ScaDS.AI
DRESDEN LEIPZIG
CENTER FOR SCALABLE DATA ANALYTICS
AND ARTIFICIAL INTELLIGENCE

NFDI4
BIOIMAGE
NATIONAL RESEARCH DATA MANAGEMENT INFRASTRUCTURE
FOR MICROSCOPY AND BIOIMAGE ANALYSIS

GloBIAS
Global BioImage Analysts' Society
GLOBAL BIOIMAGE
ANALYST'S SOCIETY

ChatGPT for Image Analysis

Robert Haase

These slides can be reused under the terms of the [CC-BY4.0](https://creativecommons.org/licenses/by/4.0/) license.

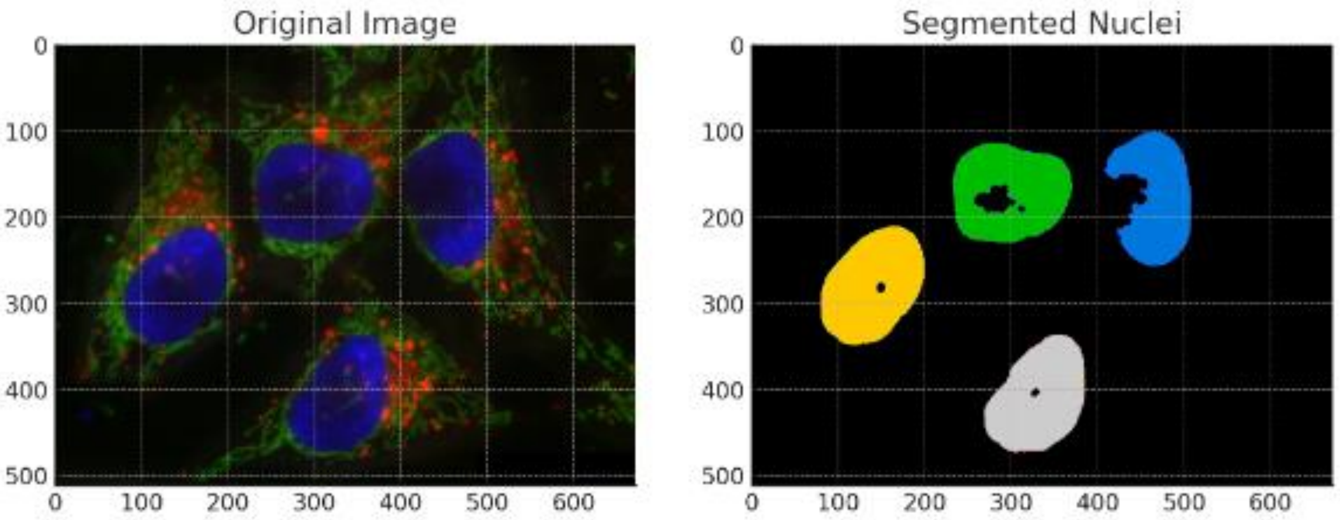
https://doi.org/10.5281/zenodo.13336075

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.

Bio-image Analysis

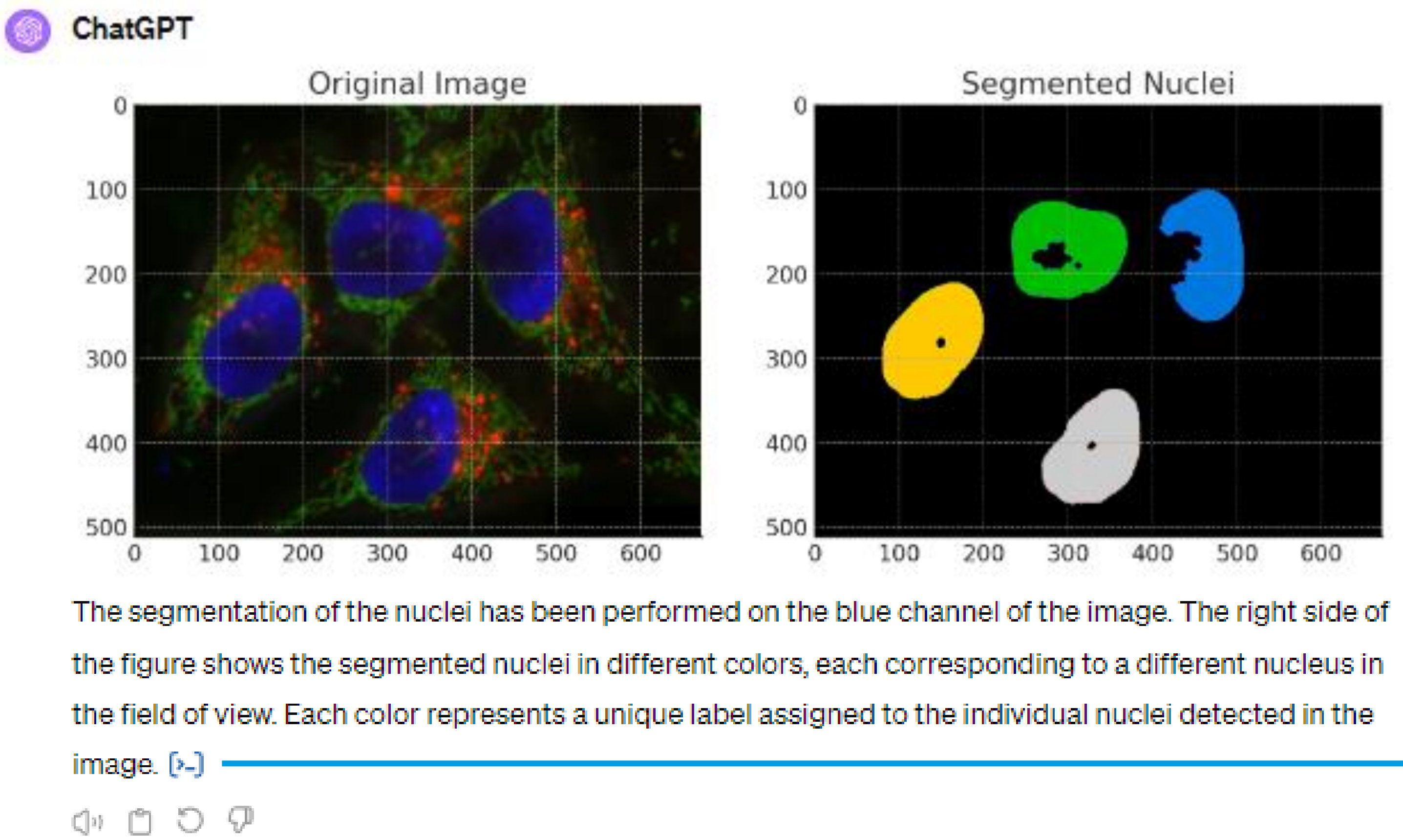
My job …



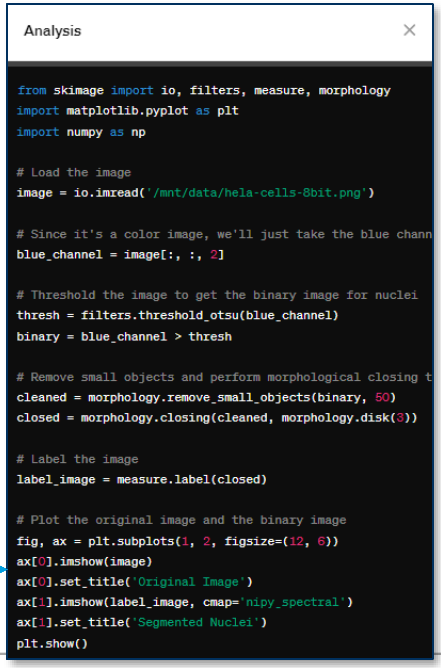
Bio-image Analysis using Large-Language Models

My job is changing, since we have ChatGPT

Prompt:
“Please segment the nuclei in the blue channel of the image”



The segmentation of the nuclei has been performed on the blue channel of the image. The right side of the figure shows the segmented nuclei in different colors, each corresponding to a different nucleus in the field of view. Each color represents a unique label assigned to the individual nuclei detected in the image. [>_]



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

# Load the image
image = io.imread('/mnt/data/hela-cells-8bit.png')

# Since it's a color image, we'll just take the blue chann
blue_channel = image[:, :, 2]

# Threshold the image to get the binary image for nuclei
thresh = filters.threshold_otsu(blue_channel)
binary = blue_channel > thresh

# Remove small objects and perform morphological closing t
cleaned = morphology.remove_small_objects(binary, 50)
closed = morphology.closing(cleaned, morphology.disk(3))

# Label the image
label_image = measure.label(closed)

# Plot the original image and the binary image
fig, ax = plt.subplots(1, 2, figsize=(12, 6))
ax[0].imshow(image)
ax[0].set_title('Original Image')
ax[1].imshow(label_image, cmap='nipy_spectral')
ax[1].set_title('Segmented Nuclei')
plt.show()
```

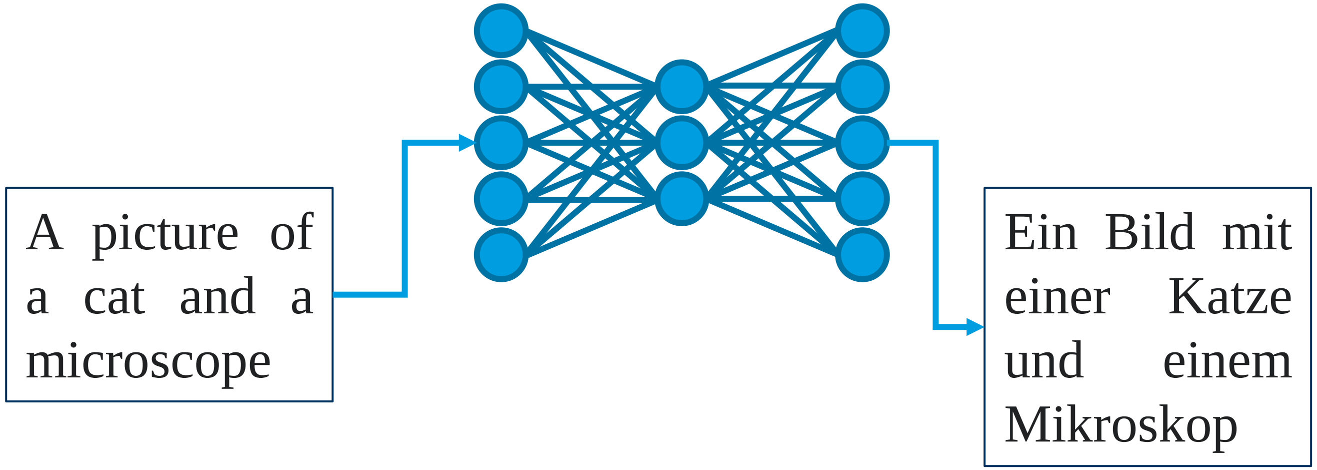
Generative Artificial Intelligence

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
- Image interpretation / analysis


A picture of a cat and a microscope

Ein Bild mit einer Katze und einem Mikroskop

---

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

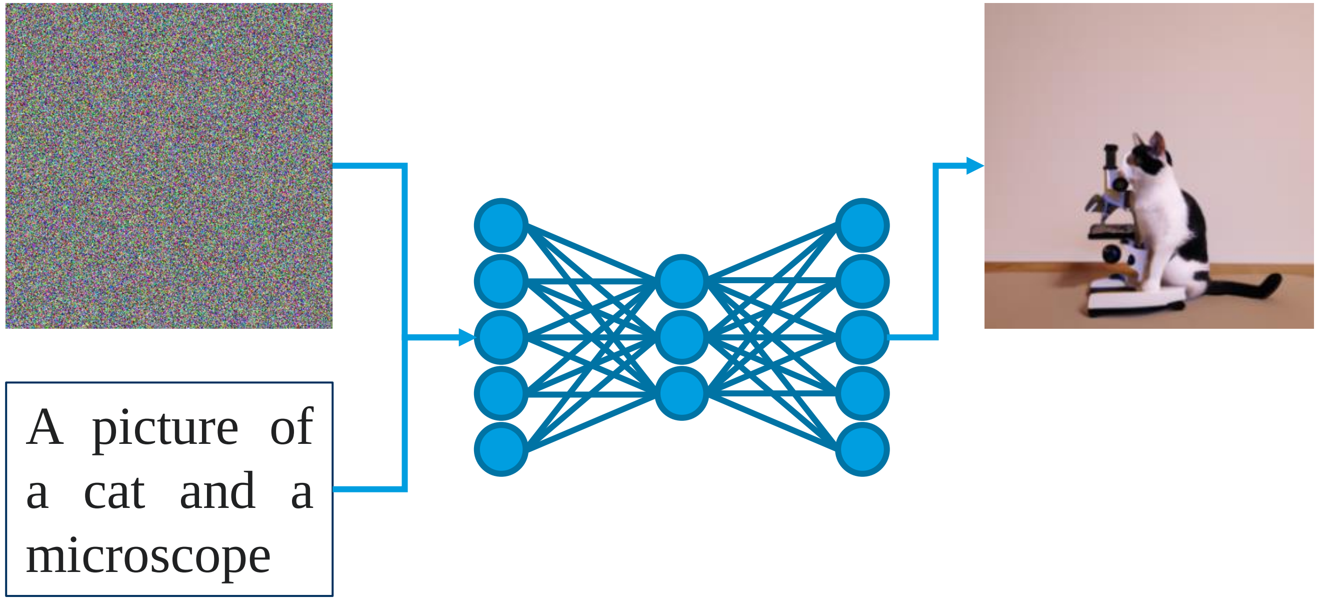
Generative Artificial Intelligence

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
- <u>Image generation</u>
- Image interpretation / analysis



A picture of a cat and a microscope

---

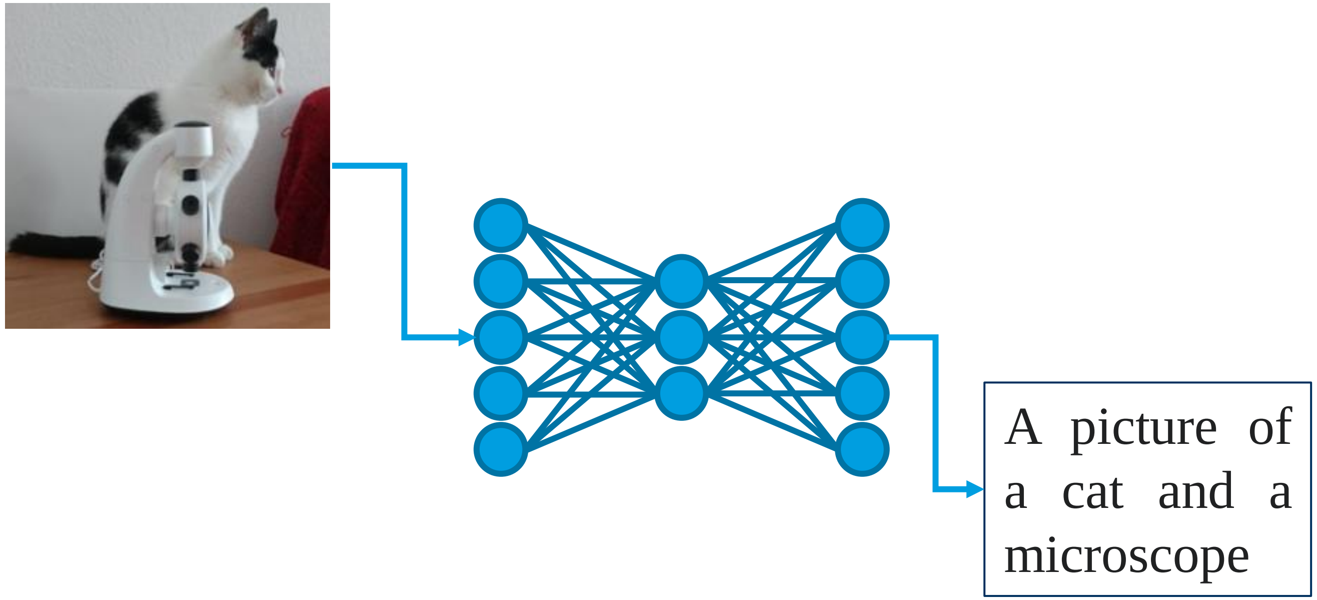
[1] source: https://en.wikipedia.org/wiki/Generative_artificial_intelligence

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)



A picture of a cat and a microscope

Use-cases
- Translating text
- Writing emails, text, grant proposals
- Summarizing articles
- Writing code
- General question answering
- Image generation
- Image interpretation / analysis

---

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

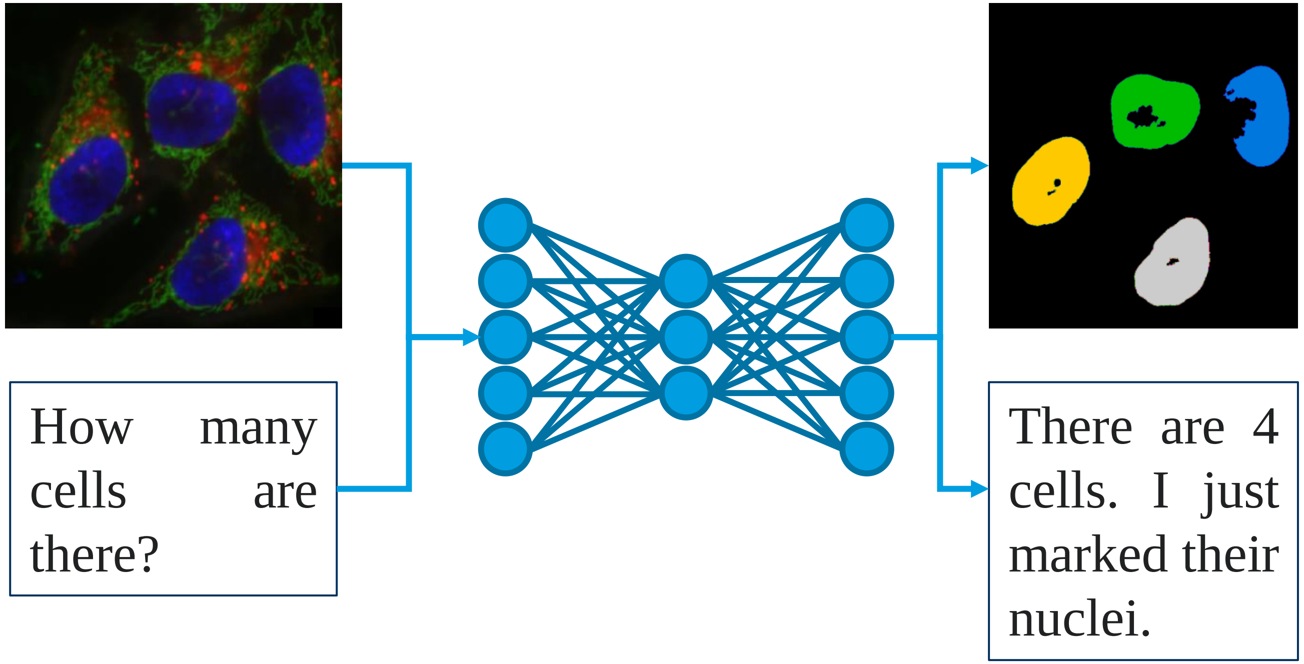
Generative Artificial Intelligence

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
- Image interpretation / analysis



How many cells are there?

There are 4 cells. I just marked their nuclei.

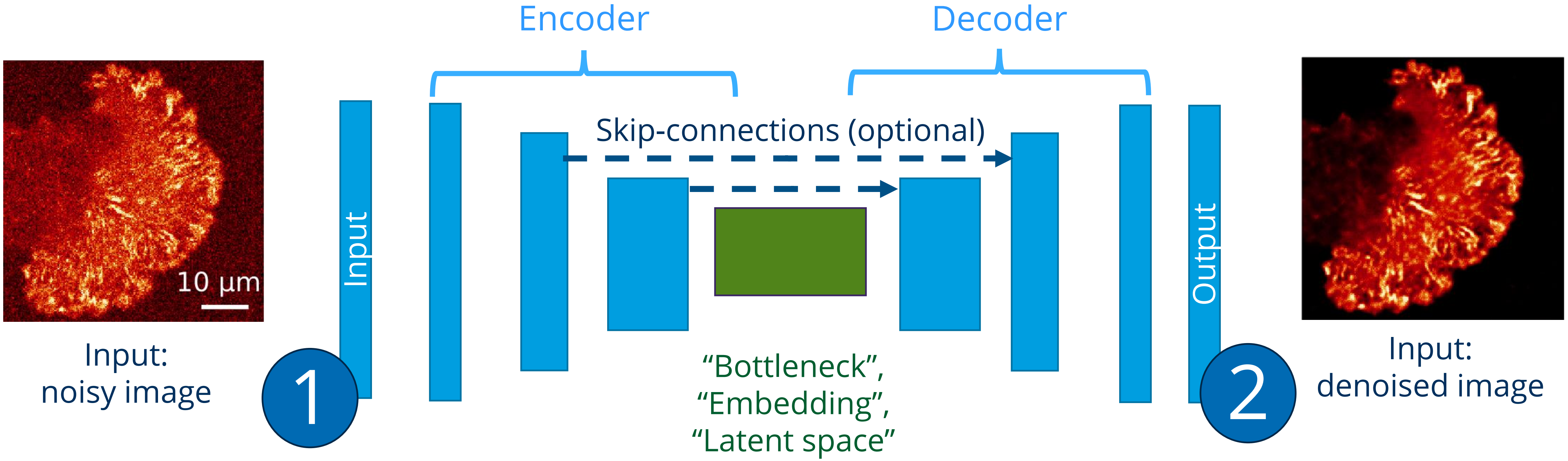
---

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

What‘s new?

Traditional architecture: Encoder-Decoder Networks

Related: „Auto-encoder“, „Variational Auto-Encoder“, „U-Net“



Encoder

Decoder

Skip-connections (optional)

Input

Output

10 µm

Input:
noisy image

①

“Bottleneck”,
“Embedding”,
“Latent space”

②

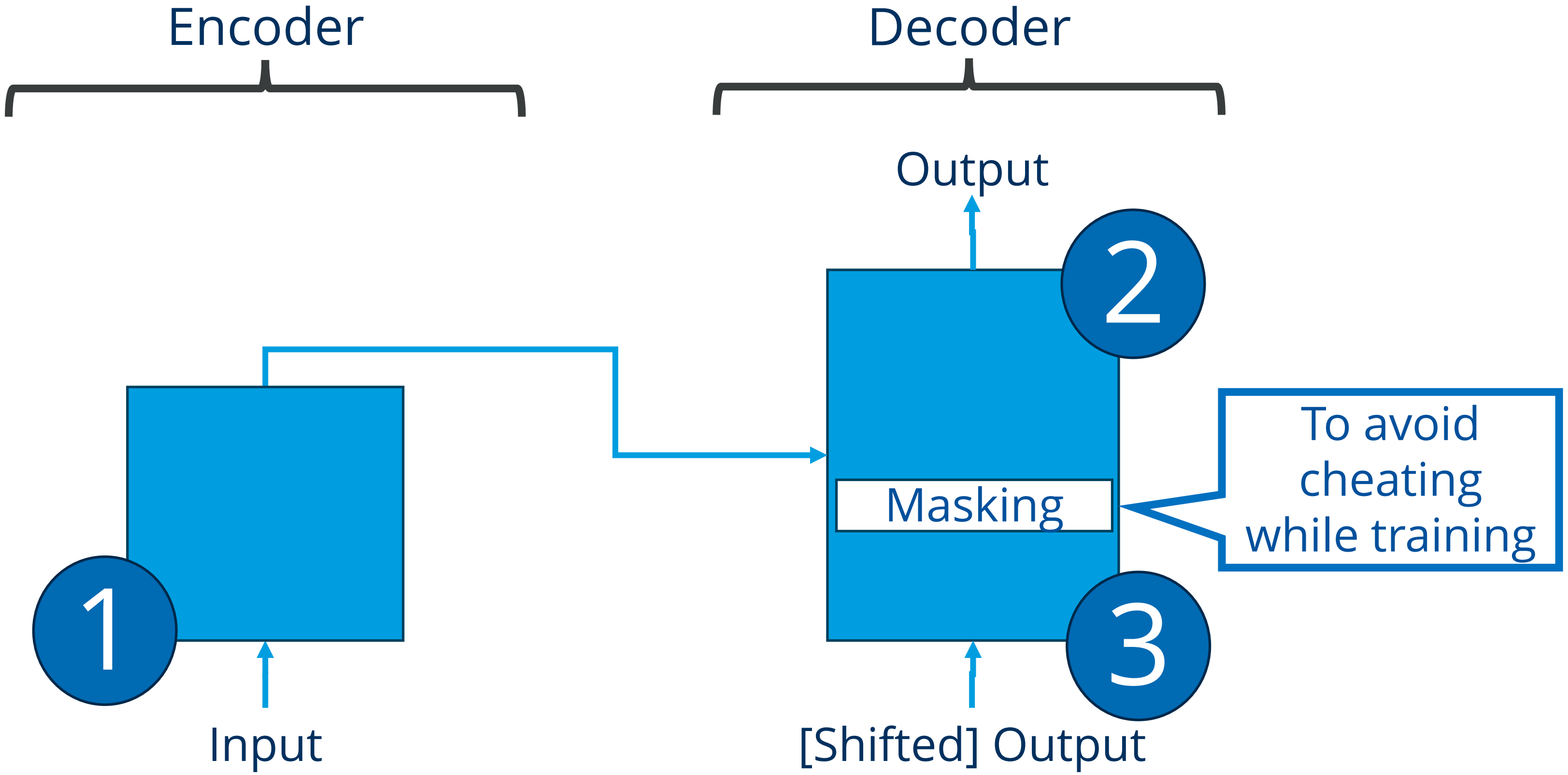
Input:
denoised image

Images demonstrating noise2void, cropped from https://github.com/HenriquesLab/ZeroCostDL4Mic/blob/master/Wiki_files/FigureS2.png license: MIT

What‘s new?

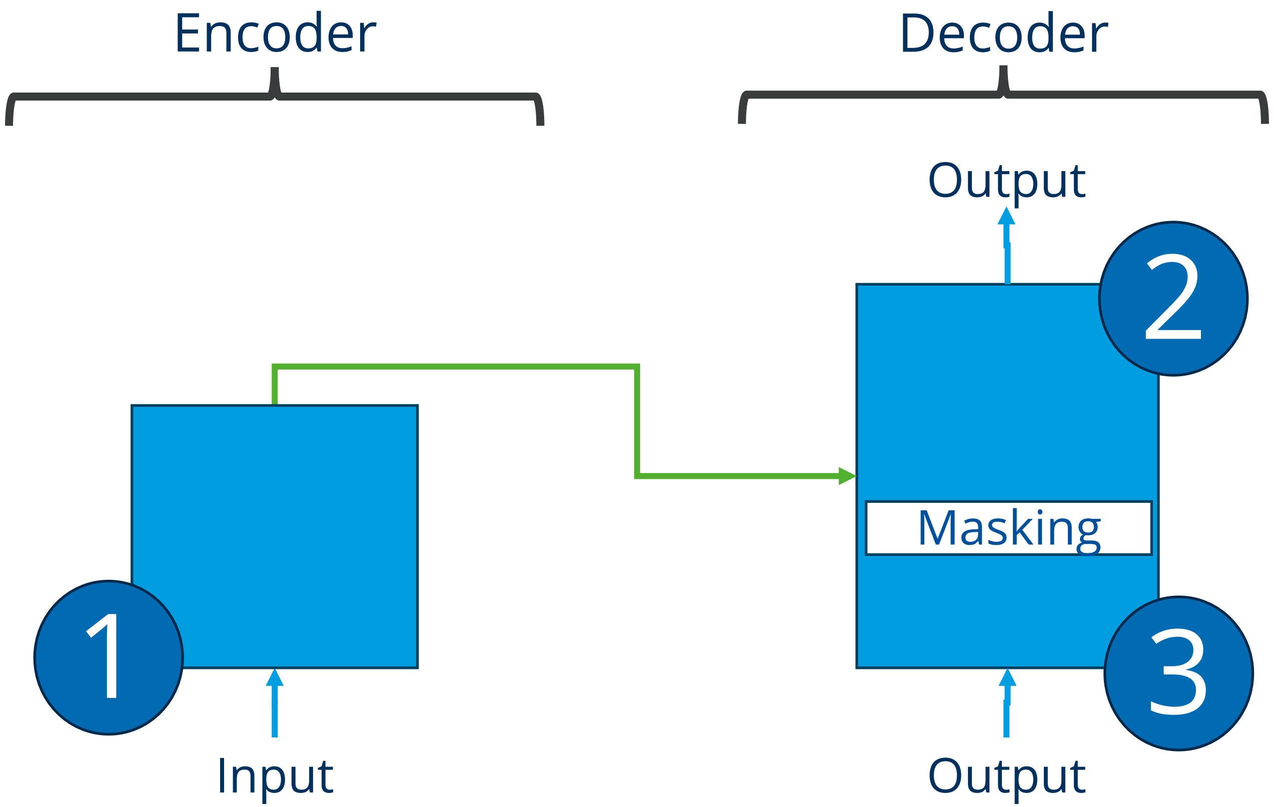
LLMs use the transformer neural network architecture

Related: Generative Pretrained Transformer (GPT)



What‘s new?

LLMs were originally developed for translation tasks and next-word prediction.



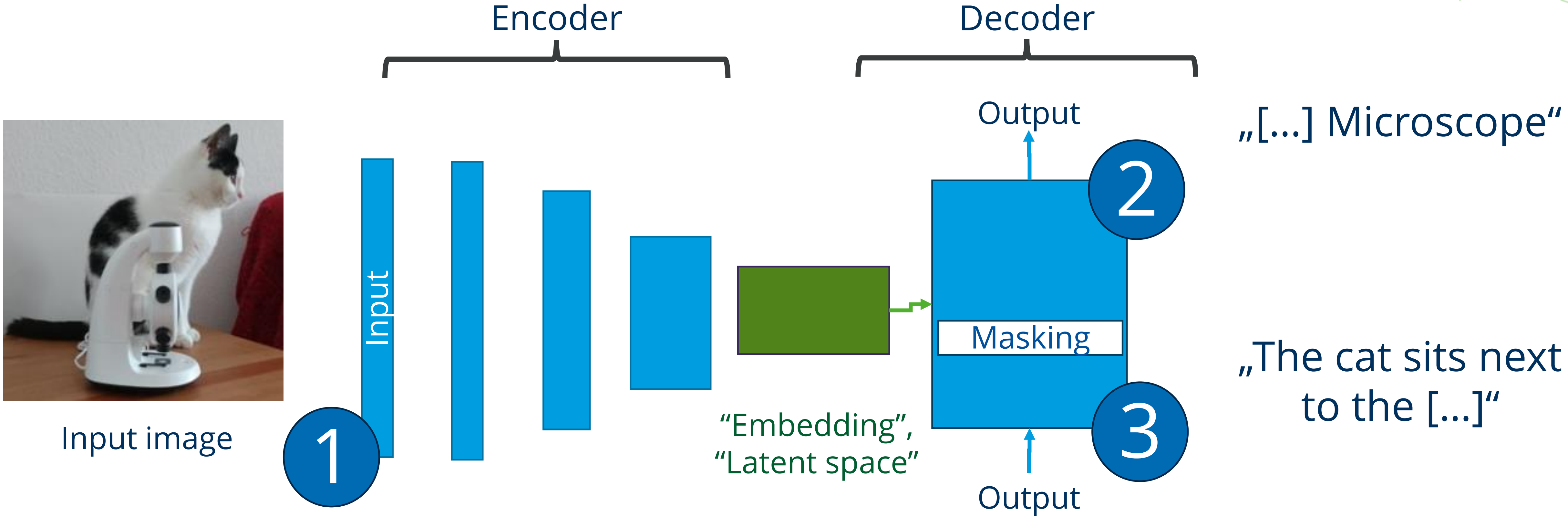
„Die Katze sitzt neben dem Mikroskop.“

„The cat sits next to the […]“

„[…] Microscope“

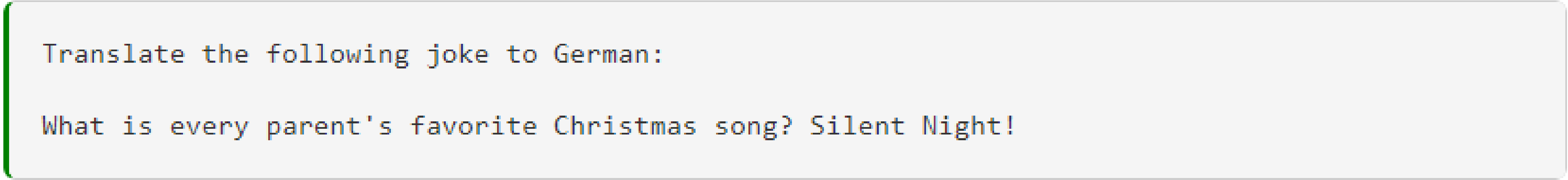
What‘s new?

VLMs use combinations of traditional neural network architectures and transformers.

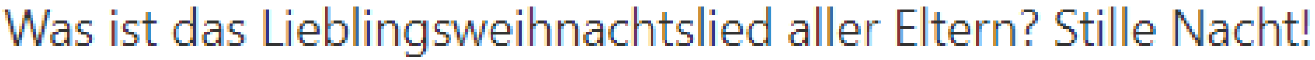


What are large language models good in?

Translation tasks



