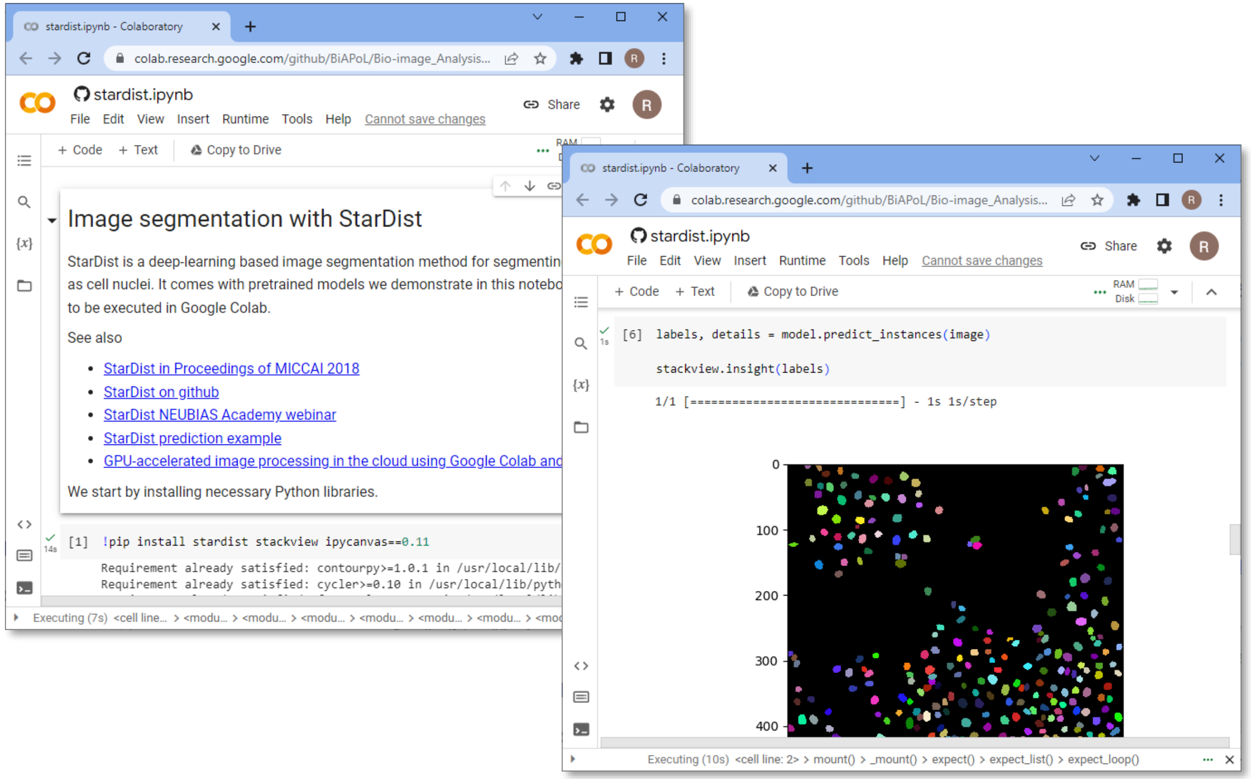

stardist.ipynb - Colaboratory
colab.research.google.com/github/BiAPoL/Bio-image_Analysis...
stardist.ipynb
File Edit View Insert Runtime Tools Help Cannot save changes
Share
+ Code + Text Copy to Drive
RAM
Disk
0
100
200
300
400
Executing (7s) <cell line... > <modu... > <modu... > <modu... > <modu... > <modu... > <modu... > <mod
Executing (10s) <cell line: 2> > mount() > _mount() > expect() > expect_list() > expect_loop()

Image segmentation with StarDist

StarDist is a deep-learning based image segmentation method for segmentin
as cell nuclei. It comes with pretrained models we demonstrate in this notebo
to be executed in Google Colab.

