- Final project instructions and rubric
- Image processing basics in Python

BILD 62

Final Project

Publish | Assign To | Edit | ⋮

The main goal of the project is to demonstrate that you can **design**, **write**, and **implement** code to fulfill some task that you choose. Your final product for this project will be a Jupyter Notebook and corresponding scripts. Throughout the notebook and scripts, clearly document your workflow and thought process in comments, docstrings, and *Markdown* cells. See the final projects slide deck & recording for more information on the objectives, scope, and timeline. See grading rubric below.

Please pay attention to feedback on project proposal

Submission (DataHub, Canvas and Google Drive)

To receive full credit for the final project, you should submit via DataHub and Canvas, and present your work on Mon Dec. 9th (3-6pm). Your grade for the project will be on Canvas.

DataHub

On DataHub, submit everything as you normally would on DataHub but **be sure** that the assignment folder **also** contains any additional scripts/modules/data that you need included. Only one person in your group needs to submit via DataHub.

Canvas

- On this Canvas assignment, submit your Jupyter Notebook and any scripts that you have (DO NOT submit data, if applicable).
- Because the Canvas submission will not be used to run the code in your notebook, it is important to make sure the PDF of your notebook has the code evaluated and outputs present (e.g., plots) so that we can read the project as is.
- Submit your final notebook to Canvas as an exported PDF (in Jupyter Notebook: File > Download As).
- **Each student should submit the project on Canvas.**
  The submission on Canvas should contain your personal reflection on the project, see rubric for details.
- Submit scripts as a second submission as a .py or .txt file.
- **If your absence from the presentation has been excused, submit a video recording of your presentation.**

Google Drive

- By Monday 3pm, upload your presentation to a shared Google Drive folder. All presentations will be made from the same computer. If you have animations, video, etc., make sure they work properly.
  https://drive.google.com/drive/folders/1u81IAGXk7Yh2cPwVtjmZgXNp2kTTOTvO?usp=sharing
- Each group should pick a timeslot on the google sheet.
  https://docs.google.com/spreadsheets/d/1piY2IsDy_6B4u8JPRGHkf6HmwxWsHLfoaAdOGp4H2uE/edit?usp=sharing

BILD 62 Final Projects

Criteria	Ratings			Pts
Project description	**10 to >7 pts** **Meets all expectations** There is a project description in a Jupyter notebook that clearly & concisely explains the project and its motivation in 200-400 words.	**7 to >4 pts** **Meets some expectations** Project description only vaguely motivates the project or is too short.	**4 to >0 pts** **Needs improvement** Project description is very weak/absent	10 pts
Team Contributions	**10 pts** **Meets all expectations** Each team member's contribution is balanced and well-defined	**8 pts** **Meets some expectations** Team member contributions are not well-defined or seem off balance.	**6 pts** **Needs improvement** Team member contributions are not defined	10 pts
Reflection	**10 to >7 pts** **Meets all expectations** Personal reflection is thoughtful, commenting on coding knowledge at the start of the class, and how the project challenged and expanded this knowledge	**7 to >4 pts** **Meets some expectations** Personal reflection is included but not thoughtful	**4 to >0 pts** **Needs improvement** No personal reflection	10 pts
Overall Approach	**20 to >16 pts** **Meets all expectations** The chosen code design & organization makes sense given the project topic outlined; code efficiently meets the challenge in the project.	**16 to >12 pts** **Meets some expectations** Code / approaches / algorithms used for the project somewhat make sense but could be optimized	**12 to >0 pts** **Needs improvement** Code / approaches / algorithms used for the project are inappropriate for the task you are trying to complete, or are not original in any way	20 pts
Functionality	**15 to >12 pts** **Meets all expectations** The code is functional - it does what it is supposed to, with no major errors or bugs, and works as intended	**12 to >9 pts** **Meets some expectations** Code is somewhat functional (small errors, inability to deal with particular use cases)	**9 to >0 pts** **Needs improvement** Multiple aspects of code do not work	15 pts