```
Translate the following joke to German:

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

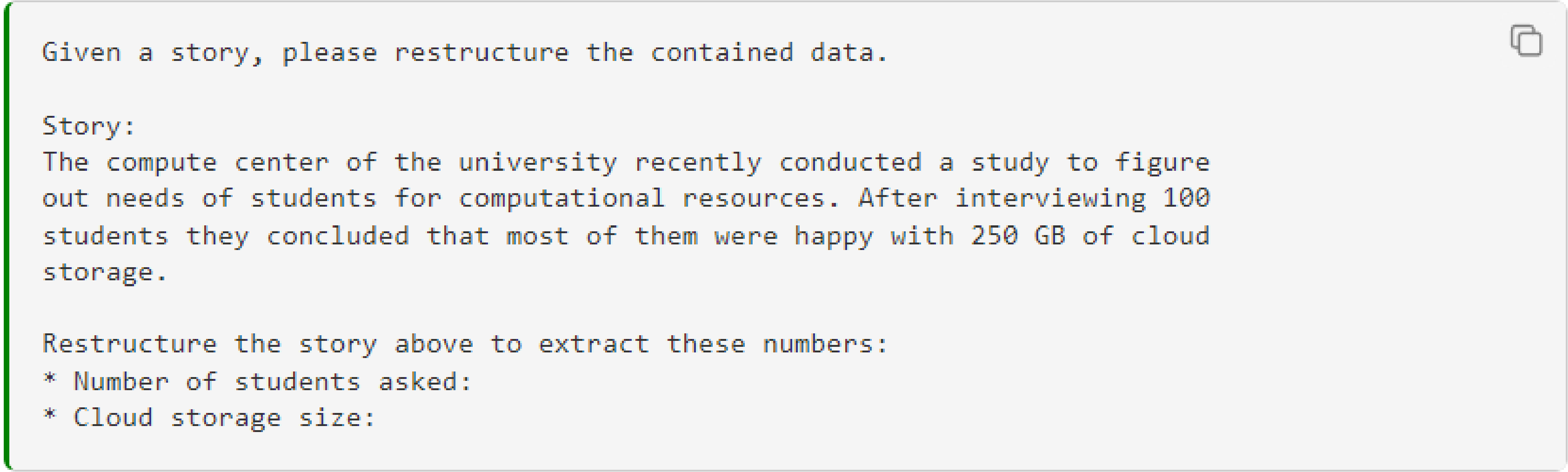


Was ist das Lieblingsweihnachtslied aller Eltern? Stille Nacht!

https://scads.github.io/prompt-engineering-basics-2024/20_use_cases.html

What are large language models good in?

Restructuring data / information retrieval

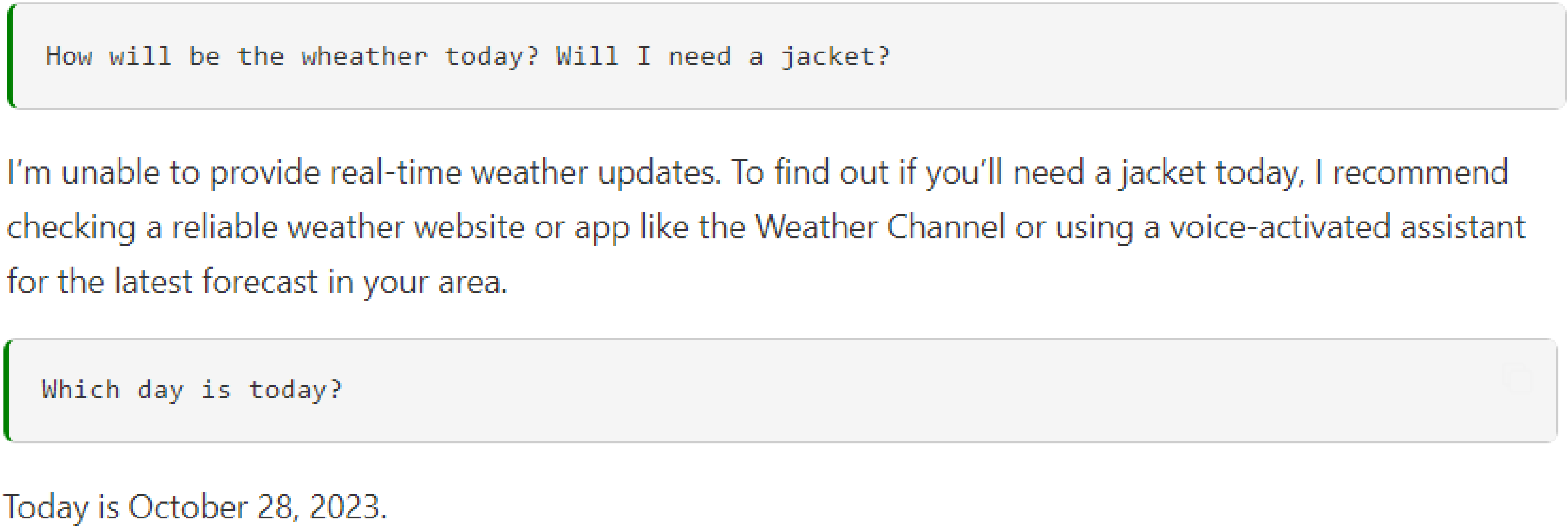


```
Given a story, please restructure the contained data.

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 (e.g. for recent information)



