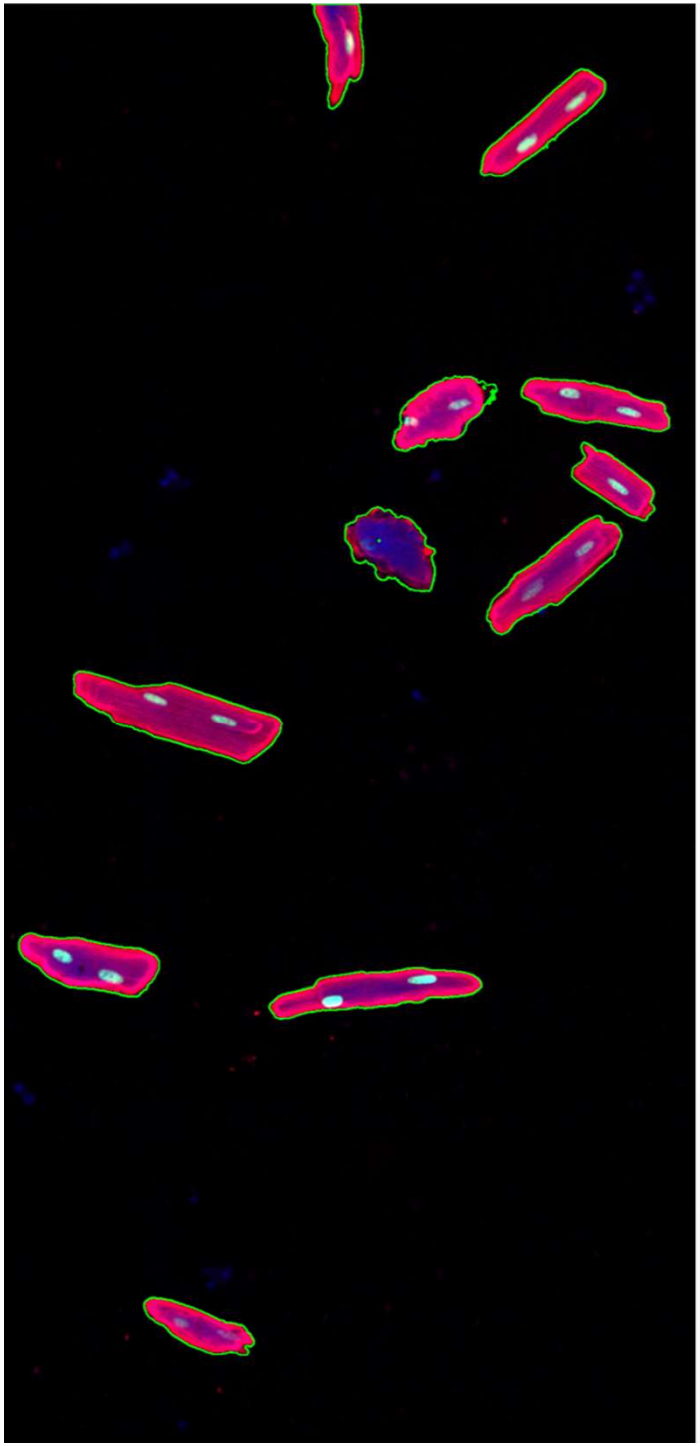


Introduction to Image Analysis

Jian Wei Tay

BioFrontiers Institute

Advanced Light Microscopy Core

Postdoctoral Association of CU Boulder

Mar 06 2020

jian.tay@colorado.edu

@JianTay1

Getting started

https://gitlab.com/jwtay/intro-image-analysis-pac2020

- Download `cell.tif` and copy it to your working directory
- If you have Git installed, you can clone the repository

  git clone https://gitlab.com/jwtay/intro-image-analysis-pac2020.git

- Create a new script

Installing required toolboxes in MATLAB

- Image Processing Toolbox
- Home tab > Add-Ons > Manage Add-Ons > Get Add-Ons (above search bar)
- Search for "Image Processing Toolbox"
- Select "Sign in to Install"/"Install"