Contains three functions and/or methods	**15 to >12 pts** **Meets all expectations** Contains three working functions that accomplish clear, specific tasks. These functions can be inspired by code used in class but should not directly copy it.	**12 to >9 pts** **Meets some expectations** Contains three working functions or methods but these are not well-defined in scope or simply re-implement a built-in function in a commonly used Python package (e.g. np.mean())	**9 to >0 pts** **Needs improvement** Contains less than three working functions or methods	15 pts
Variables, Code Constructs, and Imports	**15 to >12 pts** **Meets all expectations** Uses variables, control flow constructs and imports as needed and efficiently. Project demonstrates multiple coding elements from the class.	**12 to >9 pts** **Meets some expectations** Code constructs are unnecessarily complex, long, inefficient, some packages imported that are never used. Project demonstrates few coding elements from class.	**9 to >0 pts** **Needs improvement** Multiple elements of code are inefficient, variables are poorly named, many packages imported that are never used	15 pts
Code Organization	**10 pts** **Meets all expectations** Code is organized into modules & scripts as it makes sense. Front-facing Jupyter notebook is clean. Long functions are imported as scripts rather than in the Jupyter notebook.	**8 pts** **Meets some expectations** Code is somewhat organized but could be improved	**6 pts** **Needs improvement** Code is not clearly organized.	10 pts
Error Handling	**15 to >12 pts** **Meets all expectations** If data: code is resilient to multiple file types OR is clear about what data structure should be, giving clear messages if it does not meet specifications. If user interaction: code should handle various inputs and should be resilient to user error. Regardless: code should use some error catching strategies, such as assert, try/except, and unit tests	**12 to >9 pts** **Meets some expectations** Code implements some error handling strategies but misses several cases	**9 to >0 pts** **Needs improvement** Very little error handling strategies within code	15 pts

Presentation instructions:

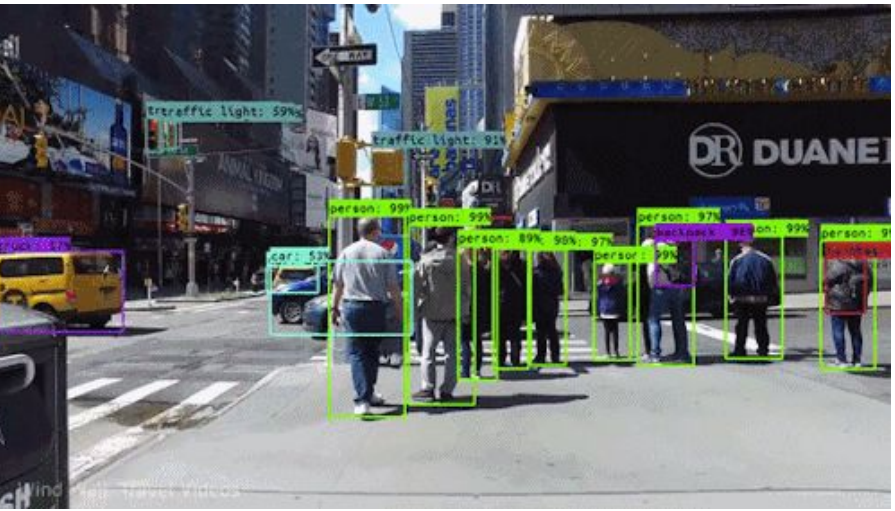
- Include a title slide with project title and names of group members.
- 3+1 minute presentations:
  - Each group member should talk *for approximately one minute* about their contribution.
  - One minute for a quick question.