```
How will be the wheather today? Will I need a jacket?
```

I'm unable to provide real-time weather updates. To find out if you'll need a jacket today, I recommend checking a reliable weather website or app like the Weather Channel or using a voice-activated assistant for the latest forecast in your area.

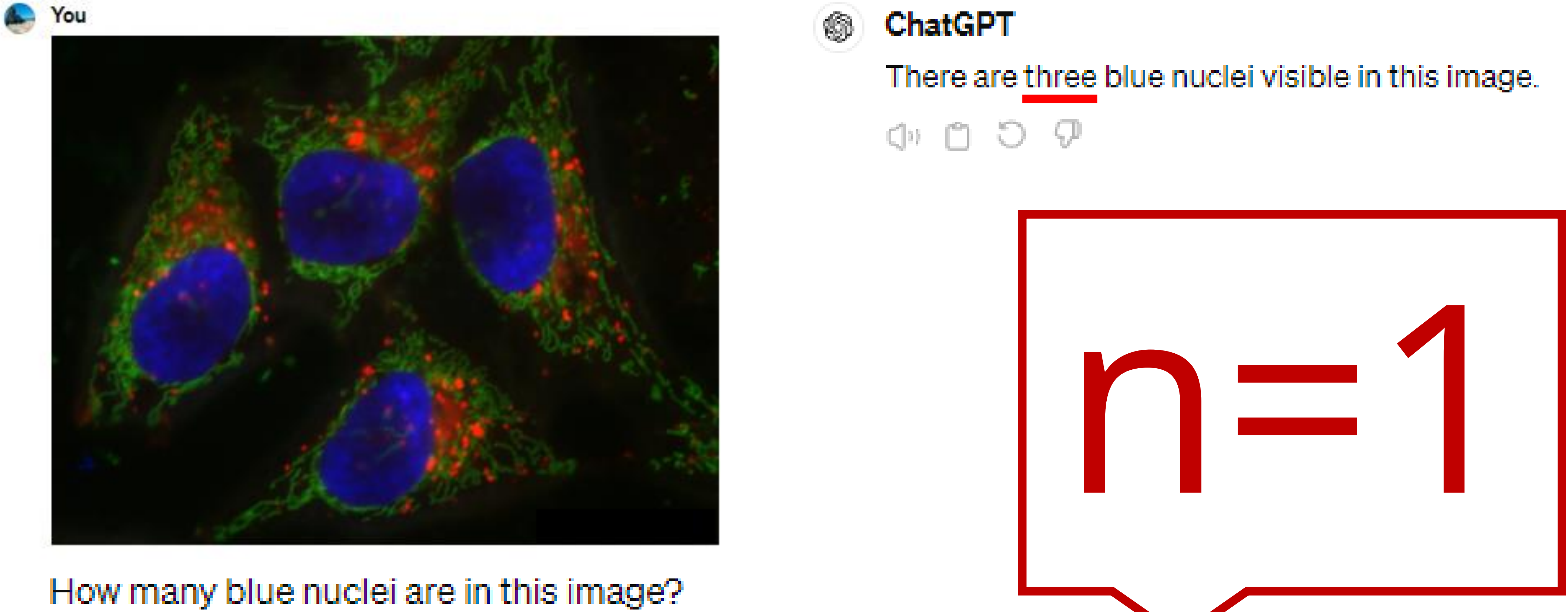
```
Which day is today?
```

Today is October 28, 2023.

https://scads.github.io/prompt-engineering-basics-2024/20_use_cases.html

Vision language models for counting objects

LLMs were not exactly built for this use-case…



You

How many blue nuclei are in this image?

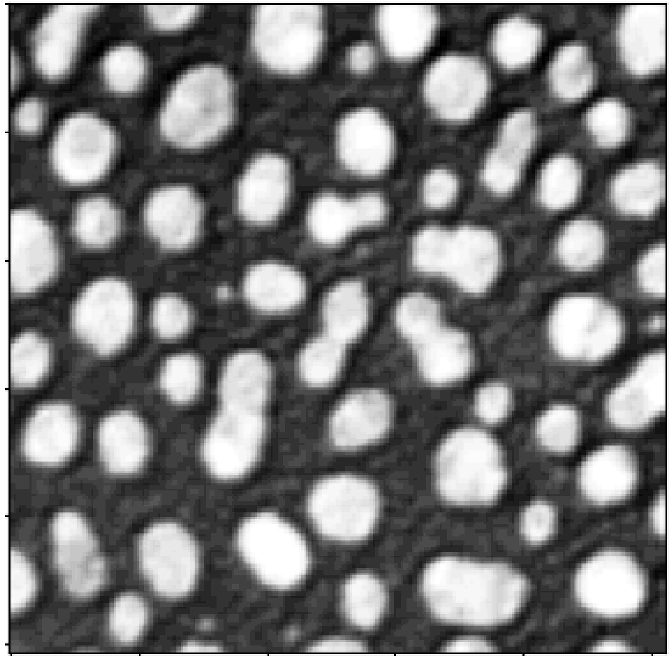
ChatGPT

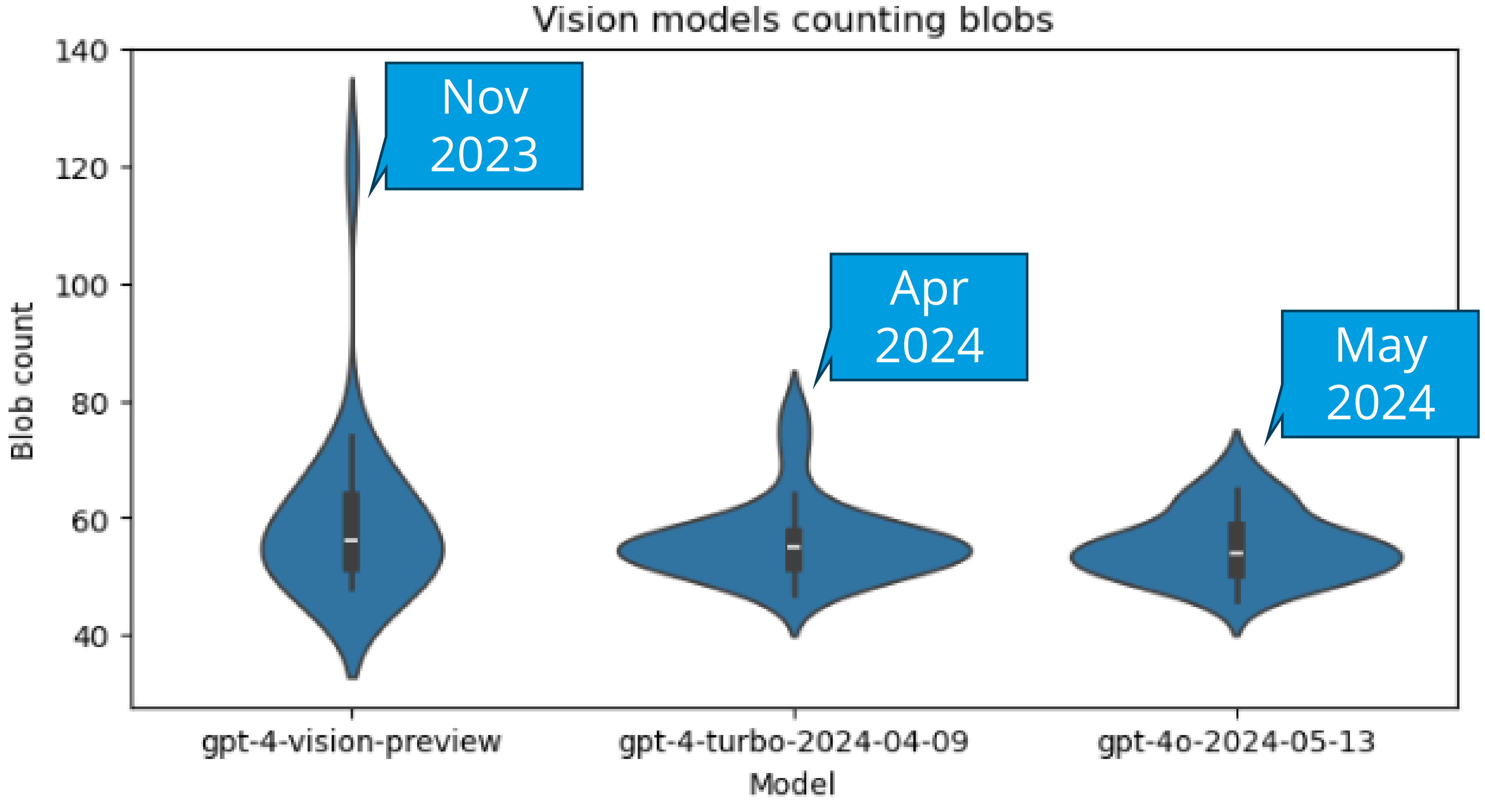
There are three blue nuclei visible in this image.

n=1

Vision language models for counting objects

Prompt: „Analyse the following image by counting the bright blobs. Respond with the number only. “ (n=25)





Nov 2023

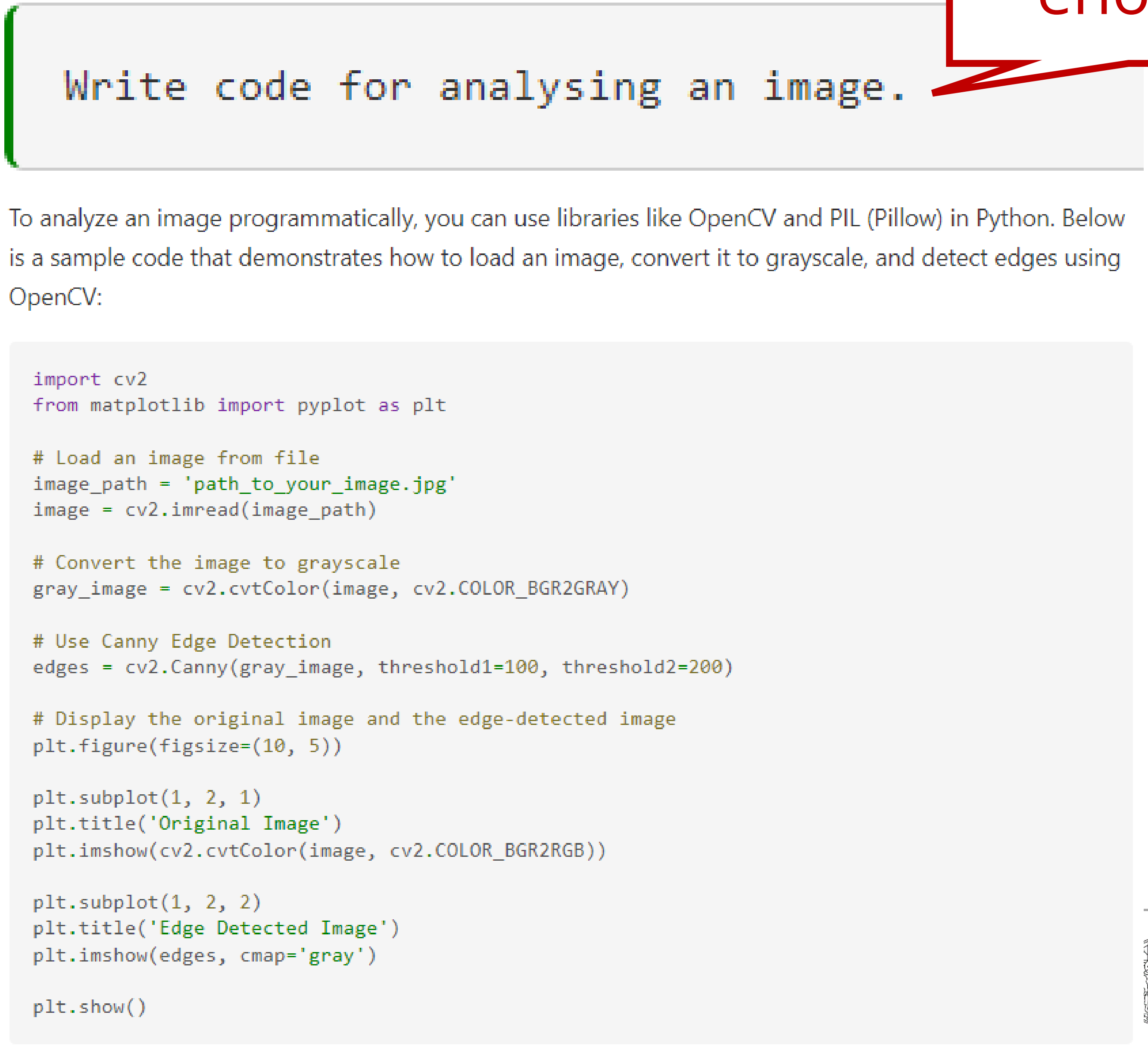
Apr 2024

May 2024

https://scads.github.io/generative-ai-notebooks/80_benchmarking_llms/20_vision_models.html

Prompt engineering

Unspecific questions lead to random-ish responses.



`Write code for analysing an image.`

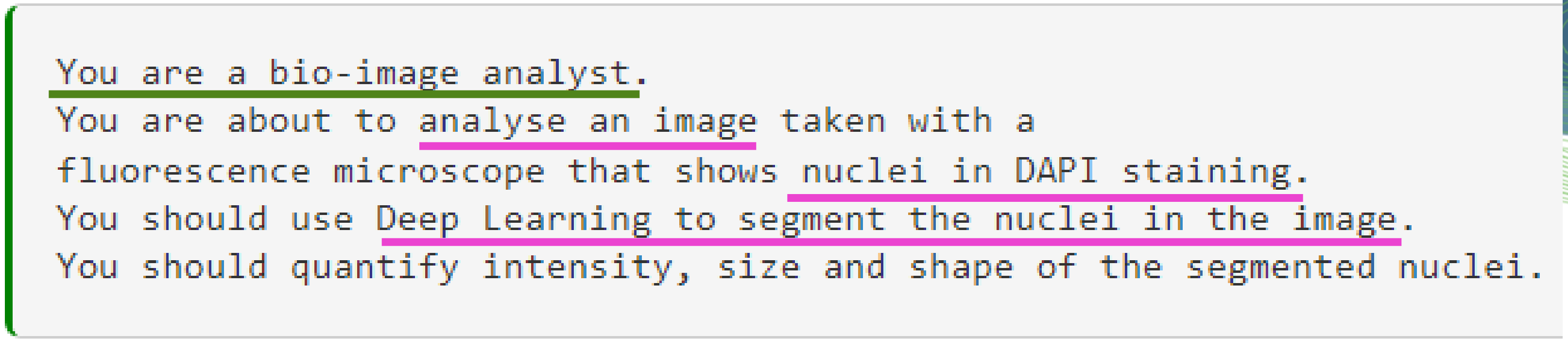
Not specific enough?

To analyze an image programmatically, you can use libraries like OpenCV and PIL (Pillow) in Python. Below is a sample code that demonstrates how to load an image, convert it to grayscale, and detect edges using OpenCV:

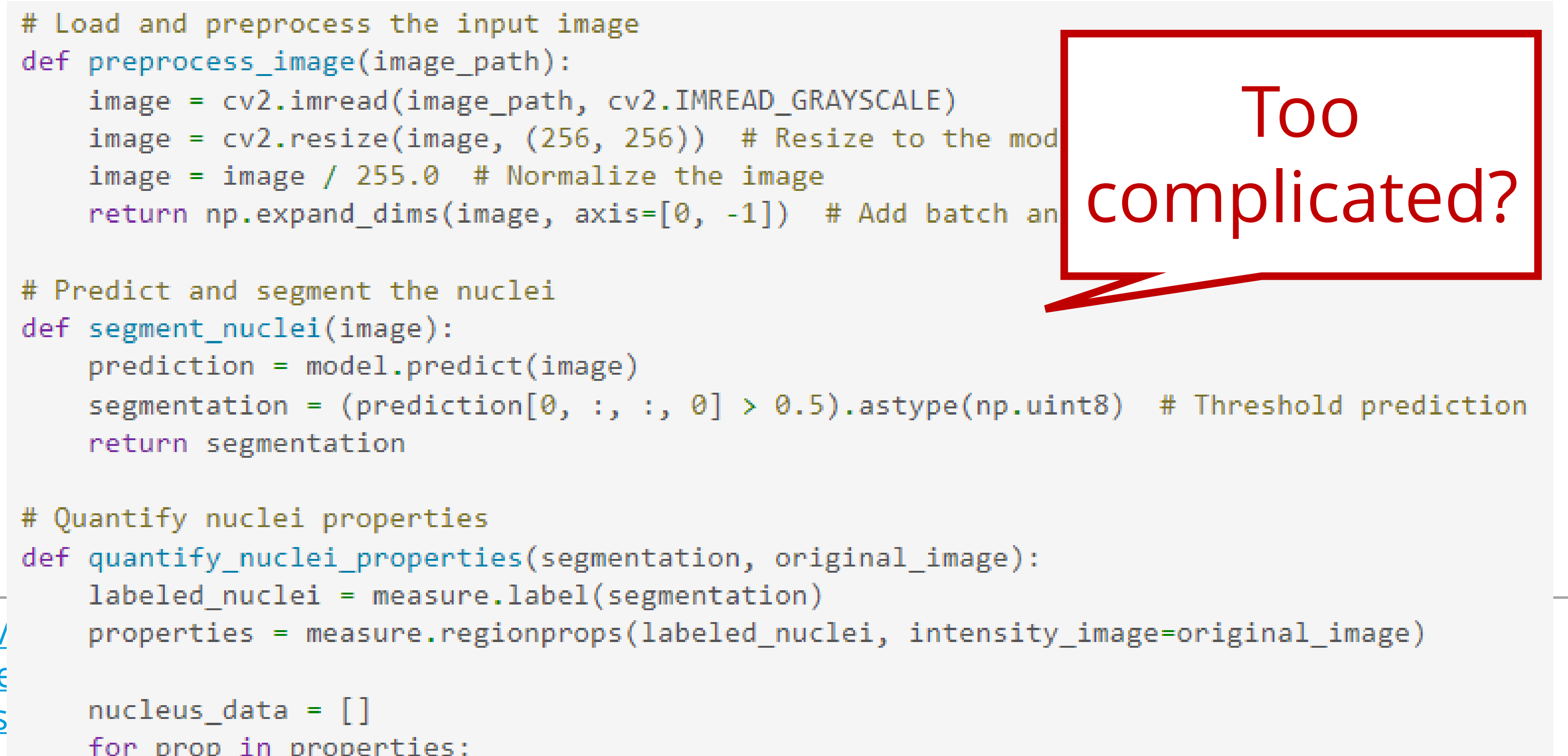
```
import cv2
from matplotlib import pyplot as plt

# Load an image from file
image_path = 'path_to_your_image.jpg'
image = cv2.imread(image_path)

# Convert the image to grayscale
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

# Use Canny Edge Detection
edges = cv2.Canny(gray_image, threshold1=100, threshold2=200)

# Display the original image and the edge-detected image
plt.figure(figsize=(10, 5))

plt.subplot(1, 2, 1)
plt.title('Original Image')
plt.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2RGB))

plt.subplot(1, 2, 2)
plt.title('Edge Detected Image')
plt.imshow(edges, cmap='gray')

plt.show()
```

Prompt engineering

- Define the role of the language model,
- Define the Task,
- be specific!
- Give examples (if possible)



```
You are a bio-image analyst.
You are about to analyse an image taken with a
fluorescence microscope that shows nuclei in DAPI staining.
You should use Deep Learning to segment the nuclei in the image.
You should quantify intensity, size and shape of the segmented nuclei.
```

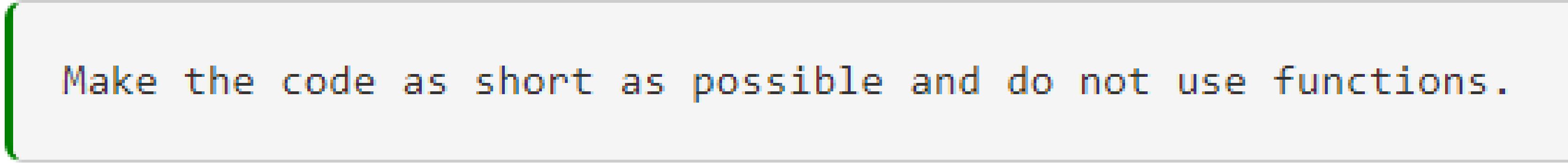


```
# Load and preprocess the input image
def preprocess_image(image_path):
    image = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)
    image = cv2.resize(image, (256, 256))  # Resize to the mod
    image = image / 255.0  # Normalize the image
    return np.expand_dims(image, axis=[0, -1])  # Add batch an

# Predict and segment the nuclei
def segment_nuclei(image):
    prediction = model.predict(image)
    segmentation = (prediction[0, :, :, 0] > 0.5).astype(np.uint8)  # Threshold prediction
    return segmentation

# Quantify nuclei properties
def quantify_nuclei_properties(segmentation, original_image):
    labeled_nuclei = measure.label(segmentation)
    properties = measure.regionprops(labeled_nuclei, intensity_image=original_image)

    nucleus_data = []
    for prop in properties:
```

Too complicated?

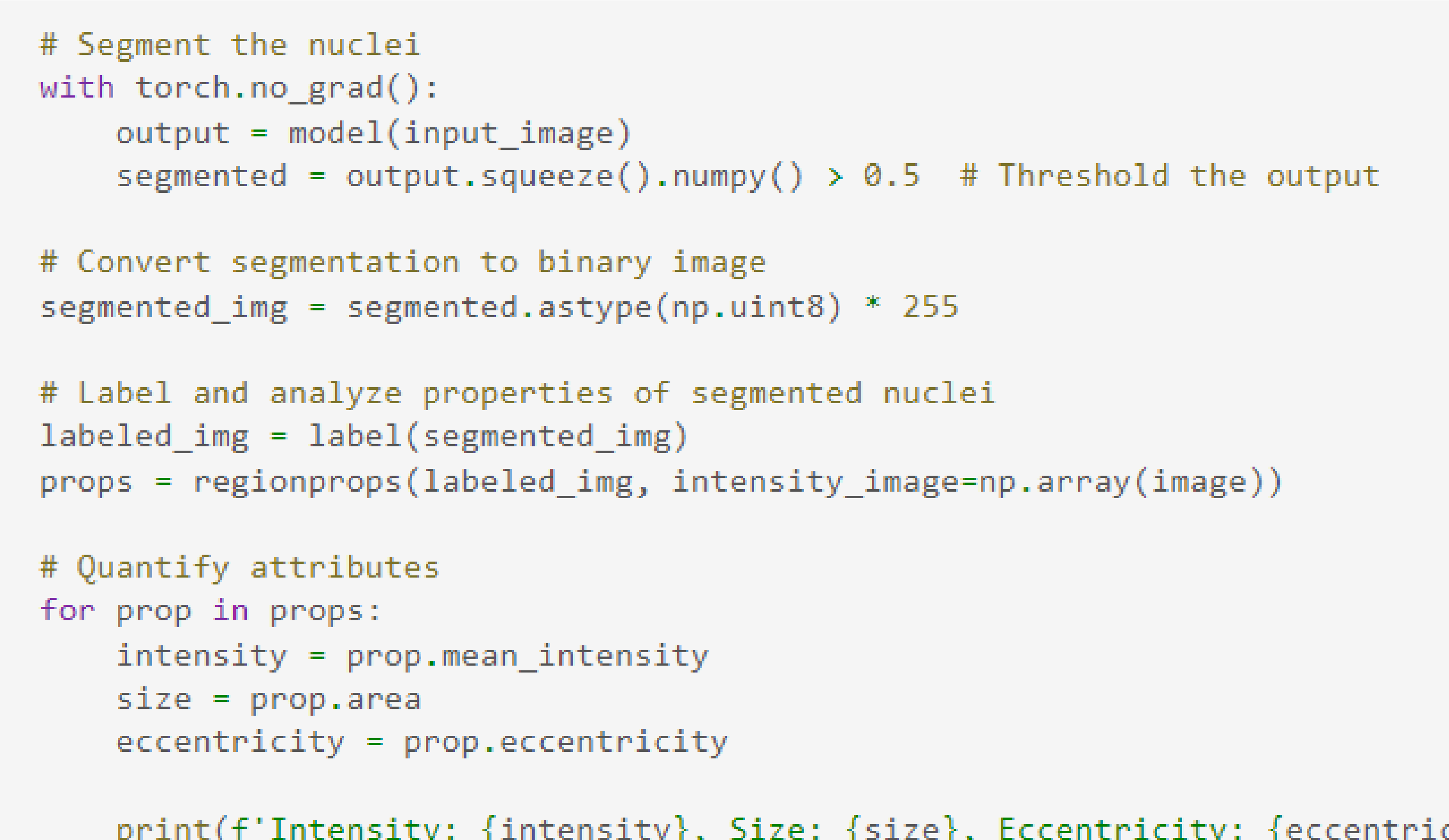
Prompt engineering

- Define the output style.