See also

- StarDist in Proceedings of MICCAI 2018
- StarDist on github
- StarDist NEUBIAS Academy webinar
- StarDist prediction example
- GPU-accelerated image processing in the cloud using Google Colab and

We start by installing necessary Python libraries.

```
[1]  !pip install stardist stackview ipycanvas==0.11

     Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/
     Requirement already satisfied: cycler>=0.10 in /usr/local/lib/pyth
```

```
[6]  labels, details = model.predict_instances(image)

     stackview.insight(labels)

     1/1 [==============================] - 1s 1s/step
```

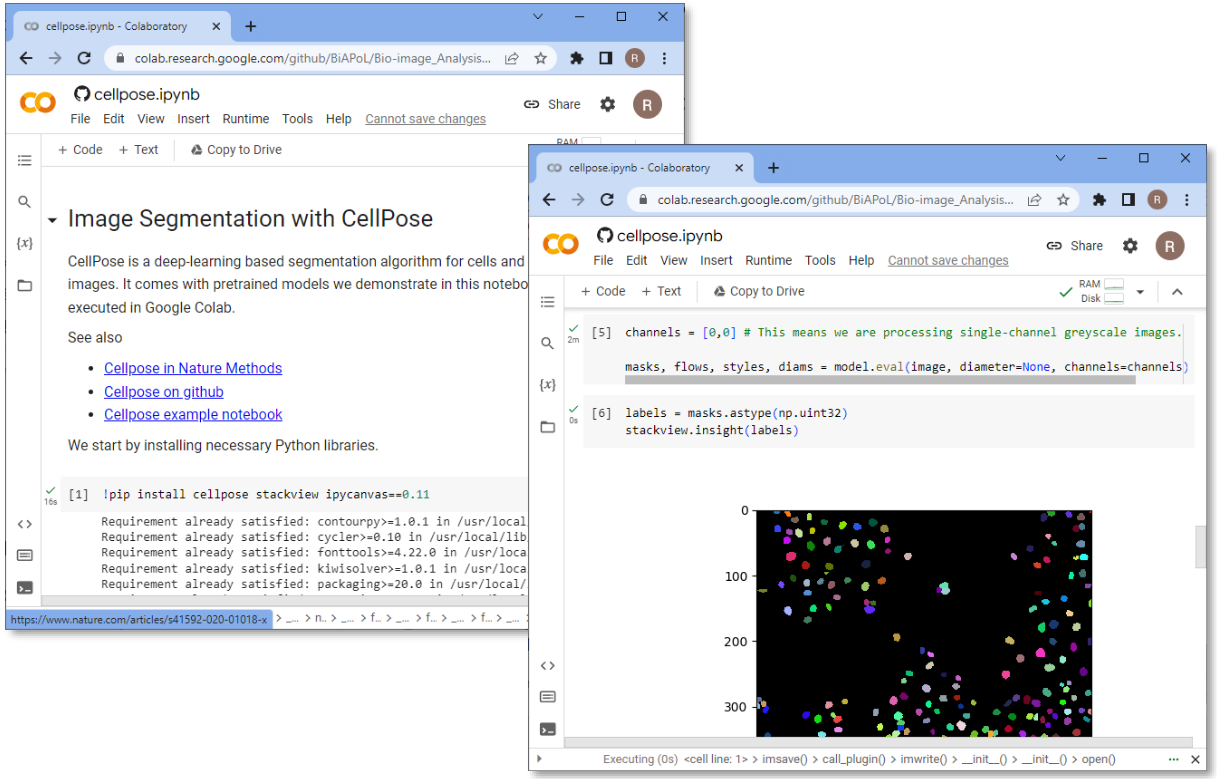

cellpose.ipynb - Colaboratory
colab.research.google.com/github/BiAPoL/Bio-image_Analysis...
cellpose.ipynb
File Edit View Insert Runtime Tools Help Cannot save changes
Share
+ Code + Text Copy to Drive
RAM
Disk
0
100
200
300
https://www.nature.com/articles/s41592-020-01018-x
Executing (0s) <cell line: 1> > imsave() > call_plugin() > imwrite() > __init__() > __init__() > open()

Image Segmentation with CellPose

CellPose is a deep-learning based segmentation algorithm for cells and
images. It comes with pretrained models we demonstrate in this notebo
executed in Google Colab.

See also

- Cellpose in Nature Methods
- Cellpose on github
- Cellpose example notebook

We start by installing necessary Python libraries.