Presentation tips:

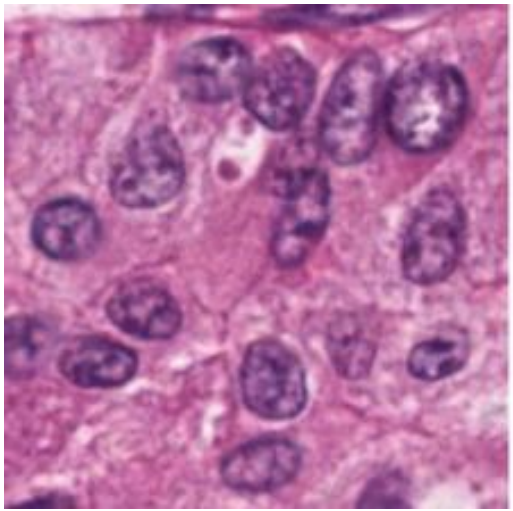
- Practice! Make sure you are on time.
- Think what questions you might be asked and think about an answer.
- Focus more on the problems you solved in Python rather than the biology background. You can include background information necessary to understand what you did, but this background should be minimal.

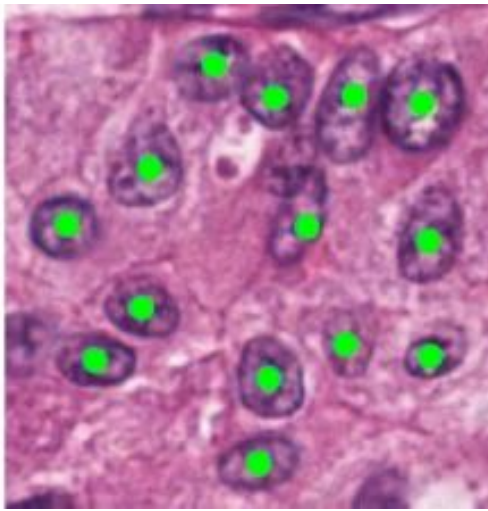
	Meets all expectations Student is present at the final showcase and engaged in presenting their project as well as interacting with other projects	**Meets some expectations** Student is present at the final showcase	**Needs improvement** Student is not present at the final showcase (without emailing professor)	10 pts
				Total Points: 150

From self driving cars to segmenting nuclei, image processing is important!



From [this article]





From [this paper]