```
Make the code as short as possible and do not use functions.
```

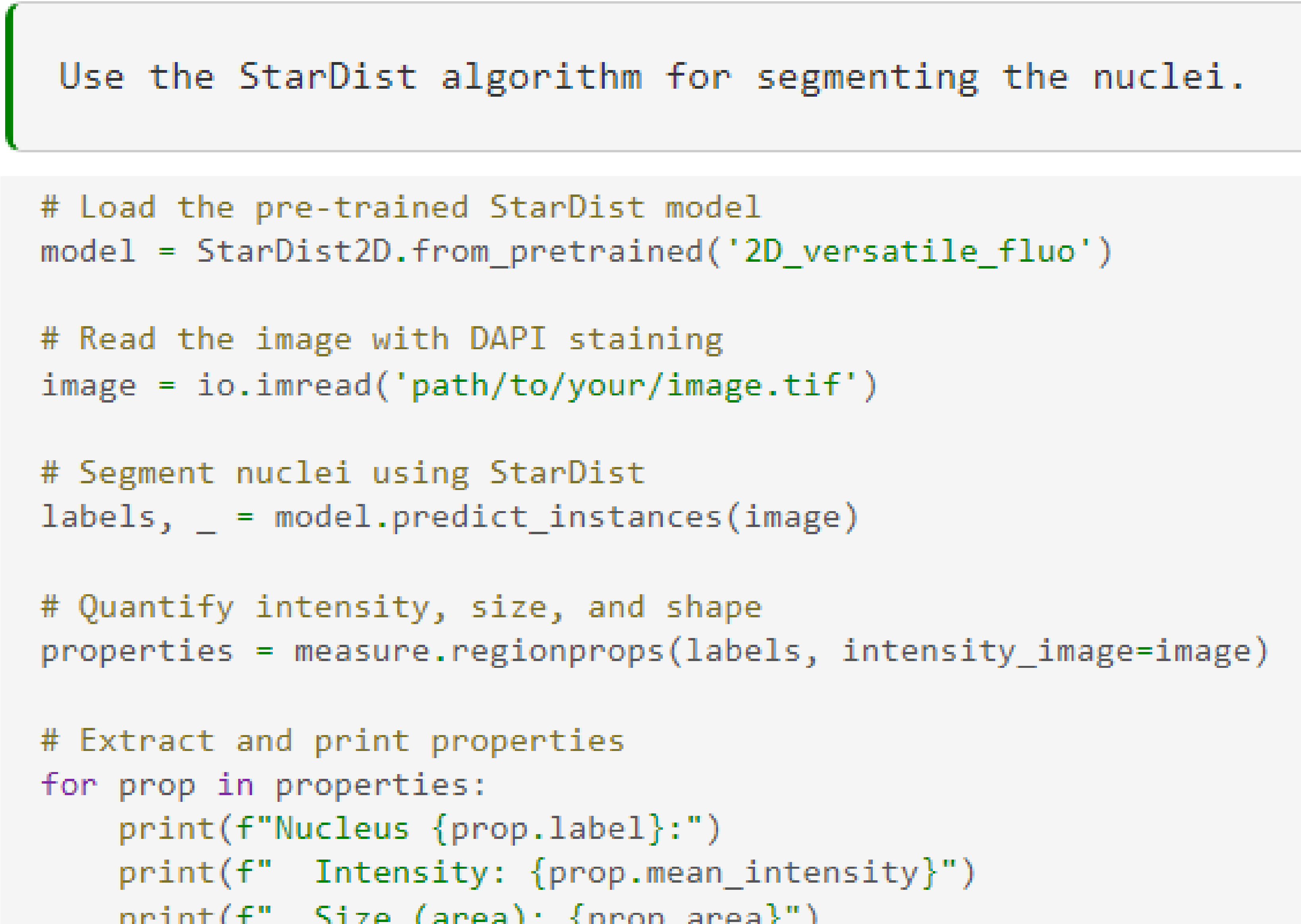
Wrong algorithm used?



```
# Segment the nuclei
with torch.no_grad():
    output = model(input_image)
    segmented = output.squeeze().numpy() > 0.5  # Threshold the output

# Convert segmentation to binary image
segmented_img = segmented.astype(np.uint8) * 255

# Label and analyze properties of segmented nuclei
labeled_img = label(segmented_img)
props = regionprops(labeled_img, intensity_image=np.array(image))

# Quantify attributes
for prop in props:
    intensity = prop.mean_intensity
    size = prop.area
    eccentricity = prop.eccentricity

    print(f'Intensity: {intensity}, Size: {size}, Eccentricity: {eccentric
```

Prompt engineering

- Use image-analysis terminology



```
Use the StarDist algorithm for segmenting the nuclei.
```

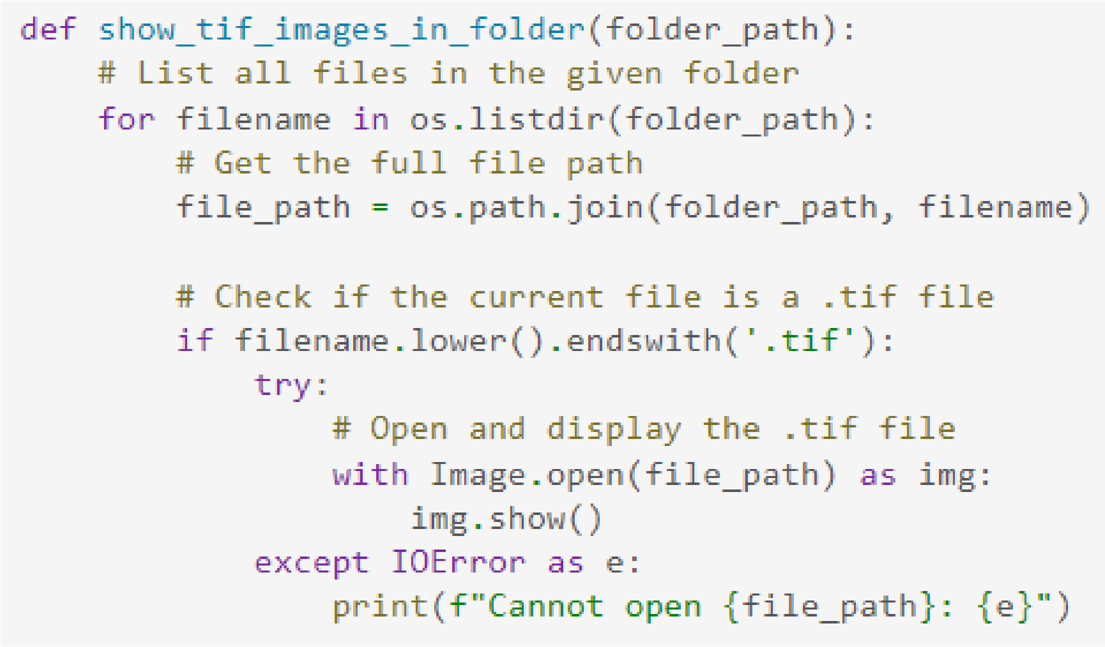
```
# Load the pre-trained StarDist model
model = StarDist2D.from_pretrained('2D_versatile_fluo')

# Read the image with DAPI staining
image = io.imread('path/to/your/image.tif')

# Segment nuclei using StarDist
labels, _ = model.predict_instances(image)

# Quantify intensity, size, and shape
properties = measure.regionprops(labels, intensity_image=image)

# Extract and print properties
for prop in properties:
    print(f"Nucleus {prop.label}:")
    print(f"  Intensity: {prop.mean_intensity}")
    print(f"  Size (area): {prop.area}")
```

Learn the technical language of the task at hand!

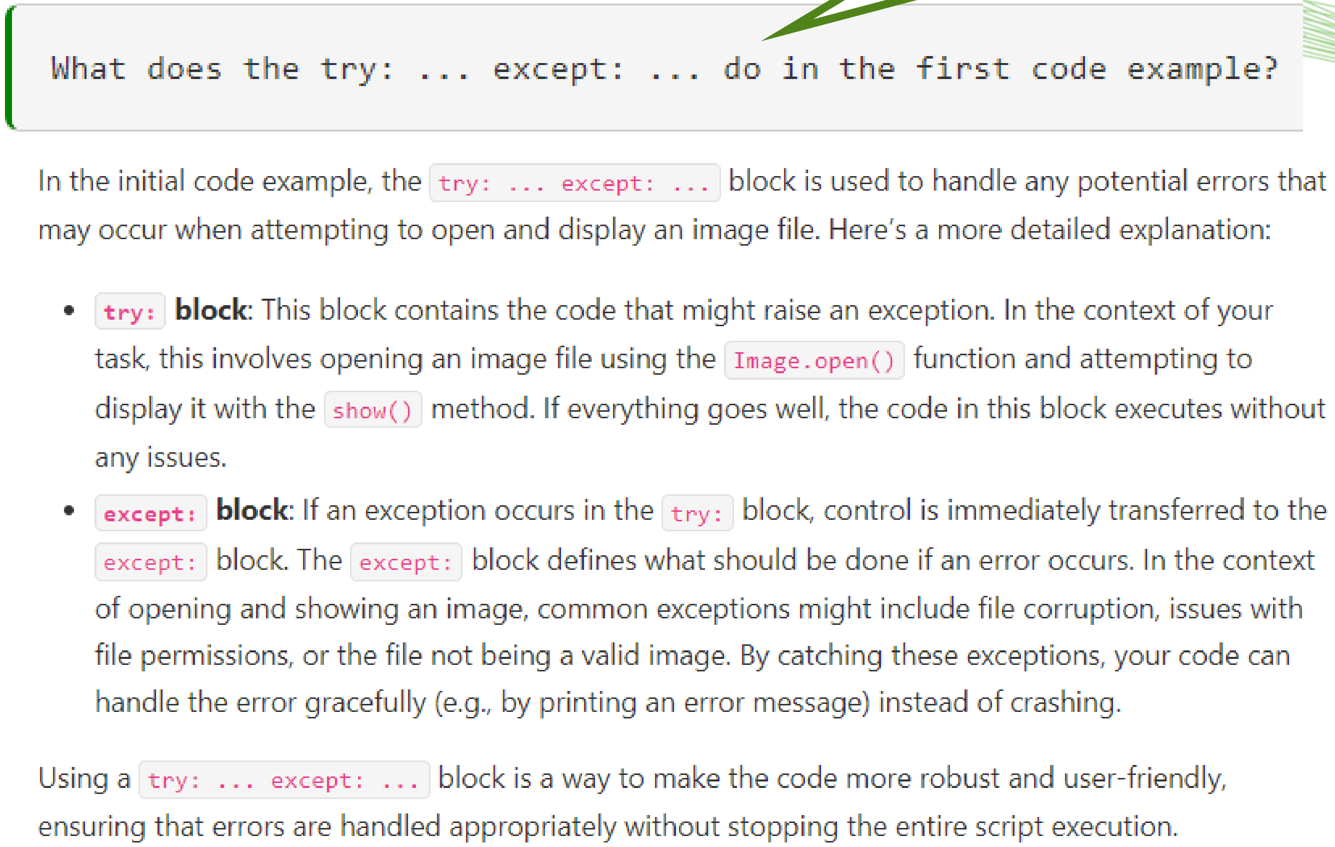
Ask for explanations!

In case you don‘t understand the code, ask for explanations!

Endless patience!



```
def show_tif_images_in_folder(folder_path):
    # List all files in the given folder
    for filename in os.listdir(folder_path):
        # Get the full file path
        file_path = os.path.join(folder_path, filename)

        # Check if the current file is a .tif file
        if filename.lower().endswith('.tif'):
            try:
                # Open and display the .tif file
                with Image.open(file_path) as img:
                    img.show()
            except IOError as e:
                print(f"Cannot open {file_path}: {e}")
```



What does the try: ... except: ... do in the first code example?

In the initial code example, the `try: ... except: ...` block is used to handle any potential errors that may occur when attempting to open and display an image file. Here's a more detailed explanation:

- `try:` **block**: This block contains the code that might raise an exception. In the context of your task, this involves opening an image file using the `Image.open()` function and attempting to display it with the `show()` method. If everything goes well, the code in this block executes without any issues.
- `except:` **block**: If an exception occurs in the `try:` block, control is immediately transferred to the `except:` block. The `except:` block defines what should be done if an error occurs. In the context of opening and showing an image, common exceptions might include file corruption, issues with file permissions, or the file not being a valid image. By catching these exceptions, your code can handle the error gracefully (e.g., by printing an error message) instead of crashing.

Using a `try: ... except: ...` block is a way to make the code more robust and user-friendly, ensuring that errors are handled appropriately without stopping the entire script execution.

Work step-by-step

If you are new in programming, work step-by-step.

- Prompt for single processing steps.
- Ask the LLM to explain what the code does.
- *Understand* the individual code snippets.

- Combine the code later (or ask the LLM to do it).

Benchmarking LLMs for Bio-image Analysis

Which LLM should I pay for?

Is GPT-4o better than Claude?

Do LLMs introduce bias?

How can we validate LLMs for BiA?

How *good* are LLMs for Bio-image Analysis?

Can LLMs act according to good scientific practice?

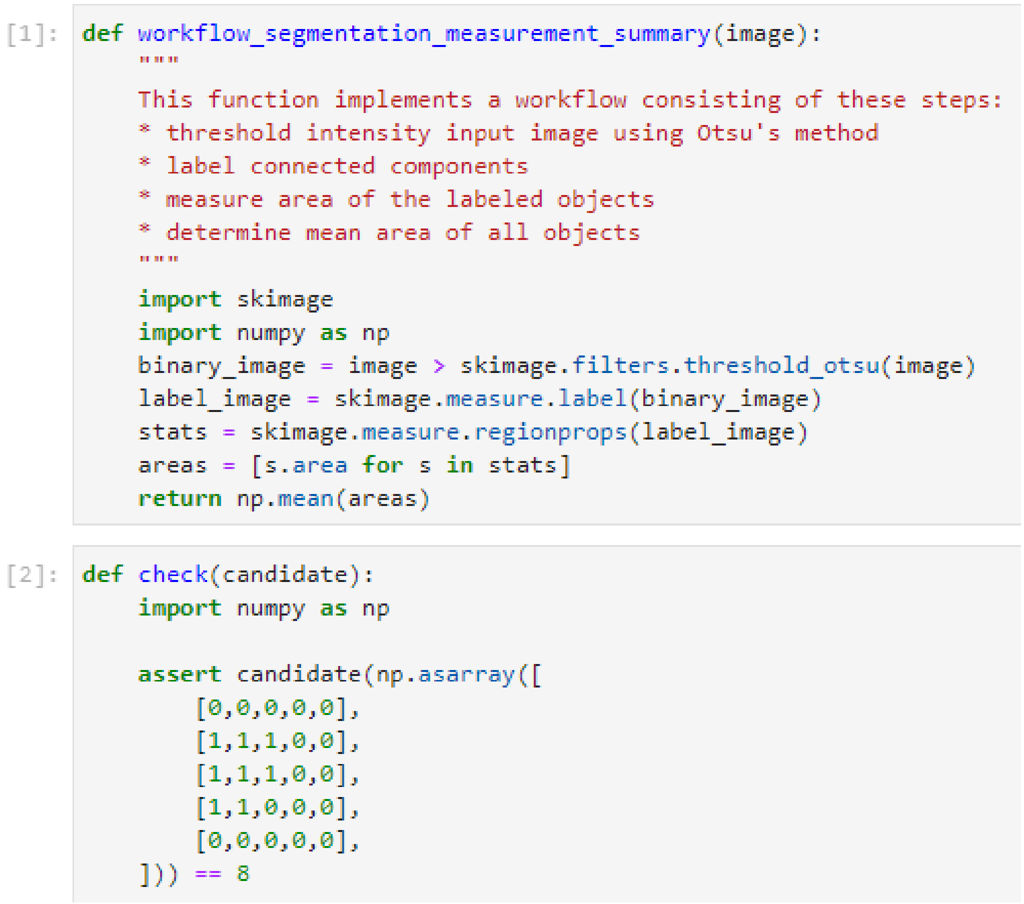
Will we all be replaced by AI anytime soon?

How can we validate specific LLM output without ground truth?

What are the limits of the LLM technology?

Benchmarking LLMs for Bio-image Analysis

Example test-case inspired by HumaEval (Chen et al 2021)



```
[1]: def workflow_segmentation_measurement_summary(image):
         """
         This function implements a workflow consisting of these steps:
         * threshold intensity input image using Otsu's method
         * label connected components
         * measure area of the labeled objects
         * determine mean area of all objects
         """
         import skimage
         import numpy as np
         binary_image = image > skimage.filters.threshold_otsu(image)
         label_image = skimage.measure.label(binary_image)
         stats = skimage.measure.regionprops(label_image)
         areas = [s.area for s in stats]
         return np.mean(areas)
```