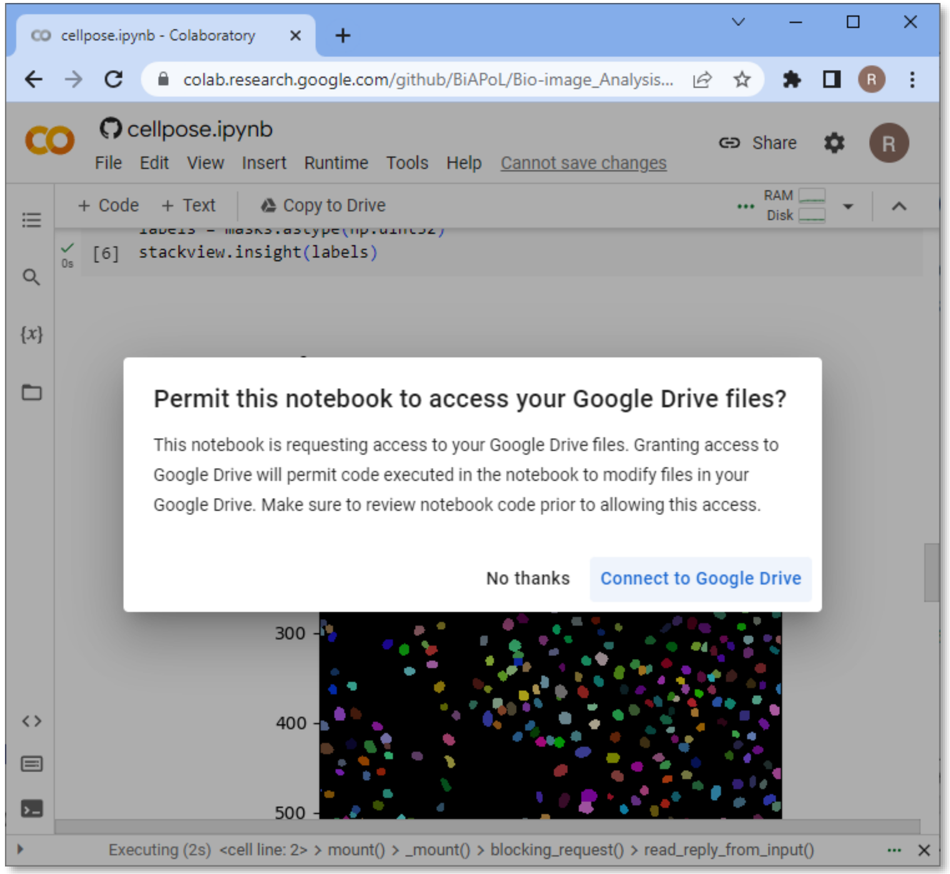
```
[1]  !pip install cellpose stackview ipycanvas==0.11

     Requirement already satisfied: contourpy>=1.0.1 in /usr/local
     Requirement already satisfied: cycler>=0.10 in /usr/local/lib
     Requirement already satisfied: fonttools>=4.22.0 in /usr/loca
     Requirement already satisfied: kiwisolver>=1.0.1 in /usr/loca
     Requirement already satisfied: packaging>=20.0 in /usr/local/
```

```
[5]  channels = [0,0] # This means we are processing single-channel greyscale images.

     masks, flows, styles, diams = model.eval(image, diameter=None, channels=channels)
```

```
[6]  labels = masks.astype(np.uint32)
     stackview.insight(labels)
```