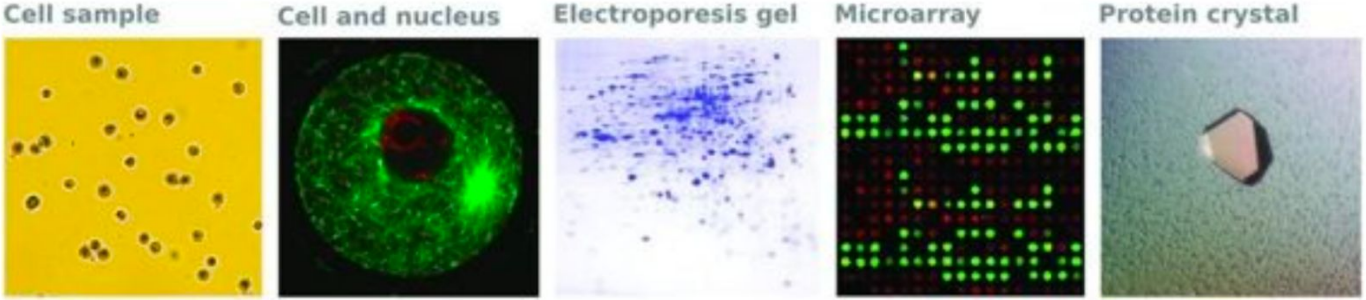


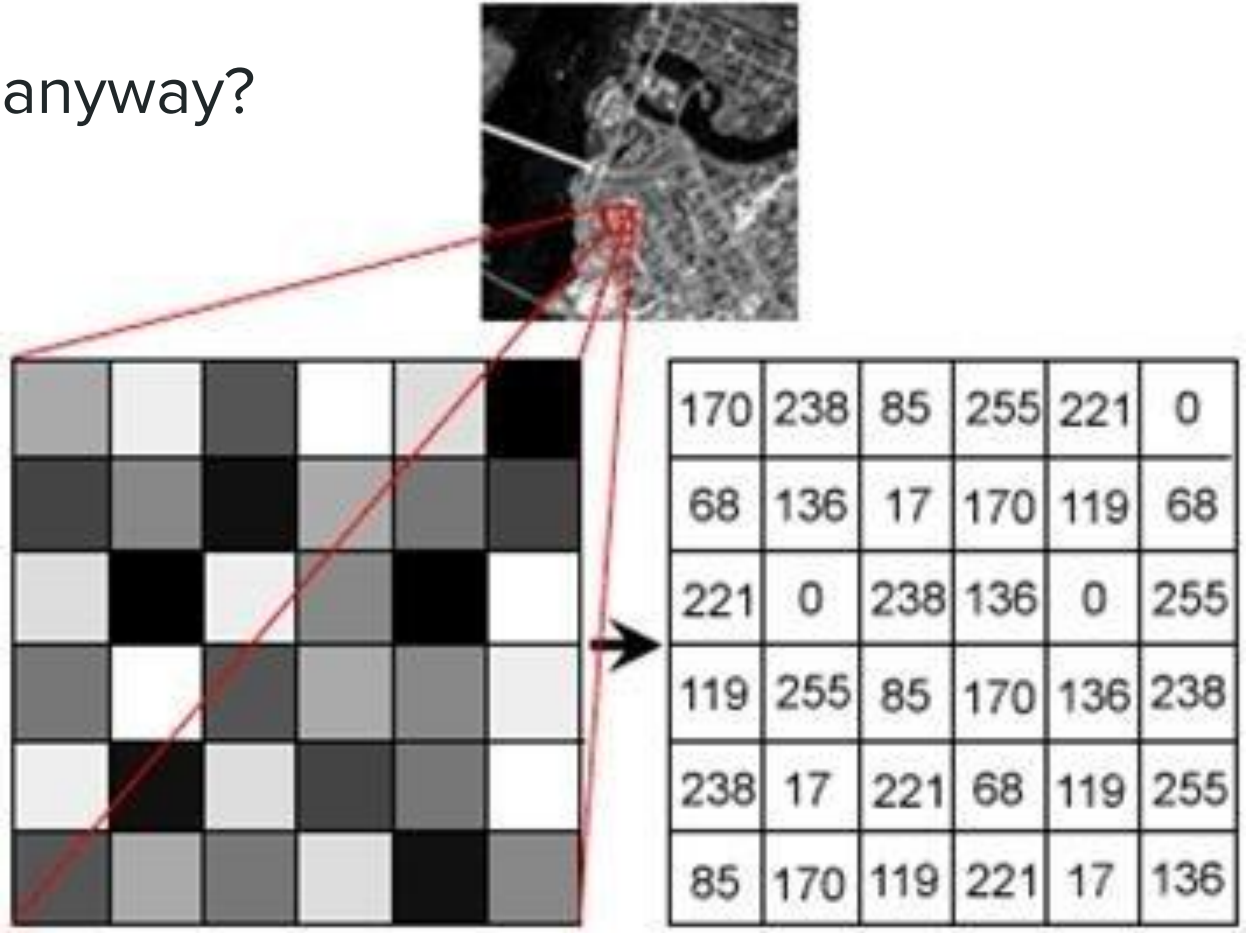
Figure 18.1 (Plate 5). Examples of a variety of different kinds of images used in biology. Shown from left to right are: a microscope image of a mammalian cell culture (courtesy Dr. Anja Winter, University of Leicester); a red-green fluorescence microscope image of an oocyte and its nucleus (courtesy Dr. Melina Schuh, MRC Laboratory of Molecular Biology); a two-dimensional electrophoresis gel of a plant proteome (courtesy Prof. Paul Dupree, University of Cambridge); an image of a DNA microarray (courtesy Karen Howarth, University of Cambridge); a protein crystal that has been grown for structure determination by X-ray crystallography (courtesy Dr. Aleksandra Watson, University of Cambridge).