```
[2]: def check(candidate):
         import numpy as np

         assert candidate(np.asarray([
             [0,0,0,0,0],
             [1,1,1,0,0],
             [1,1,1,0,0],
             [1,1,0,0,0],
             [0,0,0,0,0],
         ])) == 8
```

Prompt

Reference solution

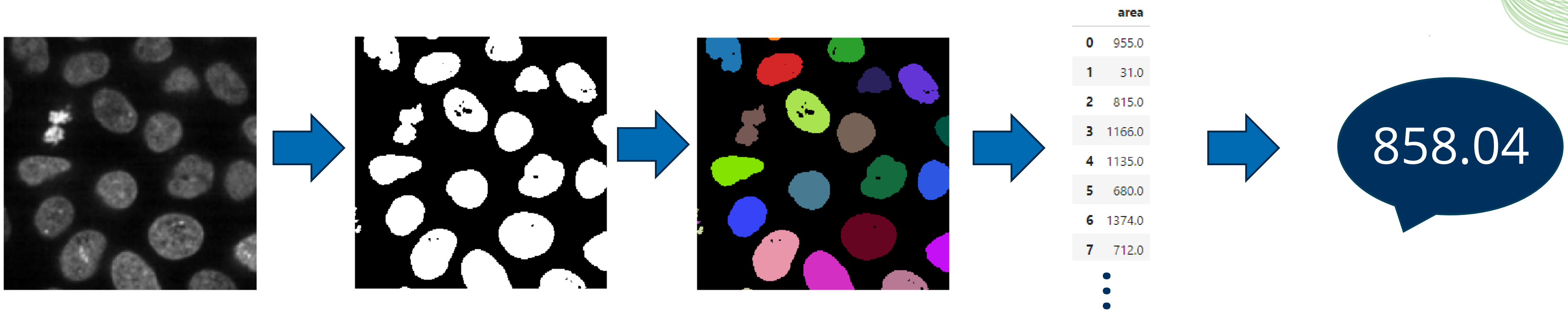
Unit test (excerpt)

We formulated 57 of such test-cases (yet)

Chen at al 2021: https://arxiv.org/abs/2107.03374
https://www.biorxiv.org/content/10.1101/2024.04.19.590278v1
https://github.com/haesleinhuepf/human-eval-bia

Benchmarking LLMs for Bio-image Analysis

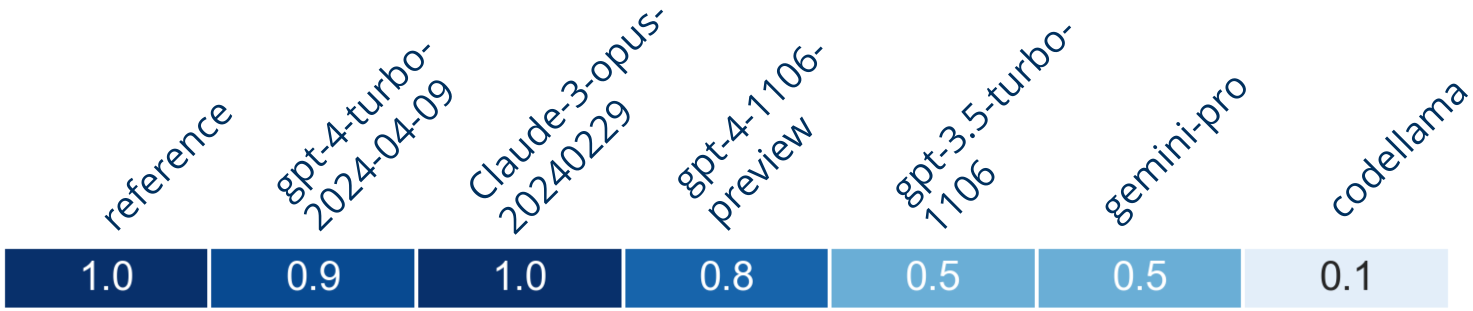
Use case: segment the image and measure the average area of objects.



	area
0	955.0
1	31.0
2	815.0
3	1166.0
4	1135.0
5	680.0
6	1374.0
7	712.0

⋮

Unit-test pass-rate (n=10):

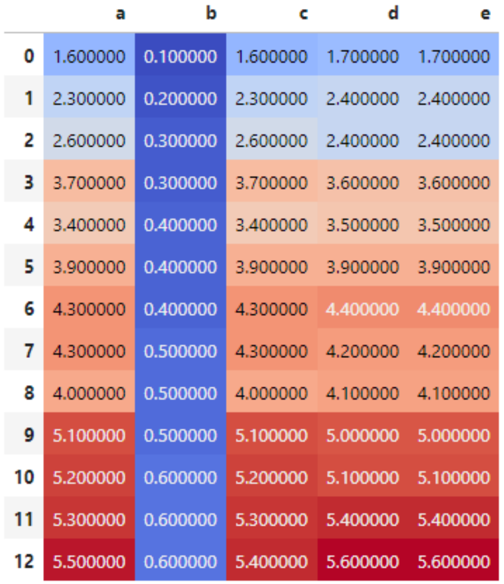


	reference	gpt-4-turbo-2024-04-09	Claude-3-opus-20240229	gpt-4-1106-preview	gpt-3.5-turbo-1106	gemini-pro	codellama
workflow_segmentation_measurement_summary	1.0	0.9	1.0	0.8	0.5	0.5	0.1

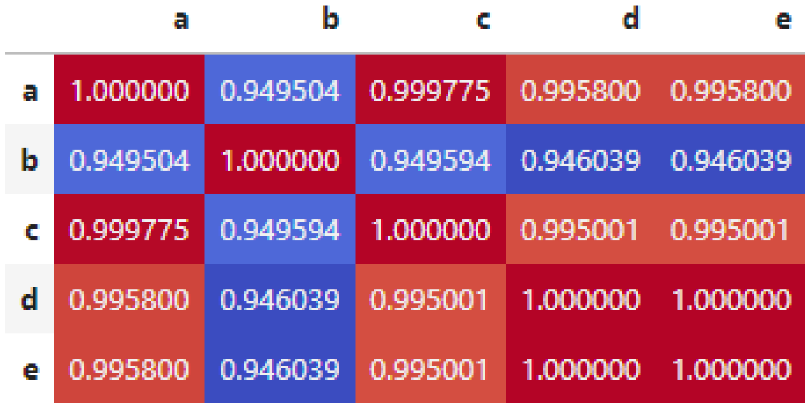
https://www.biorxiv.org/content/10.1101/2024.04.19.590278v1
https://github.com/haesleinhuepf/human-eval-bia

Benchmarking LLMs for Bio-image Analysis

Use-case: compute the correlation matrix

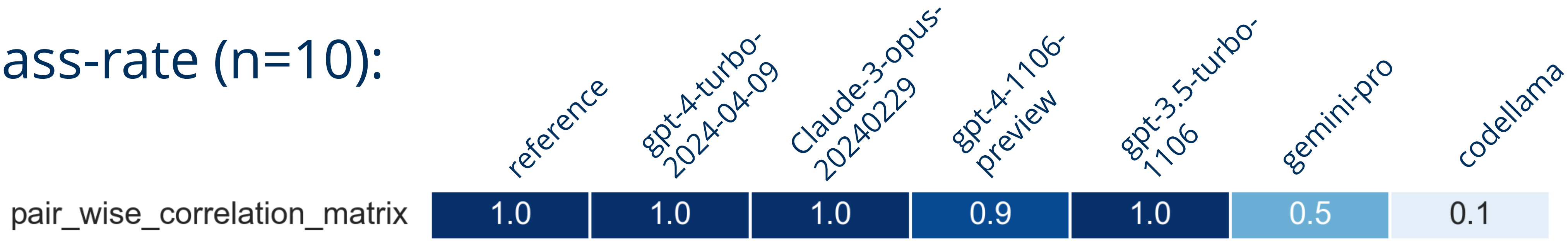


	a	b	c	d	e
0	1.600000	0.100000	1.600000	1.700000	1.700000
1	2.300000	0.200000	2.300000	2.400000	2.400000
2	2.600000	0.300000	2.600000	2.400000	2.400000
3	3.700000	0.300000	3.700000	3.600000	3.600000
4	3.400000	0.400000	3.400000	3.500000	3.500000
5	3.900000	0.400000	3.900000	3.900000	3.900000
6	4.300000	0.400000	4.300000	4.400000	4.400000
7	4.300000	0.500000	4.300000	4.200000	4.200000
8	4.000000	0.500000	4.000000	4.100000	4.100000
9	5.100000	0.500000	5.100000	5.000000	5.000000
10	5.200000	0.600000	5.200000	5.100000	5.100000
11	5.300000	0.600000	5.300000	5.400000	5.400000
12	5.500000	0.600000	5.400000	5.600000	5.600000



	a	b	c	d	e
a	1.000000	0.949504	0.999775	0.995800	0.995800
b	0.949504	1.000000	0.949594	0.946039	0.946039
c	0.999775	0.949594	1.000000	0.995001	0.995001
d	0.995800	0.946039	0.995001	1.000000	1.000000
e	0.995800	0.946039	0.995001	1.000000	1.000000

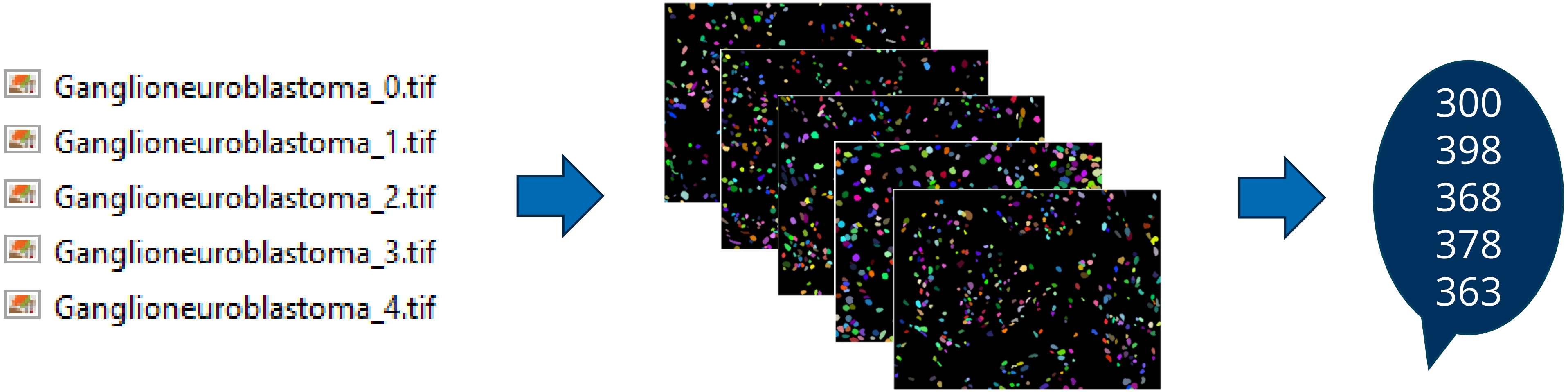
Unit-test pass-rate (n=10):



	reference	gpt-4-turbo-2024-04-09	Claude-3-opus-20240229	gpt-4-1106-preview	gpt-3.5-turbo-1106	gemini-pro	codellama
pair_wise_correlation_matrix	1.0	1.0	1.0	0.9	1.0	0.5	0.1

Benchmarking LLMs for Bio-image Analysis

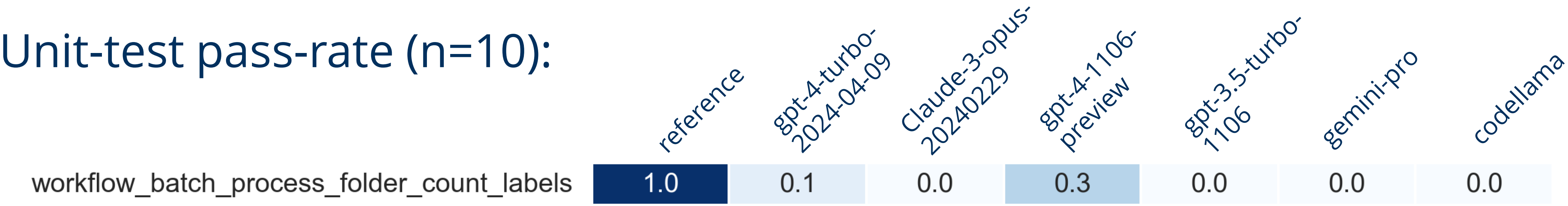
Use case: Count segmented objects in a folder of segmentation results.



Ganglioneuroblastoma_0.tif
Ganglioneuroblastoma_1.tif
Ganglioneuroblastoma_2.tif
Ganglioneuroblastoma_3.tif
Ganglioneuroblastoma_4.tif

300
398
368
378
363

Unit-test pass-rate (n=10):

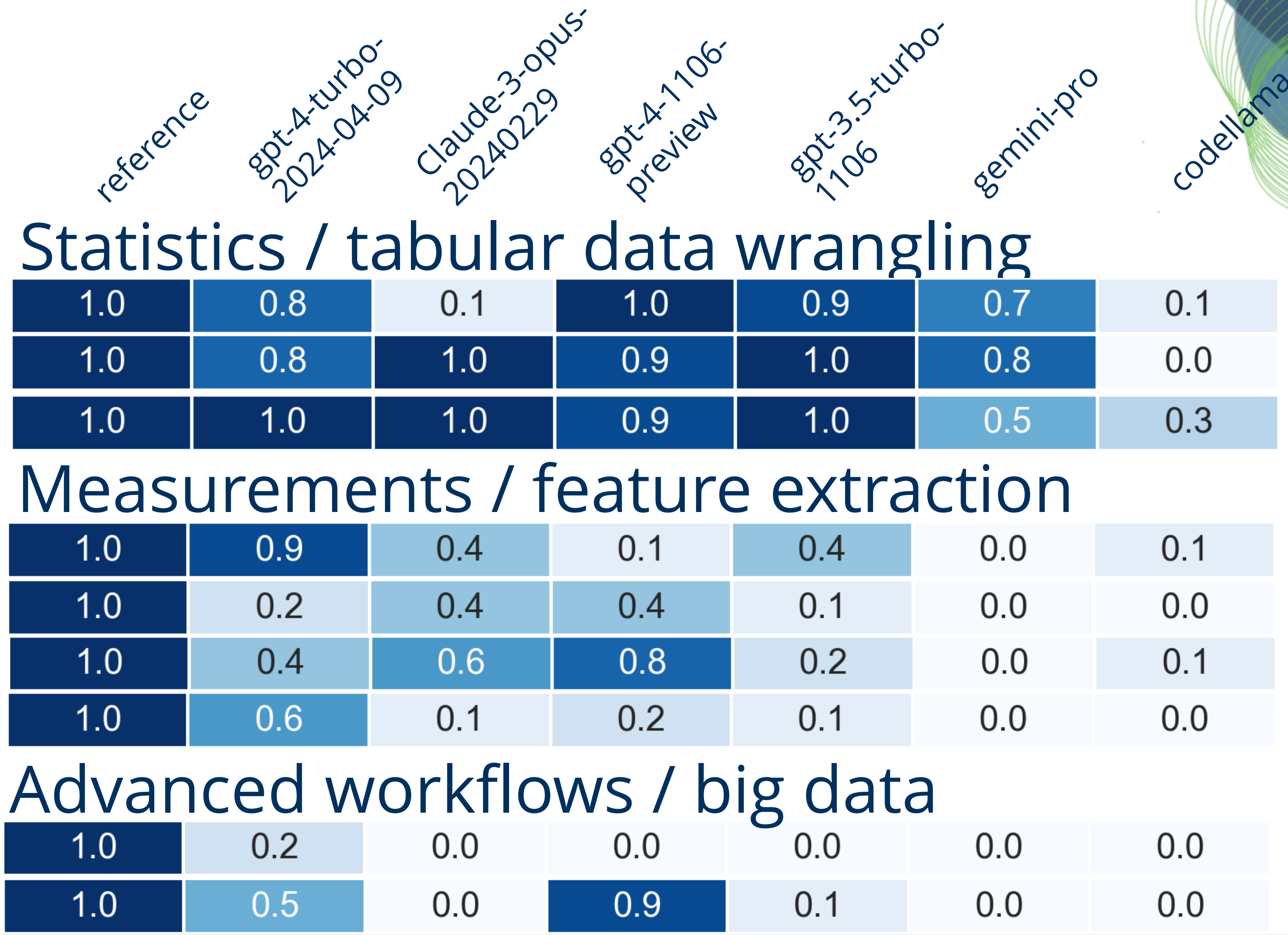


	reference	gpt-4-turbo-2024-04-09	Claude-3-opus-20240229	gpt-4-1106-preview	gpt-3.5-turbo-1106	gemini-pro	codellama
workflow_batch_process_folder_count_labels	1.0	0.1	0.0	0.3	0.0	0.0	0.0

https://www.biorxiv.org/content/10.1101/2024.04.19.590278v1
https://github.com/haesleinhuepf/human-eval-bia
Data Source: https://www.ebi.ac.uk/bioimage-archive/galleries/S-BIAD634-ai.html

Benchmarking LLMs for Bio-image Analysis

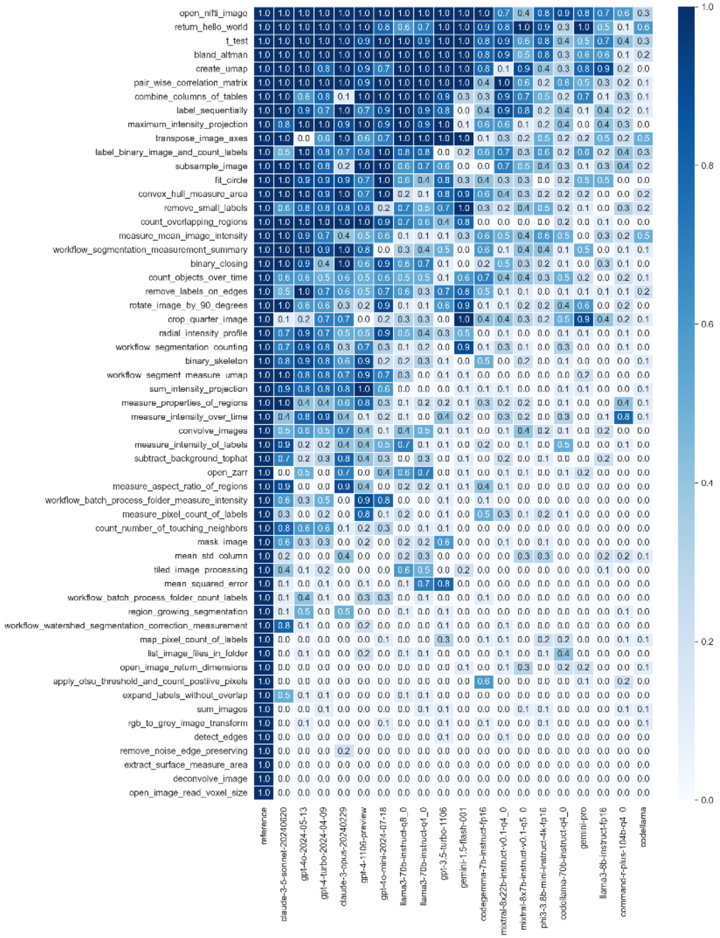
Unit-test pass-rate (n=10)



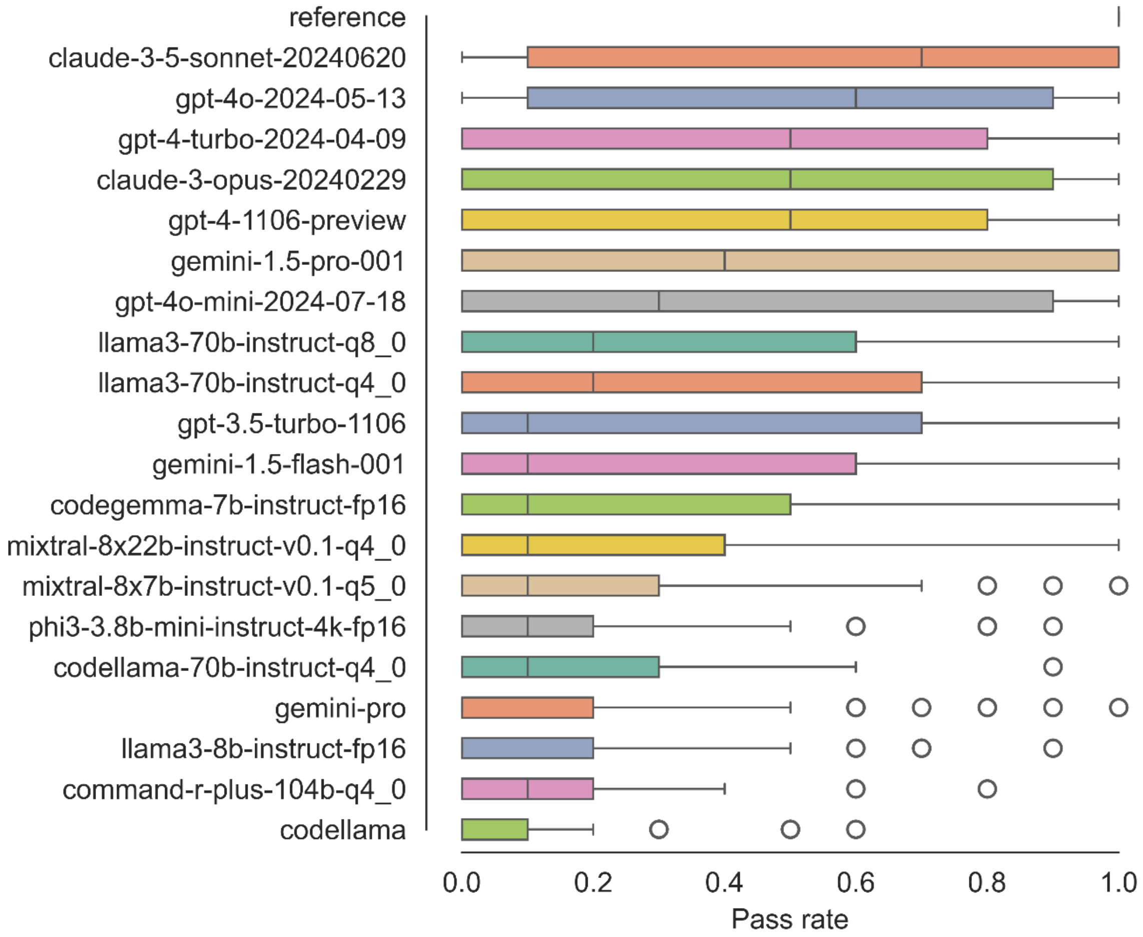
	reference	gpt-4-turbo-2024-04-09	Claude-3-opus-20240229	gpt-4-1106-preview	gpt-3.5-turbo-1106	gemini-pro	codellama
Statistics / tabular data wrangling							
combine_columns_of_tables	1.0	0.8	0.1	1.0	0.9	0.7	0.1
create_umap	1.0	0.8	1.0	0.9	1.0	0.8	0.0
t_test	1.0	1.0	1.0	0.9	1.0	0.5	0.3
Measurements / feature extraction							
measure_intensity_over_time	1.0	0.9	0.4	0.1	0.4	0.0	0.1
measure_intensity_of_labels	1.0	0.2	0.4	0.4	0.1	0.0	0.0
measure_properties_of_regions	1.0	0.4	0.6	0.8	0.2	0.0	0.1
count_number_of_touching_neighbors	1.0	0.6	0.1	0.2	0.1	0.0	0.0
Advanced workflows / big data							
tiled_image_processing	1.0	0.2	0.0	0.0	0.0	0.0	0.0
workflow_batch_process_folder_measure_intensity	1.0	0.5	0.0	0.9	0.1	0.0	0.0

Benchmarking LLMs for Bio-image Analysis

Summary: 57 use-cases (yet), 20 LLMs (yet), n=10



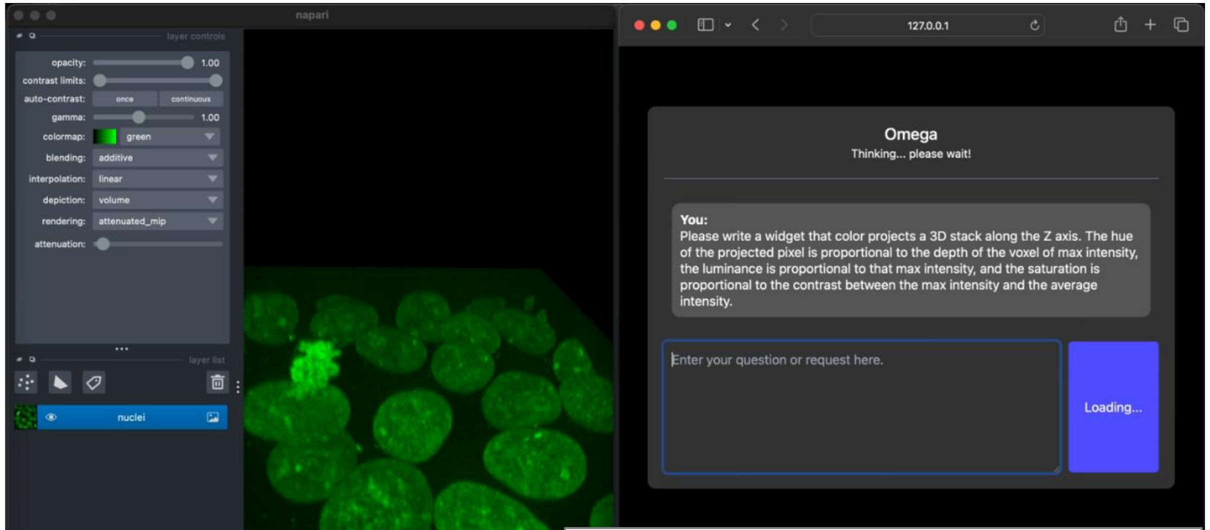
	reference	claude-3-5-sonnet-20240620	gpt-4o-2024-05-13	gpt-4-turbo-2024-04-09	claude-3-opus-20240229	gpt-4-1106-preview	gpt-4o-mini-2024-07-18	llama3-70b-instruct-q8_0	llama3-70b-instruct-q4_0	gpt-3.5-turbo-1106	gemini-1.5-flash-001	codegemma-7b-instruct-fp16	mixtral-8x22b-instruct-v0.1-q4_0	mixtral-8x7b-instruct-v0.1-q5_0	phi3-3.8b-mini-instruct-4k-fp16	codellama-70b-instruct-q4_0	gemini-pro	llama3-8b-instruct-fp16	command-r-plus-104b-q4_0	codellama
open_nifti_image	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.7	0.4	0.8	0.9	0.8	0.7	0.6	0.3
return_hello_world	1.0	1.0	1.0	1.0	1.0	1.0	0.8	0.6	0.7	1.0	1.0	0.9	0.8	1.0	0.9	0.3	1.0	0.5	0.1	0.6
t_test	1.0	1.0	1.0	1.0	1.0	0.9	1.0	1.0	0.9	1.0	1.0	0.8	0.9	0.6	0.8	0.4	0.5	0.7	0.4	0.3
bland_altman	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.8	0.9	0.5	0.8	0.3	0.6	0.6	0.1	0.2
create_umap	1.0	1.0	1.0	0.8	1.0	0.9	0.7	1.0	1.0	1.0	1.0	0.8	0.1	0.9	0.4	0.3	0.8	0.9	0.2	0.0
pair_wise_correlation_matrix	1.0	1.0	1.0	1.0	1.0	0.9	1.0	1.0	1.0	1.0	1.0	0.4	1.0	0.6	0.2	0.6	0.5	0.3	0.2	0.1
combine_columns_of_tables	1.0	1.0	0.6	0.8	0.1	1.0	1.0	1.0	1.0	0.9	0.3	0.3	0.9	0.7	0.5	0.2	0.7	0.1	0.3	0.1
label_sequentially	1.0	1.0	0.9	0.7	1.0	0.7	0.9	1.0	0.9	0.8	0.0	0.4	0.9	0.8	0.2	0.4	0.1	0.4	0.2	0.1
maximum_intensity_projection	1.0	0.8	1.0	1.0	0.9	1.0	0.9	1.0	0.9	1.0	0.1	0.6	0.6	0.1	0.2	0.4	0.0	0.4	0.3	0.0
transpose_image_axes	1.0	1.0	0.0	0.6	1.0	0.6	0.7	1.0	1.0	1.0	1.0	0.1	0.3	0.2	0.5	0.2	0.2	0.5	0.2	0.5
label_binary_image_and_count_labels	1.0	0.5	1.0	0.8	0.7	0.8	1.0	0.8	0.8	0.0	0.2	0.6	0.7	0.3	0.6	0.2	0.6	0.2	0.4	0.3
subsample_image	1.0	1.0	1.0	0.8	0.2	1.0	1.0	0.6	0.7	0.6	0.0	0.0	0.7	0.5	0.4	0.3	0.1	0.3	0.4	0.2
fit_circle	1.0	1.0	0.9	0.9	0.9	0.7	1.0	0.6	0.4	0.8	0.3	0.4	0.3	0.3	0.0	0.2	0.5	0.5	0.0	0.0
convex_hull_measure_area	1.0	1.0	1.0	0.9	1.0	0.7	1.0	0.2	0.1	0.8	0.9	0.6	0.4	0.3	0.2	0.2	0.2	0.0	0.0	0.2
remove_small_labels	1.0	0.6	0.8	0.8	0.8	0.8	0.2	0.7	0.5	0.7	1.0	0.3	0.2	0.4	0.5	0.2	0.1	0.0	0.3	0.2
count_overlapping_regions	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.7	0.6	0.4	0.8	0.0	0.0	0.0	0.0	0.2	0.0	0.1	0.0	0.0
measure_mean_image_intensity	1.0	1.0	0.9	0.7	0.4	0.5	0.6	0.1	0.1	0.1	0.3	0.6	0.5	0.4	0.6	0.5	0.0	0.3	0.2	0.5
workflow_segmentation_measurement_summary	1.0	1.0	1.0	0.9	1.0	0.8	0.0	0.3	0.4	0.5	0.0	0.6	0.1	0.4	0.4	0.1	0.5	0.0	0.1	0.1
binary_closing	1.0	1.0	0.9	0.4	1.0	0.6	0.9	0.6	0.7	0.1	0.0	0.2	0.5	0.3	0.2	0.1	0.0	0.3	0.1	0.0
count_objects_over_time	1.0	0.6	0.6	0.5	0.6	0.5	0.6	0.5	0.5	0.1	0.6	0.7	0.4	0.4	0.3	0.5	0.2	0.0	0.2	0.1
remove_labels_on_edges	1.0	0.5	1.0	0.7	0.6	0.5	0.7	0.6	0.3	0.7	0.8	0.5	0.1	0.1	0.0	0.1	0.1	0.1	0.1	0.2
rotate_image_by_90_degrees	1.0	1.0	0.6	0.6	0.3	0.2	0.9	0.1	0.1	0.6	0.9	0.1	0.1	0.2	0.2	0.4	0.6	0.0	0.2	0.0
crop_quarter_image	1.0	0.1	0.2	0.7	0.7	0.0	0.2	0.3	0.3	0.0	1.0	0.4	0.4	0.3	0.2	0.5	0.9	0.4	0.2	0.1
radial_intensity_profile	1.0	0.7	0.9	0.7	0.5	0.5	0.9	0.5	0.4	0.3	0.5	0.0	0.1	0.0	0.0	0.1	0.1	0.0	0.1	0.0
workflow_segmentation_counting	1.0	0.7	0.9	0.8	0.3	0.7	0.3	0.1	0.2	0.0	0.9	0.1	0.3	0.1	0.0	0.3	0.0	0.0	0.1	0.0
binary_skeleton	1.0	0.8	0.9	0.8	0.6	0.9	0.2	0.2	0.3	0.1	0.0	0.5	0.0	0.2	0.0	0.1	0.0	0.0	0.0	0.1
workflow_segment_measure_umap	1.0	1.0	0.8	0.8	0.7	0.9	0.7	0.3	0.0	0.1	0.1	0.0	0.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0
sum_intensity_projection	1.0	0.9	0.8	0.8	0.8	1.0	0.6	0.0	0.0	0.0	0.1	0.1	0.1	0.0	0.0	0.1	0.1	0.0	0.0	0.1
measure_properties_of_regions	1.0	1.0	0.4	0.4	0.6	0.8	0.3	0.1	0.2	0.2	0.1	0.3	0.2	0.2	0.0	0.1	0.0	0.0	0.4	0.1