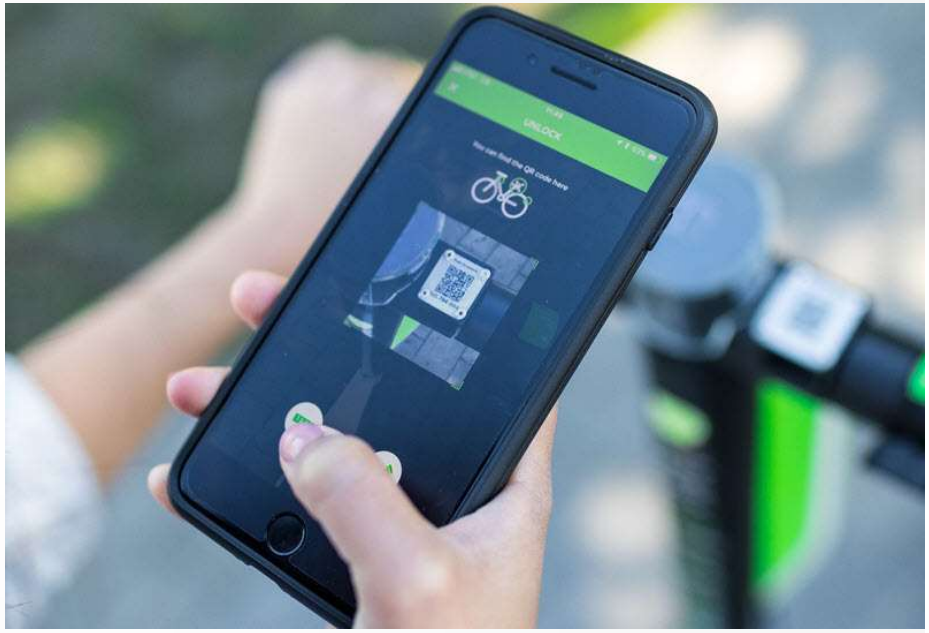
Please feel free to stop me and ask questions!

What is image analysis

- Image analysis is the process of extracting information from digital images
- Image analysis ≠ Image processing
- Image processing refers to adjustments
  - Cropping
  - Removing noise
  - Enhancing low-light images
- Image analysis typically includes image processing