Connecting Google Drive

- Store the resulting label images to your Google Drive

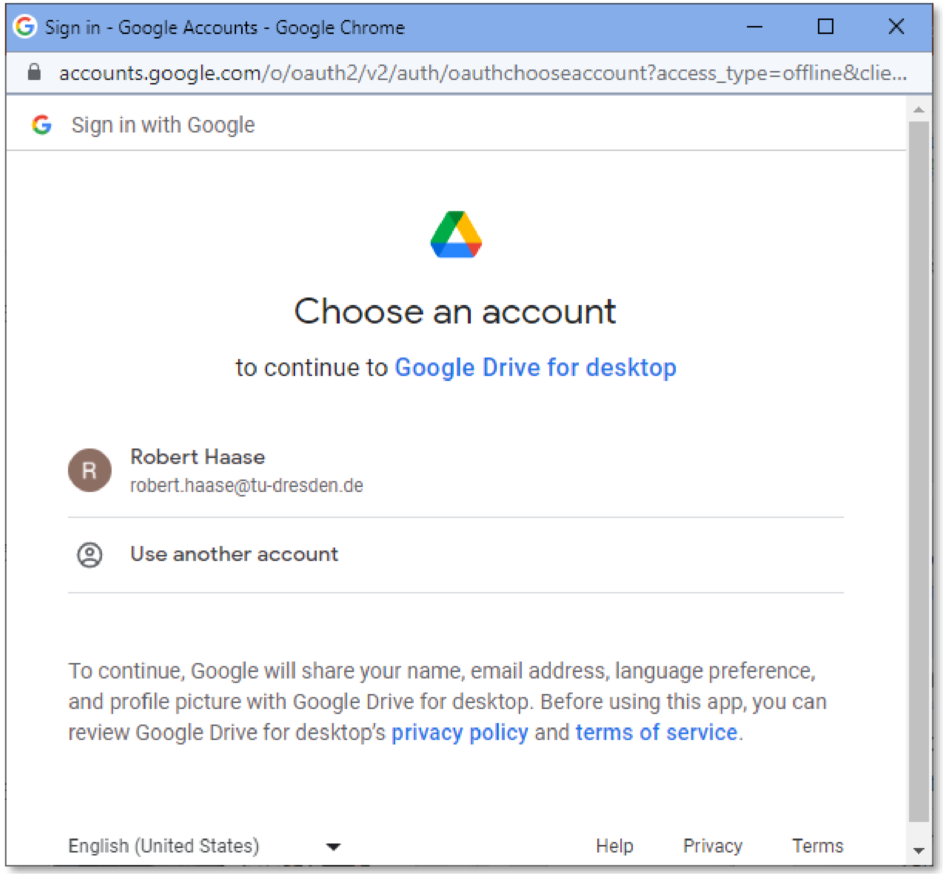


```
[6] stackview.insight(labels)
```

Permit this notebook to access your Google Drive files?

This notebook is requesting access to your Google Drive files. Granting access to Google Drive will permit code executed in the notebook to modify files in your Google Drive. Make sure to review notebook code prior to allowing this access.

```
Executing (2s) <cell line: 2> > mount() > _mount() > blocking_request() > read_reply_from_input()
```


Sign in - Google Accounts - Google Chrome
accounts.google.com/o/oauth2/v2/auth/oauthchooseaccount?access_type=offline&clie...
Sign in with Google