measure_intensity_over_time	1.0	0.4	0.8	0.9	0.4	0.1	0.2	0.1	0.0	0.4	0.2	0.0	0.3	0.2	0.0	0.3	0.0	0.1	0.8	0.1
convolve_images	1.0	0.5	0.6	0.5	0.7	0.4	0.1	0.4	0.5	0.1	0.1	0.0	0.1	0.4	0.2	0.1	0.0	0.2	0.0	0.0
measure_intensity_of_labels	1.0	0.9	0.2	0.2	0.4	0.4	0.5	0.7	0.1	0.1	0.0	0.2	0.0	0.1	0.0	0.5	0.0	0.0	0.1	0.0
subtract_background_tophat	1.0	0.7	0.2	0.3	0.8	0.4	0.3	0.0	0.3	0.0	0.1	0.0	0.2	0.0	0.1	0.0	0.0	0.2	0.0	0.0
open_zarr	1.0	0.0	0.5	0.0	0.7	0.0	0.4	0.6	0.7	0.0	0.1	0.0	0.1	0.1	0.0	0.1	0.2	0.0	0.0	0.0
measure_aspect_ratio_of_regions	1.0	0.9	0.0	0.0	0.9	0.4	0.0	0.2	0.2	0.1	0.1	0.4	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
workflow_batch_process_folder_measure_intensity	1.0	0.6	0.3	0.5	0.0	0.9	0.8	0.0	0.0	0.1	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
measure_pixel_count_of_labels	1.0	0.3	0.0	0.2	0.0	0.8	0.1	0.2	0.0	0.1	0.0	0.5	0.3	0.1	0.2	0.1	0.0	0.0	0.0	0.0
count_number_of_touching_neighbors	1.0	0.8	0.6	0.6	0.1	0.2	0.3	0.0	0.1	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
mask_image	1.0	0.6	0.3	0.3	0.0	0.2	0.0	0.2	0.2	0.6	0.0	0.0	0.1	0.0	0.1	0.0	0.0	0.0	0.0	0.0
mean_std_column	1.0	0.2	0.0	0.0	0.4	0.0	0.0	0.2	0.3	0.0	0.0	0.0	0.0	0.3	0.3	0.0	0.0	0.2	0.2	0.1
tiled_image_processing	1.0	0.4	0.1	0.2	0.0	0.0	0.0	0.6	0.5	0.0	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0
mean_squared_error	1.0	0.1	0.0	0.1	0.0	0.1	0.0	0.1	0.7	0.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
workflow_batch_process_folder_count_labels	1.0	0.1	0.4	0.1	0.0	0.3	0.3	0.0	0.1	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
region_growing_segmentation	1.0	0.1	0.5	0.0	0.5	0.0	0.0	0.1	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0
workflow_watershed_segmentation_correction_measurement	1.0	0.8	0.1	0.0	0.0	0.2	0.0	0.0	0.0	0.1	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
map_pixel_count_of_labels	1.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.3	0.0	0.1	0.1	0.0	0.2	0.2	0.0	0.0	0.1	0.1
list_image_files_in_folder	1.0	0.0	0.1	0.0	0.0	0.2	0.0	0.1	0.1	0.0	0.0	0.0	0.1	0.0	0.1	0.4	0.0	0.0	0.0	0.0
open_image_return_dimensions	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.1	0.3	0.0	0.2	0.2	0.0	0.0	0.1
apply_otsu_threshold_and_count_postiive_pixels	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.6	0.0	0.0	0.0	0.0	0.1	0.0	0.2	0.0
expand_labels_without_overlap	1.0	0.5	0.1	0.1	0.0	0.0	0.0	0.1	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
sum_images	1.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.1	0.1	0.0	0.0	0.0	0.1	0.1	0.0	0.0	0.0	0.1	0.1
rgb_to_grey_image_transform	1.0	0.0	0.1	0.0	0.0	0.0	0.1	0.0	0.0	0.1	0.0	0.1	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.1
detect_edges	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
remove_noise_edge_preserving	1.0	0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
extract_surface_measure_area	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
deconvolve_image	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
open_image_read_voxel_size	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0



https://www.biorxiv.org/content/10.1101/2024.04.19.590278v3
https://github.com/haesleinhuepf/human-eval-bia/

LLMs are everywhere

Napari-chatGPT / Omega



opacity: 1.00
contrast limits:
auto-contrast: once continuous
gamma: 1.00
colormap: green
blending: additive
interpolation: linear
depiction: volume
rendering: attenuated_mip
attenuation:

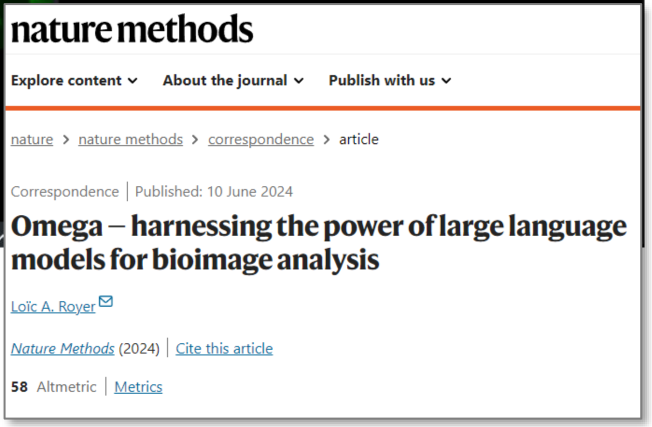
nuclei

Omega
Thinking... please wait!

You:
Please write a widget that color projects a 3D stack along the Z axis. The hue of the projected pixel is proportional to the depth of the voxel of max intensity, the luminance is proportional to that max intensity, and the saturation is proportional to the contrast between the max intensity and the average intensity.

Enter your question or request here.

Loading...



nature › nature methods › correspondence › article

Correspondence | Published: 10 June 2024

Omega — harnessing the power of large language models for bioimage analysis

Loïc A. Royer

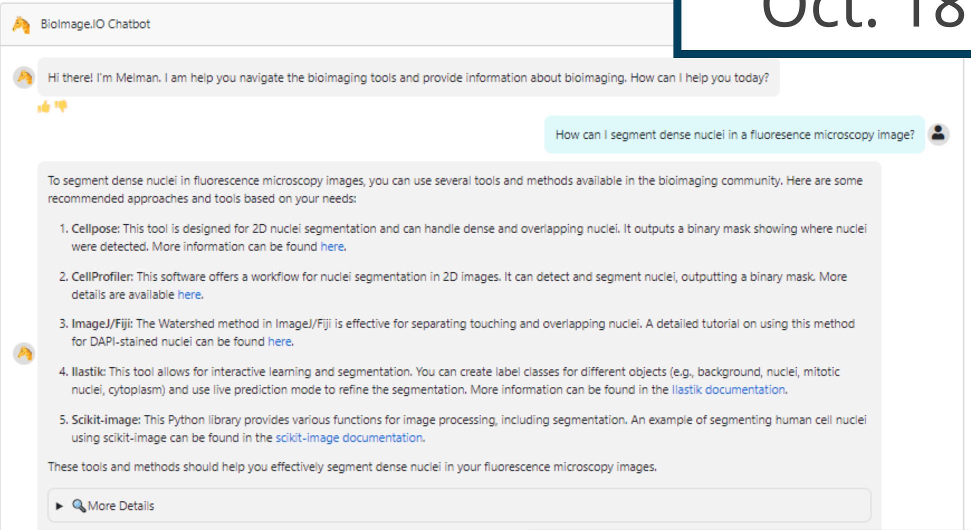
Nature Methods (2024) | Cite this article

58 Altmetric | Metrics

https://github.com/royerlab/napari-chatgpt
https://www.nature.com/articles/s41592-024-02310-w

Bioimage-io ChatBot

GloBIAS Seminar Series
Sept. 27th 2024
Oct. 18th 2024



Hi there! I'm Melman. I am help you navigate the bioimaging tools and provide information about bioimaging. How can I help you today?

How can I segment dense nuclei in a fluoresence microscopy image?

To segment dense nuclei in fluorescence microscopy images, you can use several tools and methods available in the bioimaging community. Here are some recommended approaches and tools based on your needs:

1. **Cellpose:** This tool is designed for 2D nuclei segmentation and can handle dense and overlapping nuclei. It outputs a binary mask showing where nuclei were detected. More information can be found here.
2. **CellProfiler:** This software offers a workflow for nuclei segmentation in 2D images. It can detect and segment nuclei, outputting a binary mask. More details are available here.
3. **ImageJ/Fiji:** The Watershed method in ImageJ/Fiji is effective for separating touching and overlapping nuclei. A detailed tutorial on using this method for DAPI-stained nuclei can be found here.
4. **Ilastik:** This tool allows for interactive learning and segmentation. You can create label classes for different objects (e.g., background, nuclei, mitotic nuclei, cytoplasm) and use live prediction mode to refine the segmentation. More information can be found in the Ilastik documentation.
5. **Scikit-image:** This Python library provides various functions for image processing, including segmentation. An example of segmenting human cell nuclei using scikit-image can be found in the scikit-image documentation.

These tools and methods should help you effectively segment dense nuclei in your fluorescence microscopy images.

▸ More Details

Ready to chat! Type your message and press enter!

@

Send | Edit Profile | Reset | Feedback | ••• Options

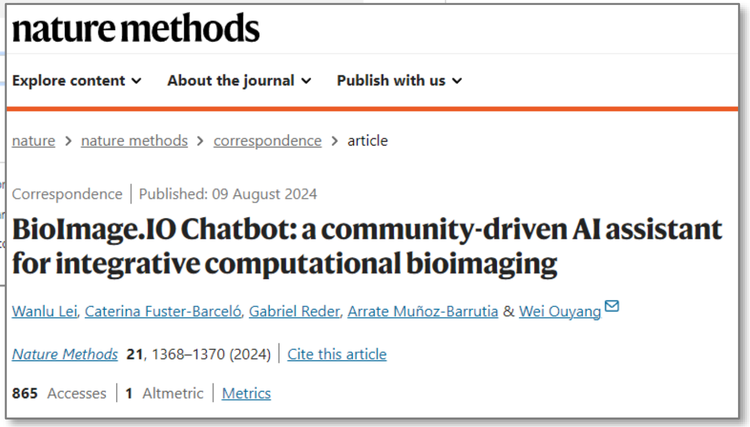
Remaining Message Quota: 19 (reset hourly)

Warning: BioImage.IO Chatbot can make mistakes. Co

Tips: Customize your profile to get personalized a

If you find our chatbot helpful for you, please c

AI4Life Icon



nature › nature methods › correspondence › article

Correspondence | Published: 09 August 2024

BioImage.IO Chatbot: a community-driven AI assistant for integrative computational bioimaging

Wanlu Lei, Caterina Fuster-Barceló, Gabriel Reder, Arrate Muñoz-Barrutia & Wei Ouyang

Nature Methods **21**, 1368–1370 (2024) | Cite this article

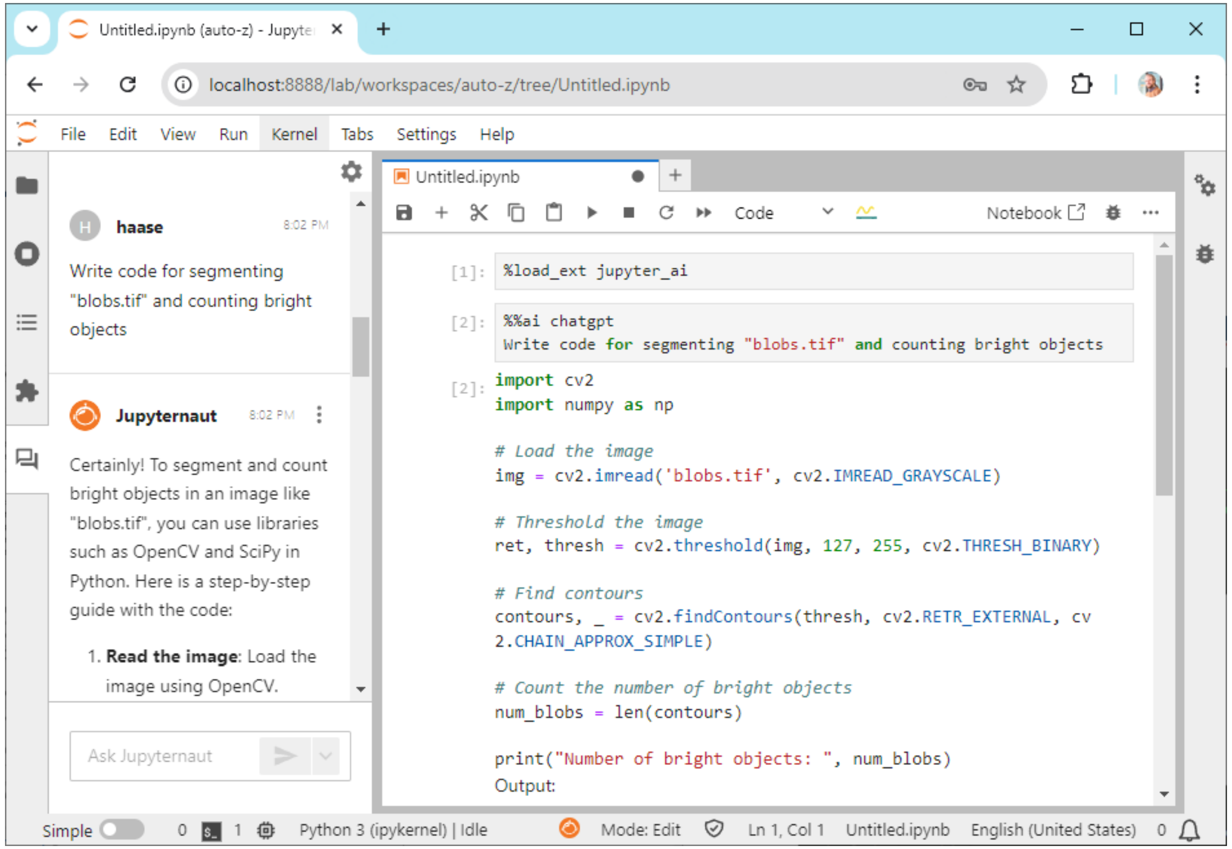
865 Accesses | **1** Altmetric | Metrics

https://bioimage.io/chat/
https://www.nature.com/articles/s41592-024-02370-y

https://www.globias.org/activities/bia-seminar-series

LLMs are everywhere

Jupyter-AI


Untitled.ipynb (auto-z) - Jupyter
localhost:8888/lab/workspaces/auto-z/tree/Untitled.ipynb
File Edit View Run Kernel Tabs Settings Help
Untitled.ipynb
Code
Notebook
Ask Jupyternaut
Simple 0 1 Python 3 (ipykernel) | Idle Mode: Edit Ln 1, Col 1 Untitled.ipynb English (United States) 0

haase 8:02 PM

Write code for segmenting "blobs.tif" and counting bright objects

Jupyternaut 8:02 PM

Certainly! To segment and count bright objects in an image like "blobs.tif", you can use libraries such as OpenCV and SciPy in Python. Here is a step-by-step guide with the code:

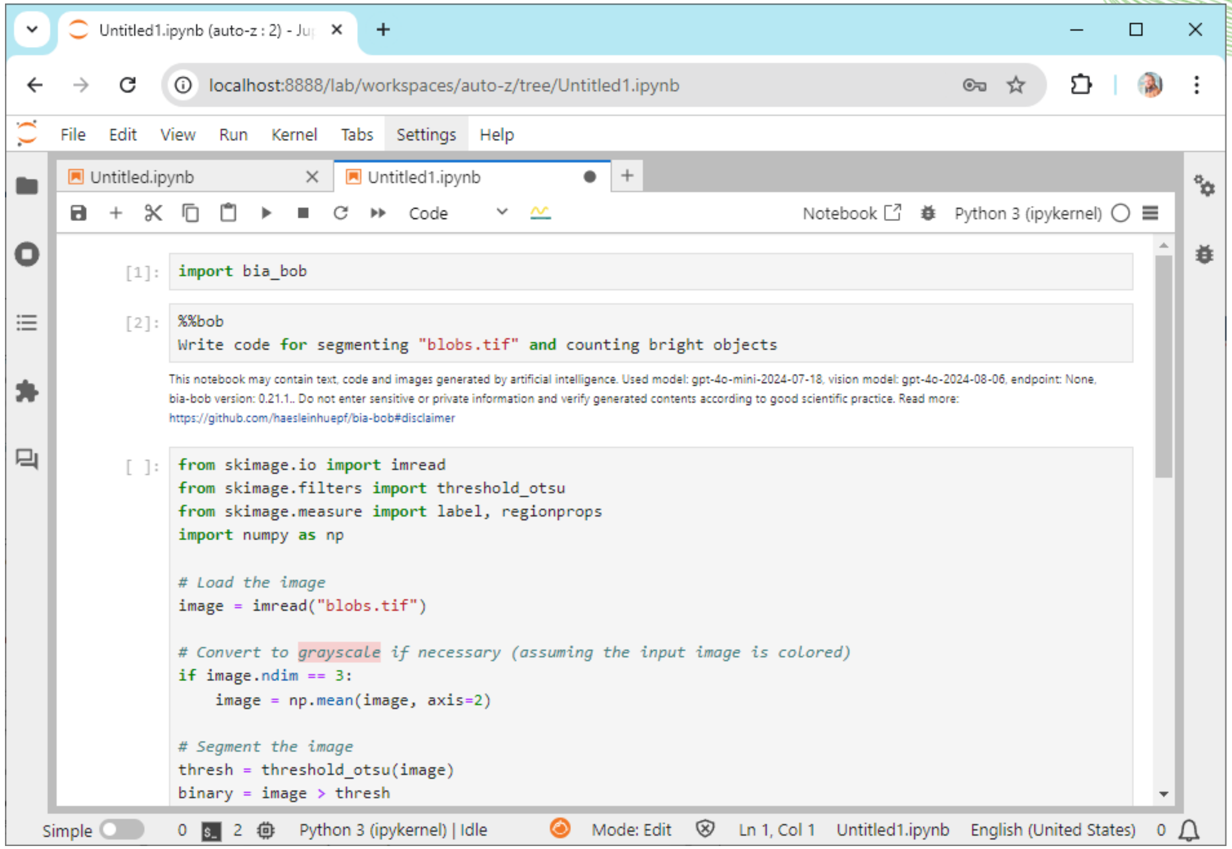
1. **Read the image**: Load the image using OpenCV.