We use lots of images in biology

Figure from Python Programming for Biology

What are images, anyway?

Gray scale images mean each pixel has just one value

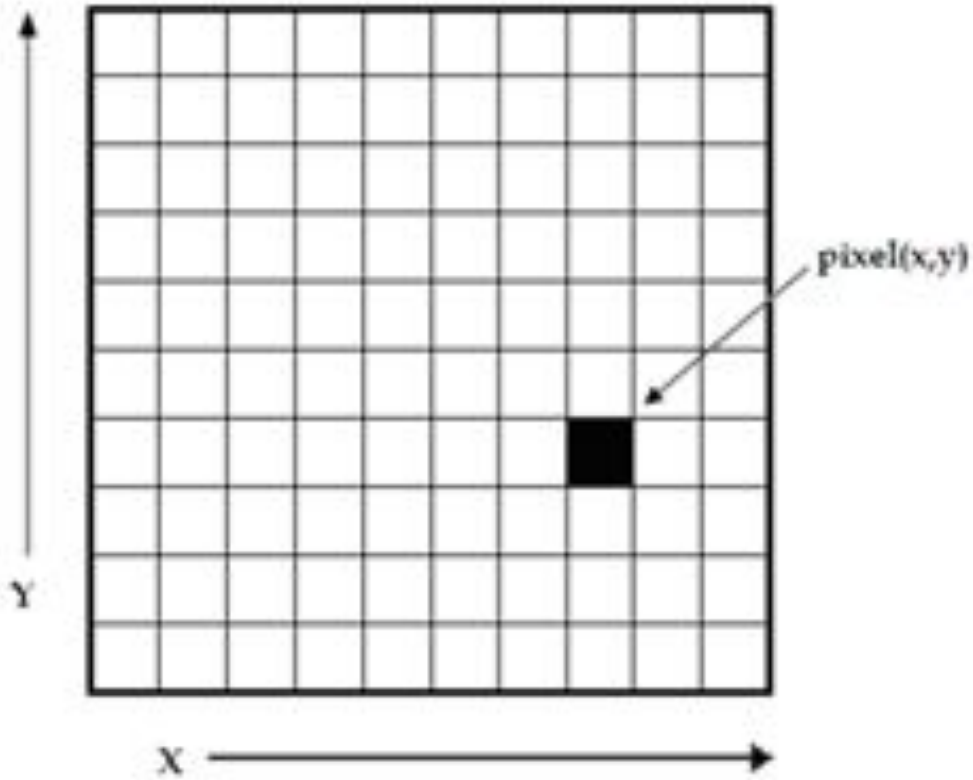


170	238	85	255	221	0
68	136	17	170	119	68
221	0	238	136	0	255
119	255	85	170	136	238
238	17	221	68	119	255
85	170	119	221	17	136

What are images, anyway?

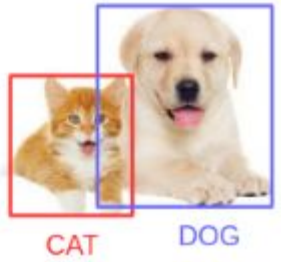
Images can be represented as 2D arrays

By convention [0,0] is the top left corner



Often, we want to perform different types of image segmentation: localization or object detection