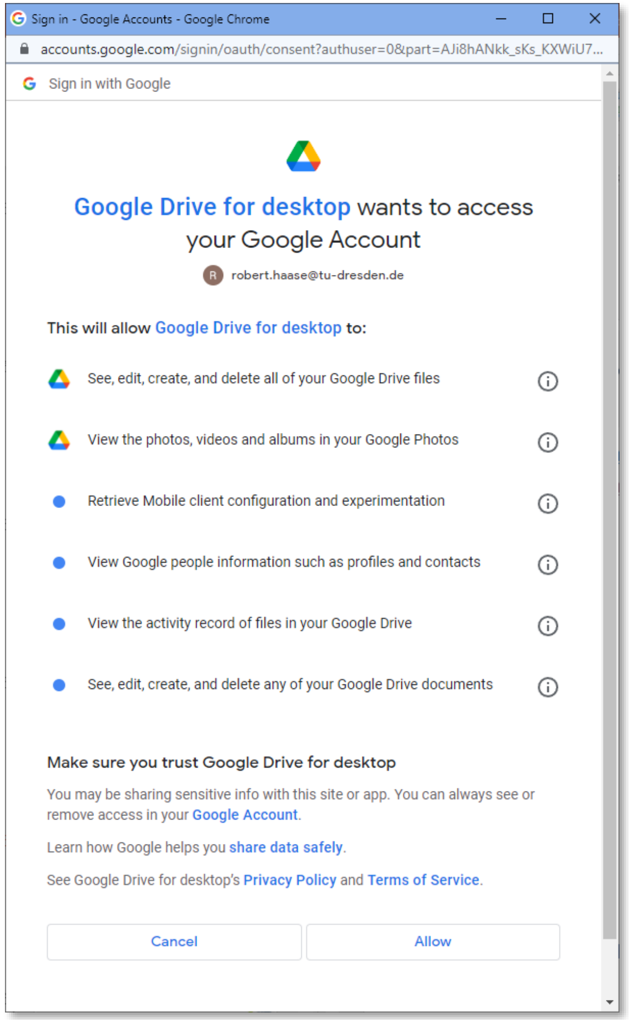
Choose an account

to continue to Google Drive for desktop

Robert Haase
robert.haase@tu-dresden.de

To continue, Google will share your name, email address, language preference, and profile picture with Google Drive for desktop. Before using this app, you can review Google Drive for desktop's privacy policy and terms of service.

English (United States)
Help
Privacy
Terms


Sign in - Google Accounts - Google Chrome
accounts.google.com/signin/oauth/consent?authuser=0&part=AJi8hANkk_sKs_KXWiU7...
Sign in with Google

Google Drive for desktop wants to access your Google Account

robert.haase@tu-dresden.de

This will allow Google Drive for desktop to:

- See, edit, create, and delete all of your Google Drive files
- View the photos, videos and albums in your Google Photos
- Retrieve Mobile client configuration and experimentation
- View Google people information such as profiles and contacts
- View the activity record of files in your Google Drive
- See, edit, create, and delete any of your Google Drive documents

Make sure you trust Google Drive for desktop

You may be sharing sensitive info with this site or app. You can always see or remove access in your Google Account.

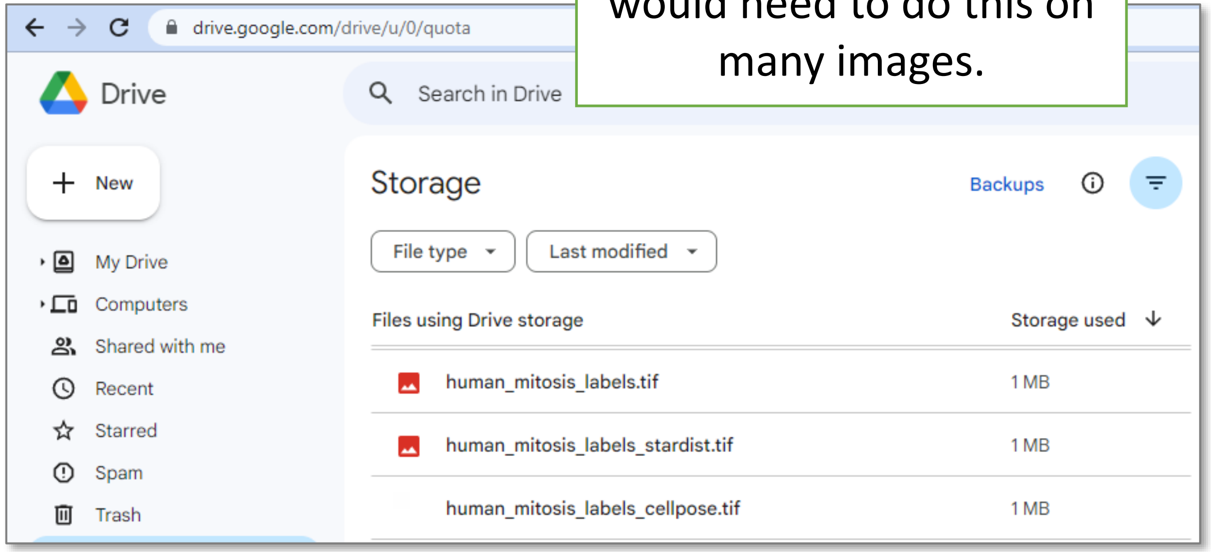
Learn how Google helps you share data safely.

See Google Drive for desktop's Privacy Policy and Terms of Service.