```
[1]: %load_ext jupyter_ai
```

```
[2]: %%ai chatgpt
     Write code for segmenting "blobs.tif" and counting bright objects
```

```
[2]: import cv2
     import numpy as np

     # Load the image
     img = cv2.imread('blobs.tif', cv2.IMREAD_GRAYSCALE)

     # Threshold the image
     ret, thresh = cv2.threshold(img, 127, 255, cv2.THRESH_BINARY)

     # Find contours
     contours, _ = cv2.findContours(thresh, cv2.RETR_EXTERNAL, cv
     2.CHAIN_APPROX_SIMPLE)

     # Count the number of bright objects
     num_blobs = len(contours)

     print("Number of bright objects: ", num_blobs)
     Output:
```

bia-bob


Untitled1.ipynb (auto-z : 2) - Ju
localhost:8888/lab/workspaces/auto-z/tree/Untitled1.ipynb
File Edit View Run Kernel Tabs Settings Help
Untitled.ipynb
Untitled1.ipynb
Code
Notebook
Python 3 (ipykernel)
Simple 0 2 Python 3 (ipykernel) | Idle Mode: Edit Ln 1, Col 1 Untitled1.ipynb English (United States) 0

```
[1]: import bia_bob
```

```
[2]: %%bob
     Write code for segmenting "blobs.tif" and counting bright objects
```

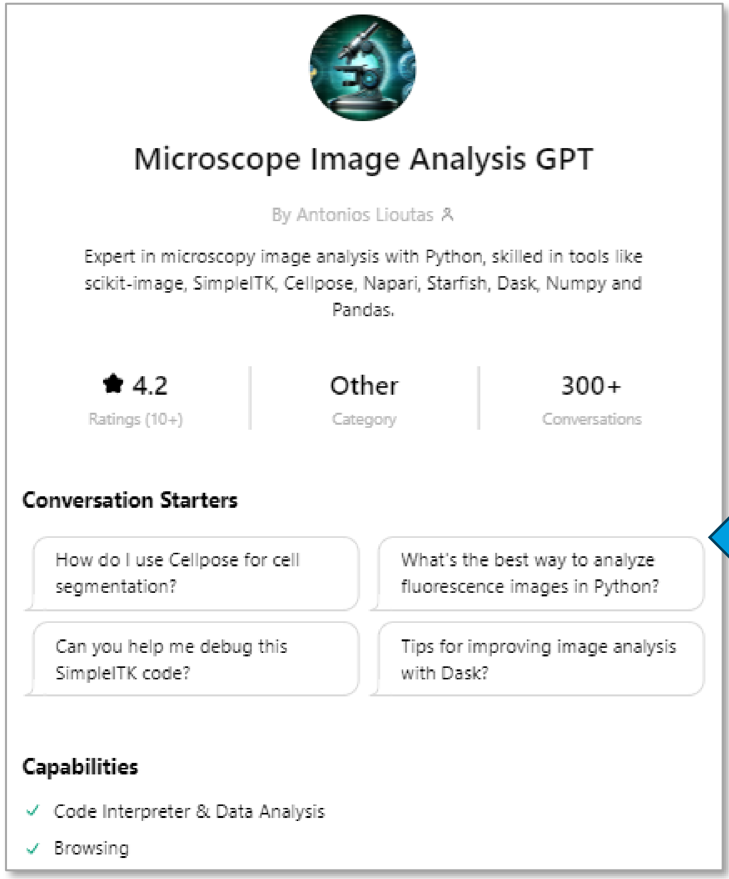
This notebook may contain text, code and images generated by artificial intelligence. Used model: gpt-4o-mini-2024-07-18, vision model: gpt-4o-2024-08-06, endpoint: None, bia-bob version: 0.21.1.. Do not enter sensitive or private information and verify generated contents according to good scientific practice. Read more: https://github.com/haesleinhuepf/bia-bob#disclaimer

```
[ ]: from skimage.io import imread
     from skimage.filters import threshold_otsu
     from skimage.measure import label, regionprops
     import numpy as np

     # Load the image
     image = imread("blobs.tif")

     # Convert to grayscale if necessary (assuming the input image is colored)
     if image.ndim == 3:
         image = np.mean(image, axis=2)

     # Segment the image
     thresh = threshold_otsu(image)
     binary = image > thresh
```

https://jupyter-ai.readthedocs.io/

https://github.com/haesleinhuepf/bia-bob

LLMs are everywhere



Microscope Image Analysis GPT

By Antonios Lioutas

Expert in microscopy image analysis with Python, skilled in tools like scikit-image, SimpleITK, Cellpose, Napari, Starfish, Dask, Numpy and Pandas.

★ 4.2	Other	300+
Ratings (10+)	Category	Conversations

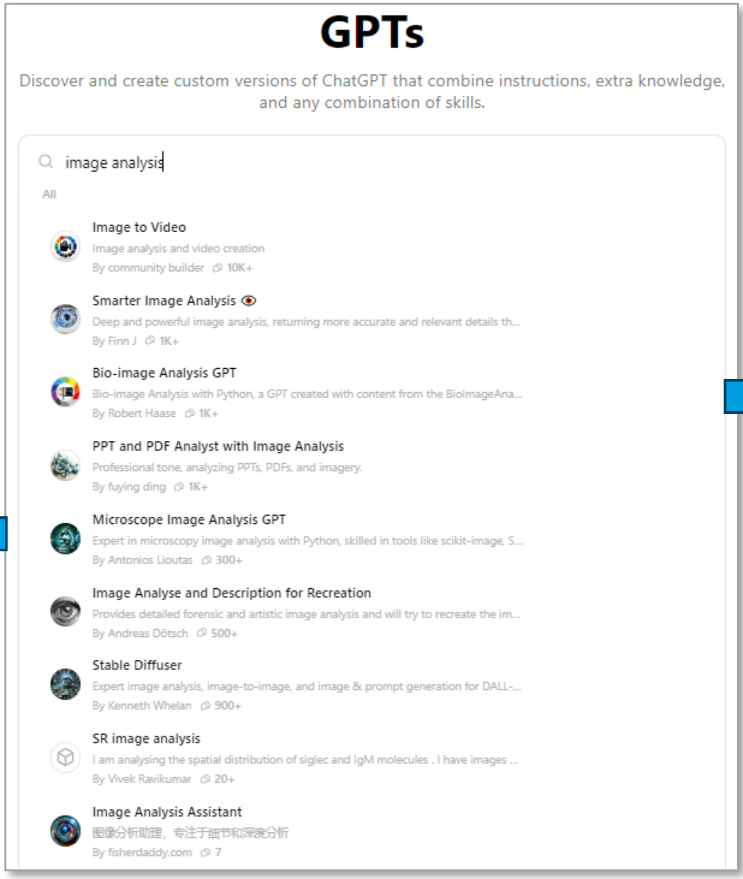
Conversation Starters

- How do I use Cellpose for cell segmentation?
- What's the best way to analyze fluorescence images in Python?
- Can you help me debug this SimpleITK code?
- Tips for improving image analysis with Dask?

Capabilities

- ✓ Code Interpreter & Data Analysis
- ✓ Browsing

https://chatgpt.com/g/g-FGdNx7MIl-microscope-image-analysis-gpt



GPTs

Discover and create custom versions of ChatGPT that combine instructions, extra knowledge, and any combination of skills.

image analysis

All

Image to Video
Image analysis and video creation
By community builder 10K+

Smarter Image Analysis
Deep and powerful image analysis, returning more accurate and relevant details th...
By Finn J 1K+

Bio-image Analysis GPT
Bio-image Analysis with Python, a GPT created with content from the BioImageAna...
By Robert Haase 1K+

PPT and PDF Analyst with Image Analysis
Professional tone, analyzing PPTs, PDFs, and imagery.
By fuying ding 1K+

Microscope Image Analysis GPT
Expert in microscopy image analysis with Python, skilled in tools like scikit-image, S...
By Antonios Lioutas 300+

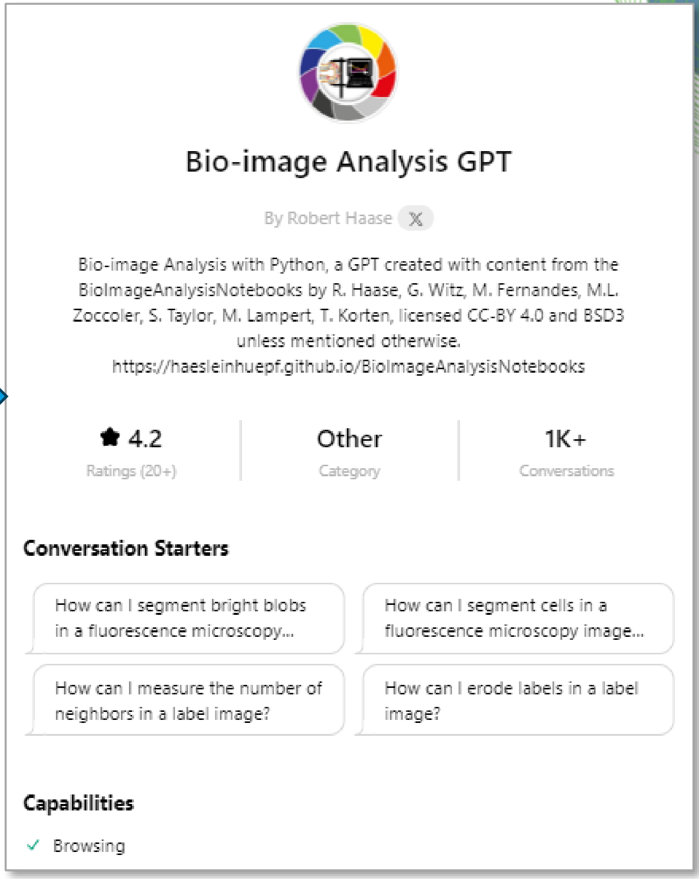
Image Analyse and Description for Recreation
Provides detailed forensic and artistic image analysis and will try to recreate the im...
By Andreas Dötsch 500+

Stable Diffuser
Expert image analysis, image-to-image, and image & prompt generation for DALL-...
By Kenneth Whelan 900+

SR image analysis
I am analysing the spatial distribution of siglec and IgM molecules . I have images ...
By Vivek Ravikumar 20+

Image Analysis Assistant
图像分析助理，专注于细节和深度分析
By fisherdaddy.com 7

http://chat.openai.com/gpts



Bio-image Analysis GPT

By Robert Haase

Bio-image Analysis with Python, a GPT created with content from the BioImageAnalysisNotebooks by R. Haase, G. Witz, M. Fernandes, M.L. Zoccoler, S. Taylor, M. Lampert, T. Korten, licensed CC-BY 4.0 and BSD3 unless mentioned otherwise. https://haesleinhuepf.github.io/BioImageAnalysisNotebooks

★ 4.2	Other	1K+
Ratings (20+)	Category	Conversations

Conversation Starters

- How can I segment bright blobs in a fluorescence microscopy...
- How can I segment cells in a fluorescence microscopy image...
- How can I measure the number of neighbors in a label image?
- How can I erode labels in a label image?

Capabilities

- ✓ Browsing

https://chatgpt.com/g/g-psAohb1OY-bio-image-analysis-gpt

ScaDS.AI
DRESDEN LEIPZIG
CENTER FOR SCALABLE DATA ANALYTICS
AND ARTIFICIAL INTELLIGENCE

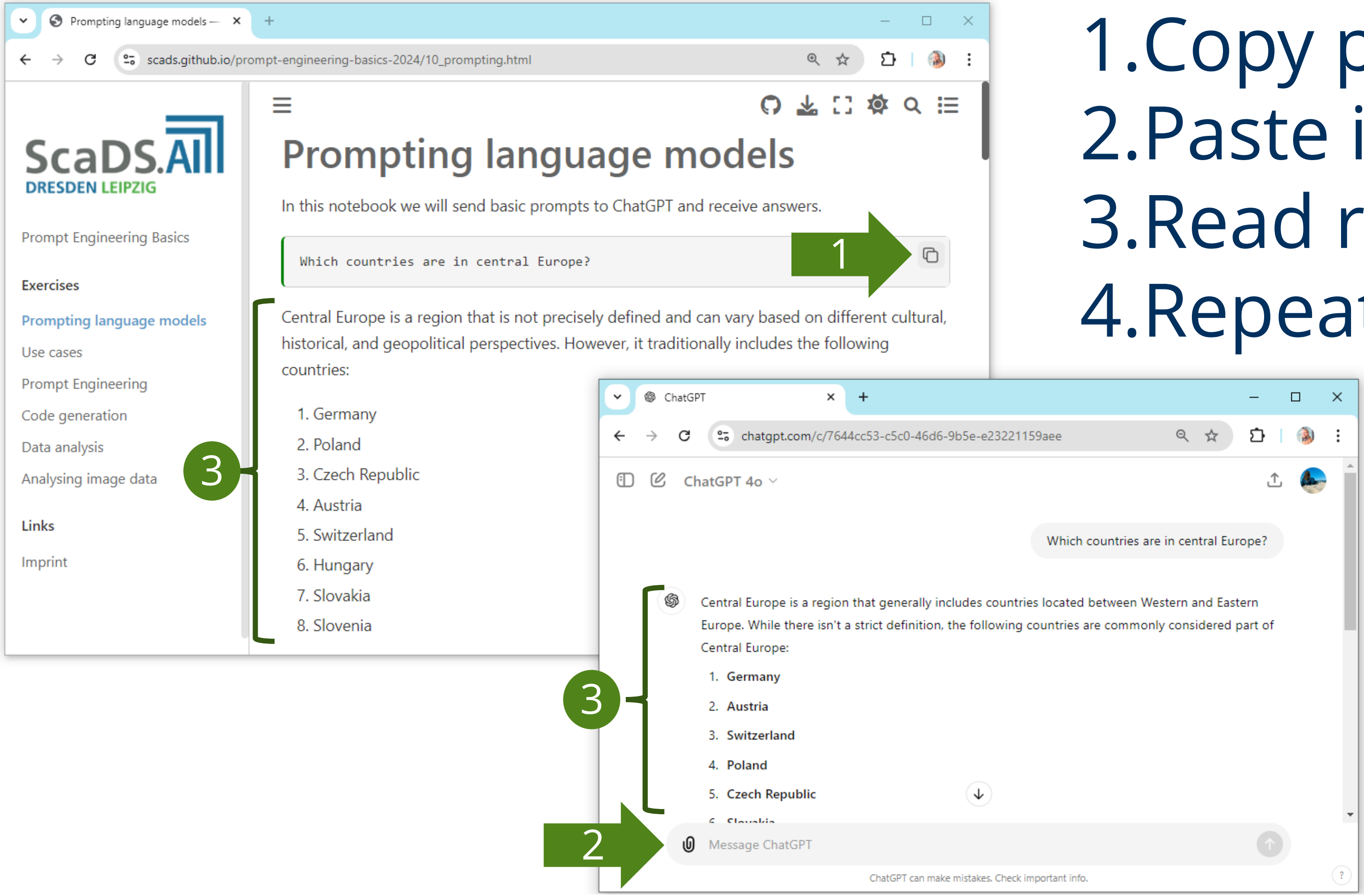
Exercises

Robert Haase

https://doi.org/10.5281/zenodo.13336075

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.

Exercises


Prompting language models –
scads.github.io/prompt-engineering-basics-2024/10_prompting.html
ScaDS.AI
DRESDEN LEIPZIG
Prompt Engineering Basics
Exercises
Prompting language models
Use cases
Prompt Engineering
Code generation
Data analysis
Analysing image data
Links
Imprint
1
3
ChatGPT
chatgpt.com/c/7644cc53-c5c0-46d6-9b5e-e23221159aee
ChatGPT 4o
3
2
Message ChatGPT
ChatGPT can make mistakes. Check important info.

Prompting language models

In this notebook we will send basic prompts to ChatGPT and receive answers.