Image Localization

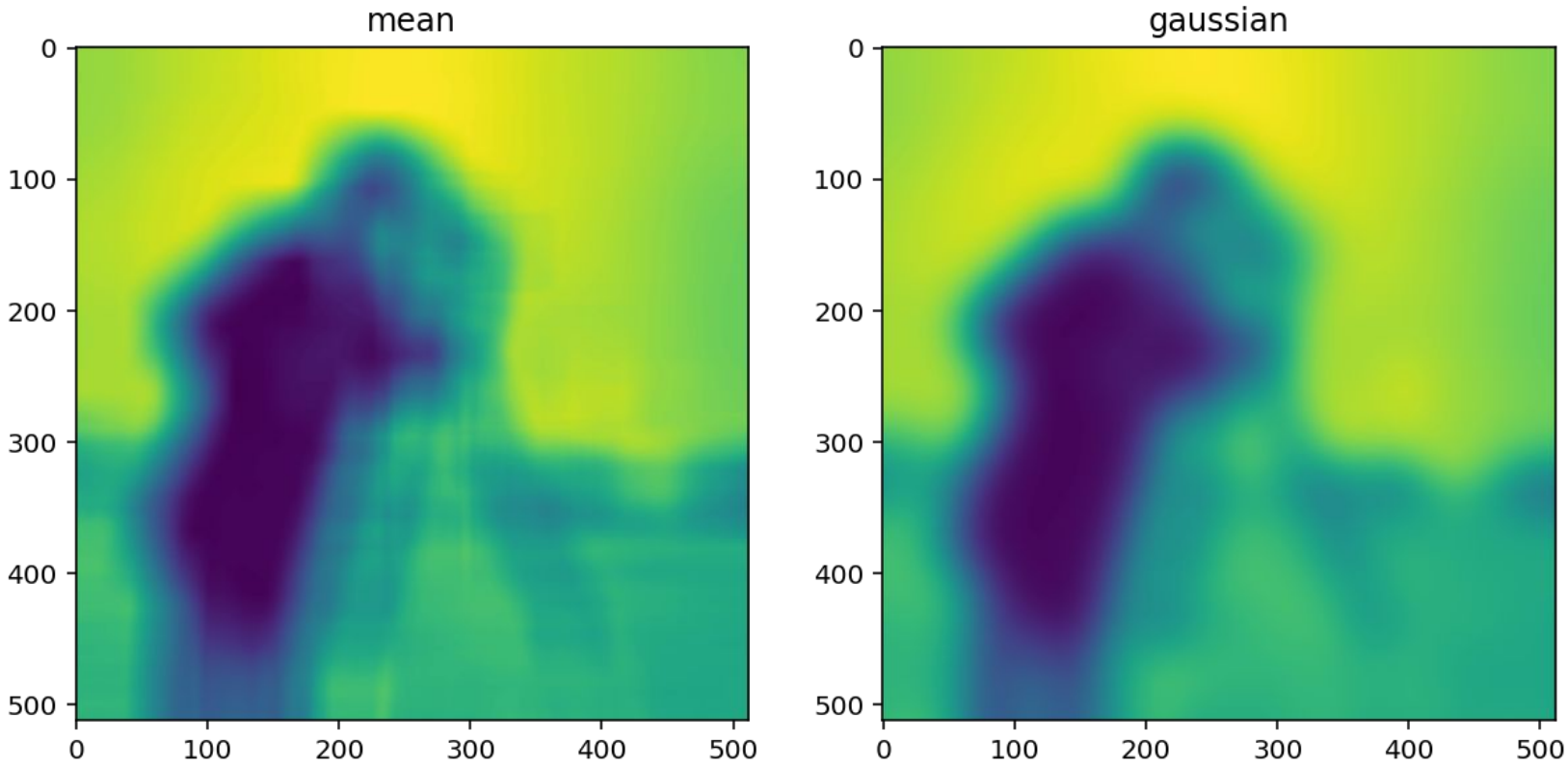


Object Detection

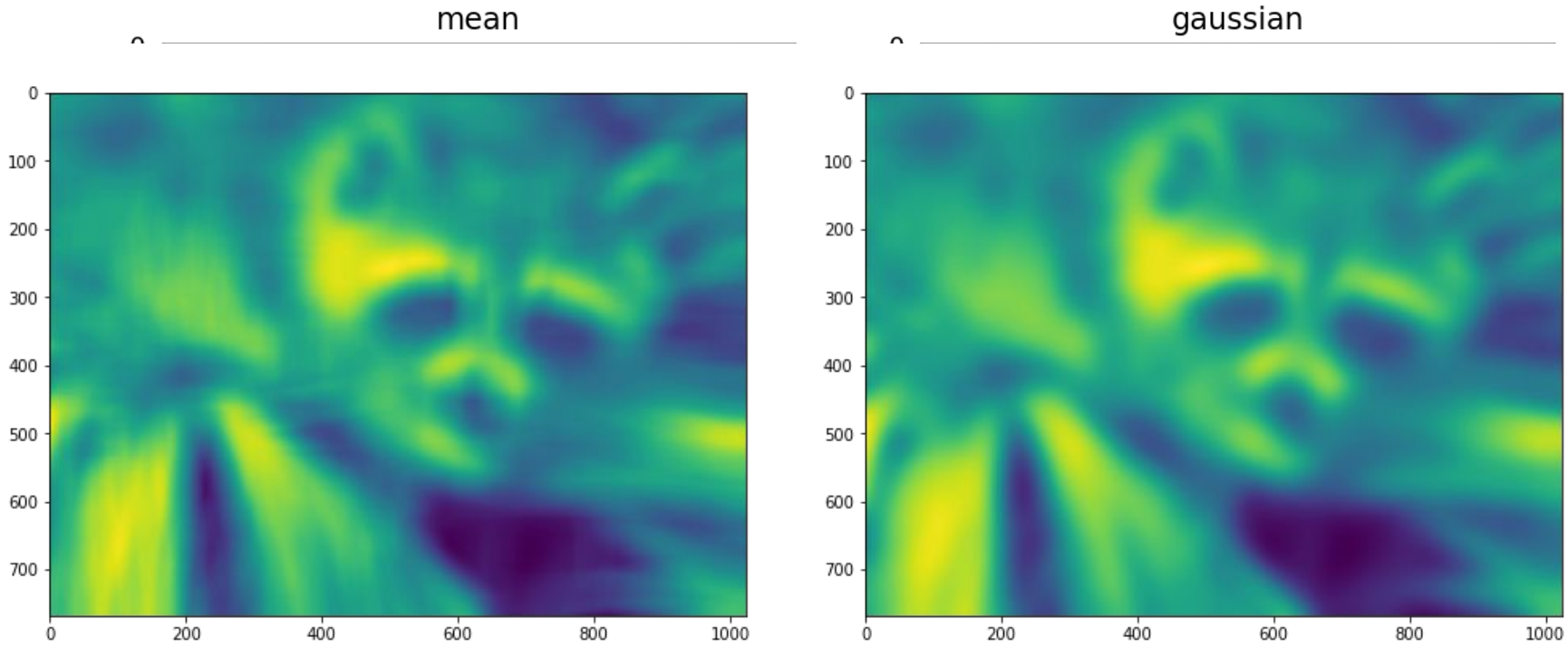
https://www.analyticsvidhya.com/blog/2019/04/introduction-image-segmentation-techniques-python/

Commonly used filters for biological images

- Gaussian filter — to smooth and remove irregularities
- Edge filters — to detect edges
  - Sobel filter



Mean vs. Gaussian smoothing



Mean vs. Gaussian smoothing

Image processing tools based in Python

cellpose https://github.com/MouseLand/cellpose

Napari https://github.com/napari/napari

Additional resources

https://www.youtube.com/watch?v=1GUgD2SBl9A

https://www.youtube.com/watch?v=uihBwtPIBxM

https://www.analyticsvidhya.com/blog/2019/04/introduction-image-segmentation-techniques-python/

https://jni.github.io/i2k-skimage-napari/lectures/0_i2k_bioimage_analysis_fundamentals.html