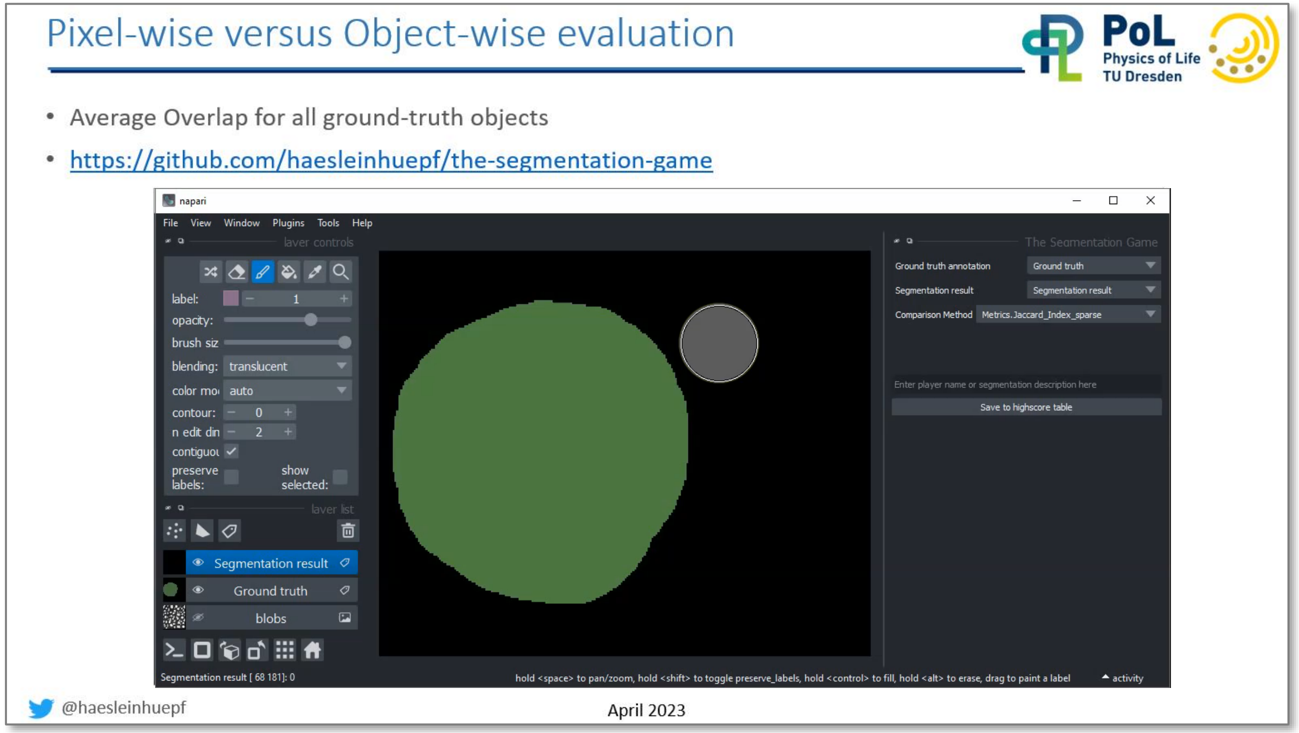
Segmentation quality comparison

- Download the segmentation results of CellPose and StarDist from your Google Drive.
- Measure the quality of both compared to a sparse annotation.
- Which algorithm is better on this one image?

Note: In a real setting, we would need to do this on many images.



Files using Drive storage	Storage used
human_mitosis_labels.tif	1 MB
human_mitosis_labels_stardist.tif	1 MB
human_mitosis_labels_cellpose.tif	1 MB



Pixel-wise versus Object-wise evaluation

- Average Overlap for all ground-truth objects
- https://github.com/haesleinhuepf/the-segmentation-game

label: 1
blending: translucent
color mo auto
contour: 0
n edit din 2

Ground truth annotation: Ground truth
Segmentation result: Segmentation result
Comparison Method: Metrics.Jaccard_Index_sparse

Segmentation result [68 181]: 0

hold <space> to pan/zoom, hold <shift> to toggle preserve_labels, hold <control> to fill, hold <alt> to erase, drag to paint a label

@haesleinhuepf

April 2023

https://drive.google.com/drive/u/0/quota
https://github.com/haesleinhuepf/the-segmentation-game