```
Which countries are in central Europe?
```

Central Europe is a region that is not precisely defined and can vary based on different cultural, historical, and geopolitical perspectives. However, it traditionally includes the following countries:

1. Germany
2. Poland
3. Czech Republic
4. Austria
5. Switzerland
6. Hungary
7. Slovakia
8. Slovenia

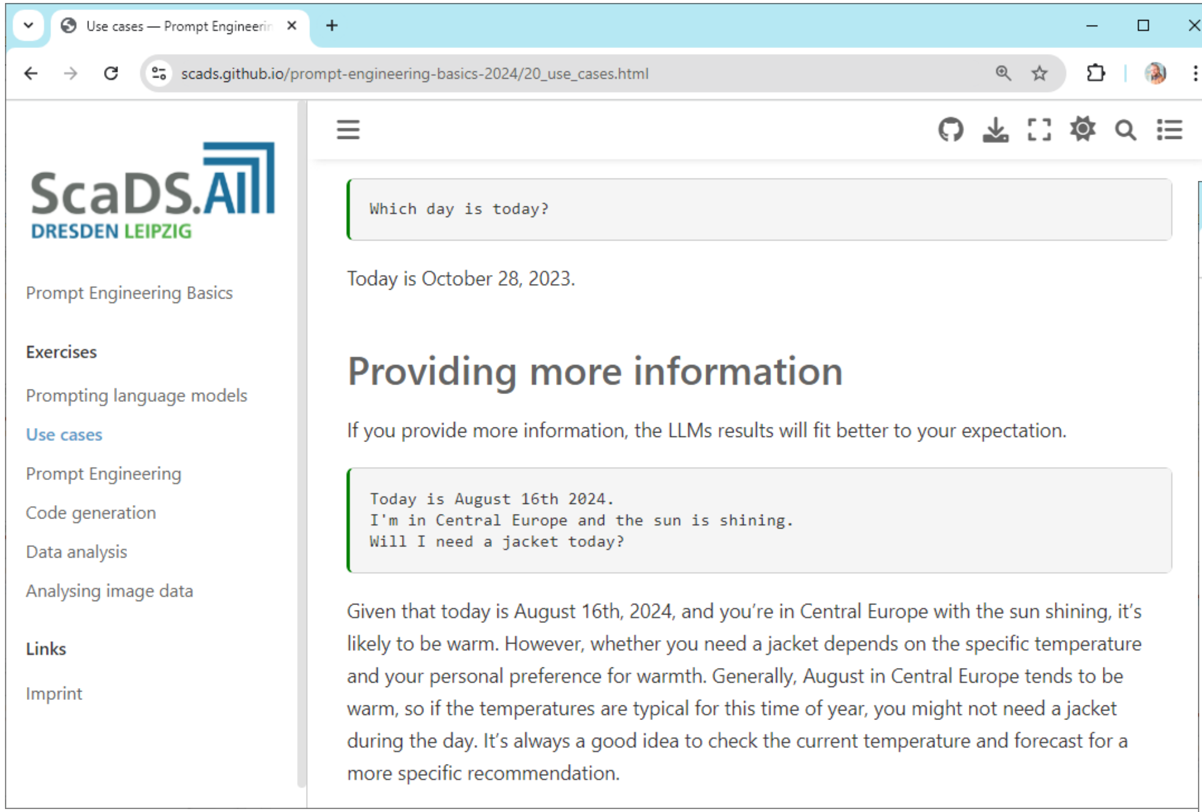
Which countries are in central Europe?

Central Europe is a region that generally includes countries located between Western and Eastern Europe. While there isn't a strict definition, the following countries are commonly considered part of Central Europe:

1. **Germany**
2. **Austria**
3. **Switzerland**
4. **Poland**
5. **Czech Republic**

1. Copy prompt
2. Paste it into ChatGPT
3. Read response
4. Repeat ;-)

Exercises: Prompting basics (optional)


Use cases — Prompt Engineerin
scads.github.io/prompt-engineering-basics-2024/20_use_cases.html
ScaDS.AI DRESDEN LEIPZIG
Prompt Engineering Basics
Exercises
Prompting language models
Use cases
Prompt Engineering
Code generation
Data analysis
Analysing image data
Links
Imprint

```
Which day is today?
```

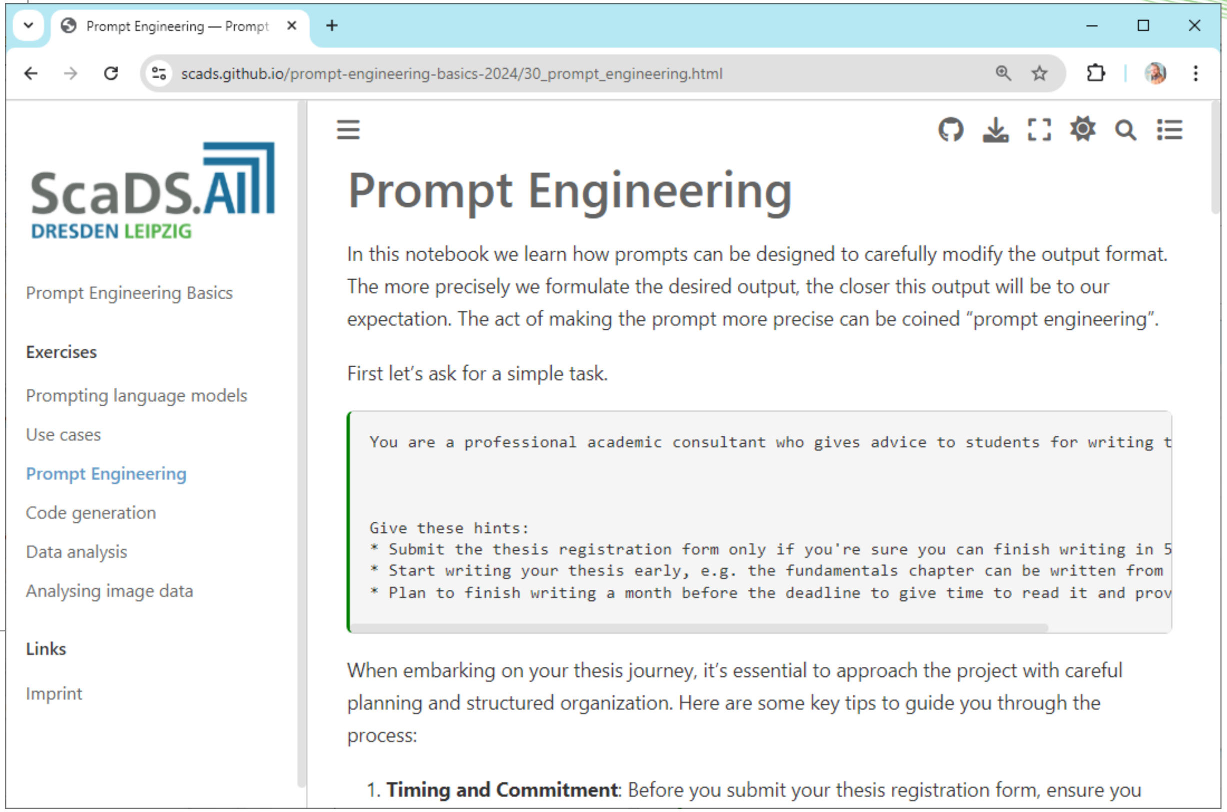
Today is October 28, 2023.

Providing more information

If you provide more information, the LLMs results will fit better to your expectation.

```
Today is August 16th 2024.
I'm in Central Europe and the sun is shining.
Will I need a jacket today?
```

Given that today is August 16th, 2024, and you're in Central Europe with the sun shining, it's likely to be warm. However, whether you need a jacket depends on the specific temperature and your personal preference for warmth. Generally, August in Central Europe tends to be warm, so if the temperatures are typical for this time of year, you might not need a jacket during the day. It's always a good idea to check the current temperature and forecast for a more specific recommendation.


Prompt Engineering — Prompt
scads.github.io/prompt-engineering-basics-2024/30_prompt_engineering.html
ScaDS.AI DRESDEN LEIPZIG
Prompt Engineering Basics
Exercises
Prompting language models
Use cases
Prompt Engineering
Code generation
Data analysis
Analysing image data
Links
Imprint

Prompt Engineering

In this notebook we learn how prompts can be designed to carefully modify the output format. The more precisely we formulate the desired output, the closer this output will be to our expectation. The act of making the prompt more precise can be coined "prompt engineering".

First let's ask for a simple task.

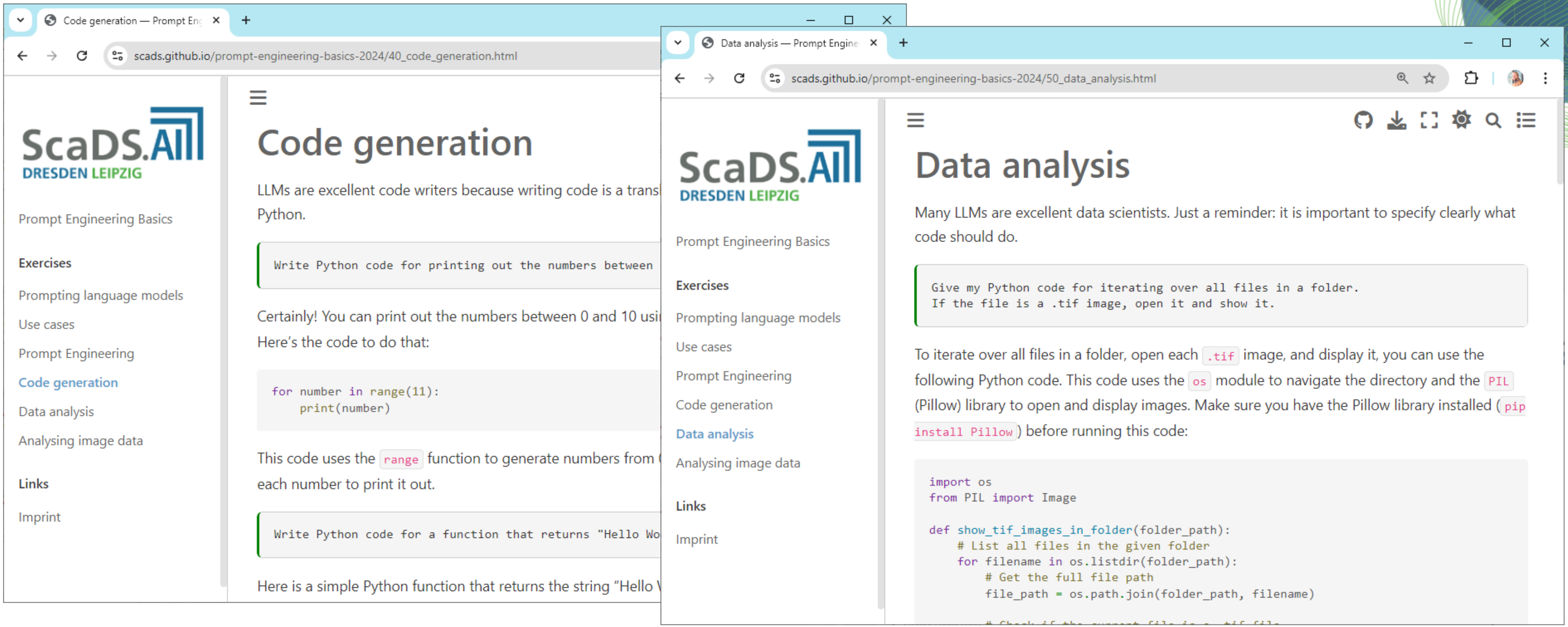
```
You are a professional academic consultant who gives advice to students for writing t


Give these hints:
* Submit the thesis registration form only if you're sure you can finish writing in 5
* Start writing your thesis early, e.g. the fundamentals chapter can be written from
* Plan to finish writing a month before the deadline to give time to read it and prov
```

When embarking on your thesis journey, it's essential to approach the project with careful planning and structured organization. Here are some key tips to guide you through the process:

1. **Timing and Commitment**: Before you submit your thesis registration form, ensure you

https://scads.github.io/prompt-engineering-basics-2024/20_use_cases.html
https://scads.github.io/prompt-engineering-basics-2024/30_prompt_engineering.html

Exercises: Coding + [Image] Data Analysis


Code generation — Prompt En
scads.github.io/prompt-engineering-basics-2024/40_code_generation.html
ScaDS.AI DRESDEN LEIPZIG
Prompt Engineering Basics
Exercises
Prompting language models
Use cases
Prompt Engineering
Code generation
Data analysis
Analysing image data
Links
Imprint
Data analysis — Prompt Engine
scads.github.io/prompt-engineering-basics-2024/50_data_analysis.html

Code generation

LLMs are excellent code writers because writing code is a trans

Python.

```
Write Python code for printing out the numbers between
```

Certainly! You can print out the numbers between 0 and 10 usi

Here's the code to do that:

```
for number in range(11):
    print(number)
```

This code uses the `range` function to generate numbers from

each number to print it out.

```
Write Python code for a function that returns "Hello Wo
```

Here is a simple Python function that returns the string "Hello

Data analysis

Many LLMs are excellent data scientists. Just a reminder: it is important to specify clearly what code should do.

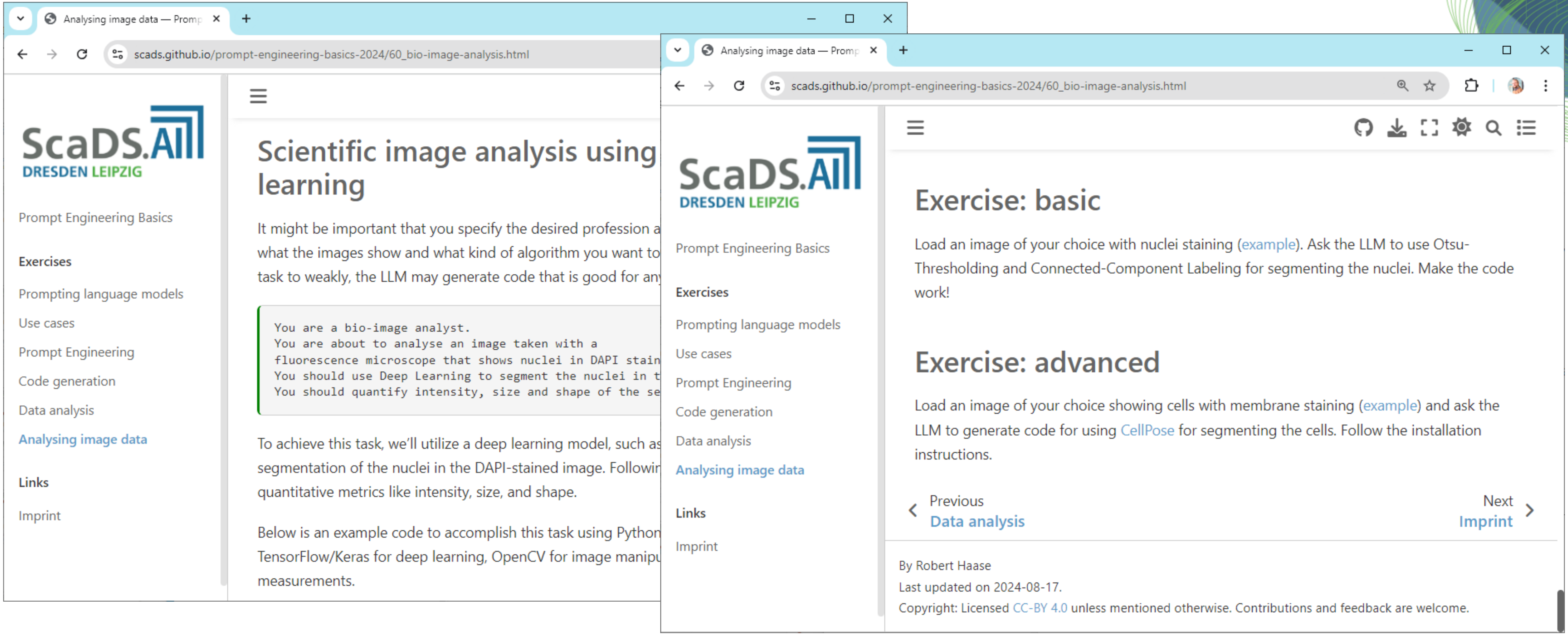
```
Give my Python code for iterating over all files in a folder.
If the file is a .tif image, open it and show it.
```

To iterate over all files in a folder, open each `.tif` image, and display it, you can use the following Python code. This code uses the `os` module to navigate the directory and the `PIL` (Pillow) library to open and display images. Make sure you have the Pillow library installed (`pip install Pillow`) before running this code:

```
import os
from PIL import Image

def show_tif_images_in_folder(folder_path):
    # List all files in the given folder
    for filename in os.listdir(folder_path):
        # Get the full file path
        file_path = os.path.join(folder_path, filename)
```

https://scads.github.io/prompt-engineering-basics-2024/40_code_generation.html
https://scads.github.io/prompt-engineering-basics-2024/50_data_analysis.html

Exercises: Bio-image Analysis


Analysing image data — Promp
scads.github.io/prompt-engineering-basics-2024/60_bio-image-analysis.html
ScaDS.AI
DRESDEN LEIPZIG
Prompt Engineering Basics
Exercises
Prompting language models
Use cases
Prompt Engineering
Code generation
Data analysis
Analysing image data
Links
Imprint

Scientific image analysis using learning

It might be important that you specify the desired profession a what the images show and what kind of algorithm you want to task to weakly, the LLM may generate code that is good for an

```
You are a bio-image analyst.
You are about to analyse an image taken with a
fluorescence microscope that shows nuclei in DAPI stain
You should use Deep Learning to segment the nuclei in t
You should quantify intensity, size and shape of the se
```

To achieve this task, we'll utilize a deep learning model, such as segmentation of the nuclei in the DAPI-stained image. Followin quantitative metrics like intensity, size, and shape.

Below is an example code to accomplish this task using Python TensorFlow/Keras for deep learning, OpenCV for image manipu measurements.

Exercise: basic

Load an image of your choice with nuclei staining (example). Ask the LLM to use Otsu-Thresholding and Connected-Component Labeling for segmenting the nuclei. Make the code work!

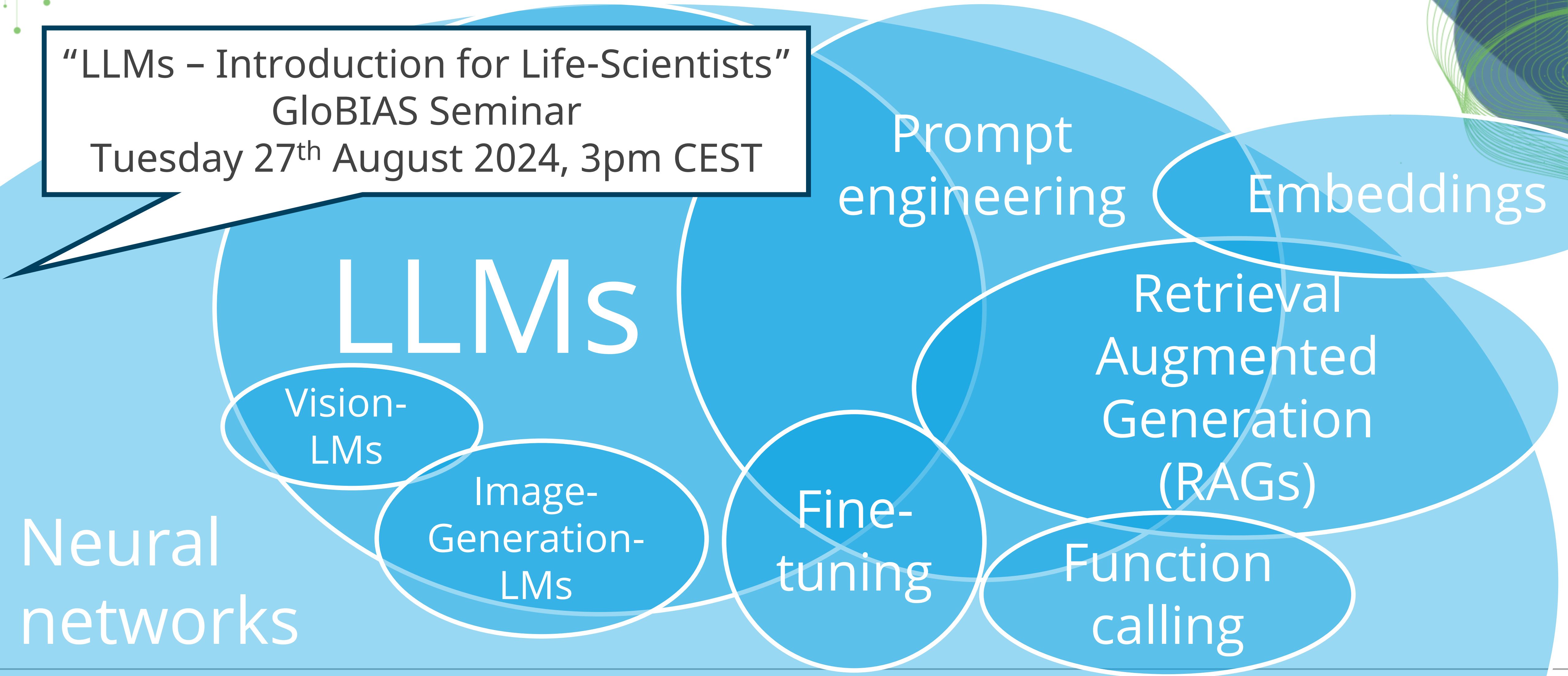
Exercise: advanced

Load an image of your choice showing cells with membrane staining (example) and ask the LLM to generate code for using CellPose for segmenting the cells. Follow the installation instructions.

By Robert Haase
Last updated on 2024-08-17.
Copyright: Licensed CC-BY 4.0 unless mentioned otherwise. Contributions and feedback are welcome.

Outlook

“LLMs – Introduction for Life-Scientists”
GloBIAS Seminar
Tuesday 27th August 2024, 3pm CEST



https://www.globias.org/activities/bia-seminar-series#h.l3k3h33clzzc

Summary & outlook

- LLMs can generate code to analyze biological microscopy images
- Open-source benchmarks can help targeting further improvement
- Challenges:
  - Identify best strategy (prompt-engineering)
  - Introduce good scientific practice -> trustworthy AI

https://doi.org/10.5281/zenodo.13336075

Acknowledgements

Christian Tischer
(EMBL Heidelberg)

Nico Scherf
(MPI CBS Leipzig)

Jean-Karim Hériché
(EMBL Heidelberg)

Kevin Yamauchi
(ETH Zurich / Basel)

Loic Royer
(CZ Biohub San Francisco)

Matthias Täschner
(ScaDS.AI)

PoL
Physics of Life
TU Dresden

BiAPoL team

- Marcelo Zoccoler
- Johannes Soltwedel
- Maleeha Hassan
- Stefan Hahmann

Former lab members:

- Ryan George Savill
- Laura Zigutyte
- Mara Lampert
- Allyson Ryan
- Conni Wetzker
- Till Korten
- Somashekhar Kulkarni

Communities & platforms

NFDI4BIOIMAGE

image.sc

German BioImaging
Gesellschaft für Mikroskopie und Bildanalyse

neubias
network of european bioimage analysts

GloBIAS
Global BioImage Analysts' Society

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

DFG Deutsche Forschungsgemeinschaft

ScaDS.AI
DRESDEN LEIPZIG

Workshop on AI in microscopy
ChatGPT for Image Analysis
Robert Haase @haesleinhuepf
August 25th 2024

RH acknowledges the financial support by the Federal Ministry of Education and Research of Germany and by Sächsische Staatsministerium für Wissenschaft, Kultur und Tourismus in the programme Center of Excellence for AI-research „Center for Scalable Data Analytics and Artificial Intelligence Dresden/Leipzig“, project identification number: ScaDS.AI

TECHNISCHE UNIVERSITÄT DRESDEN

UNIVERSITÄT LEIPZIG