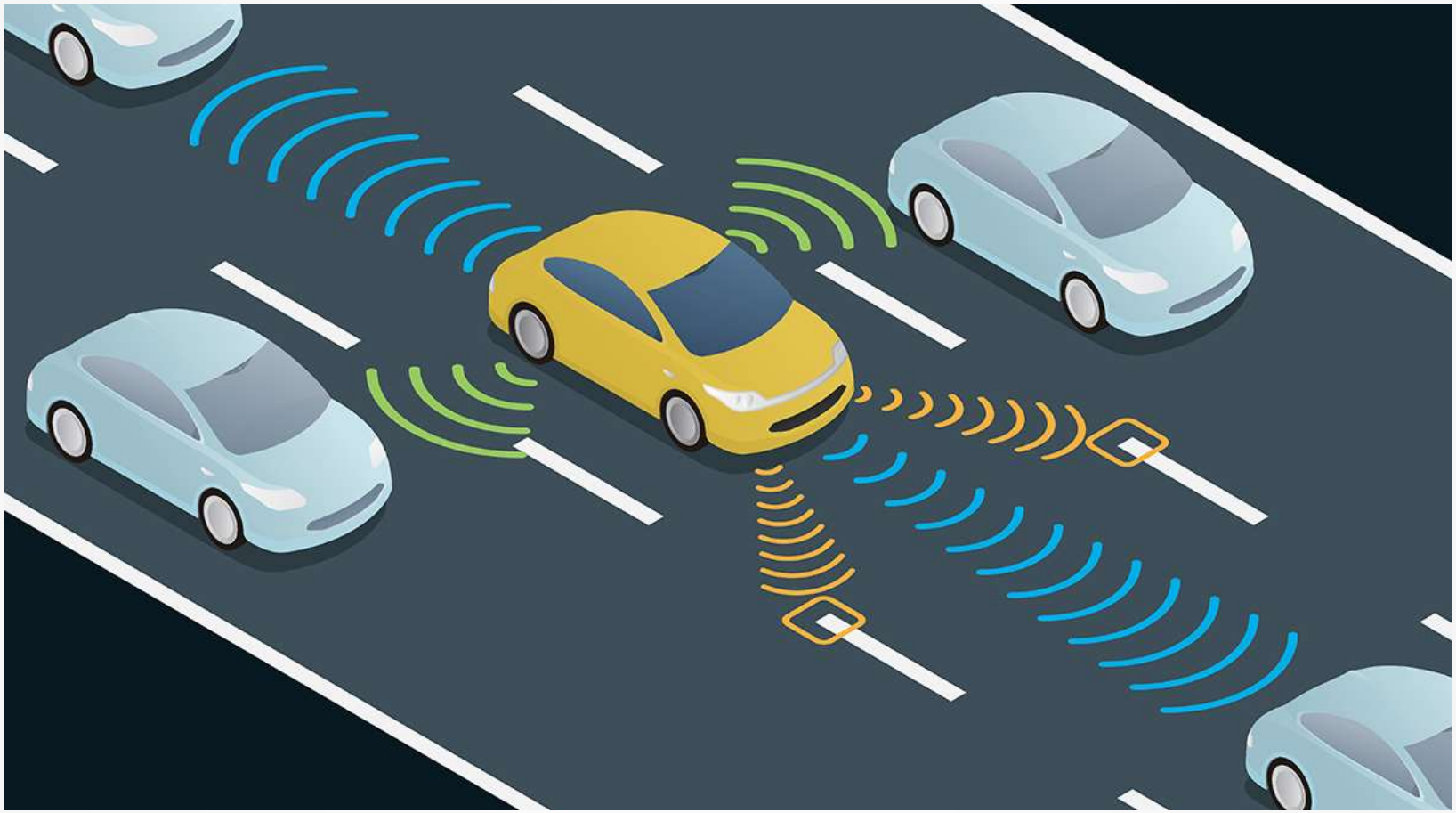
Examples of image analysis

Reading QR Codes



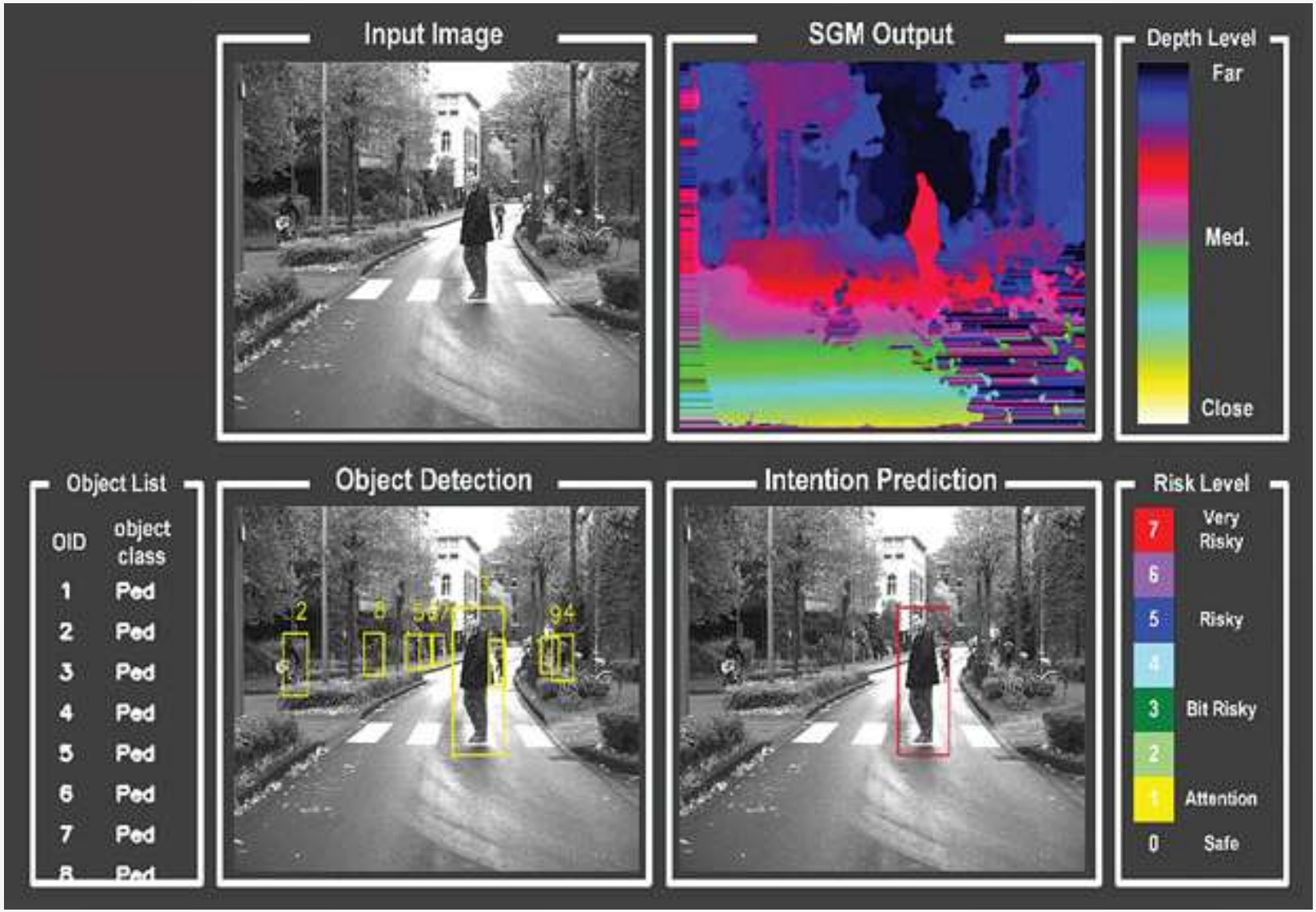
Lime

Self-driving cars



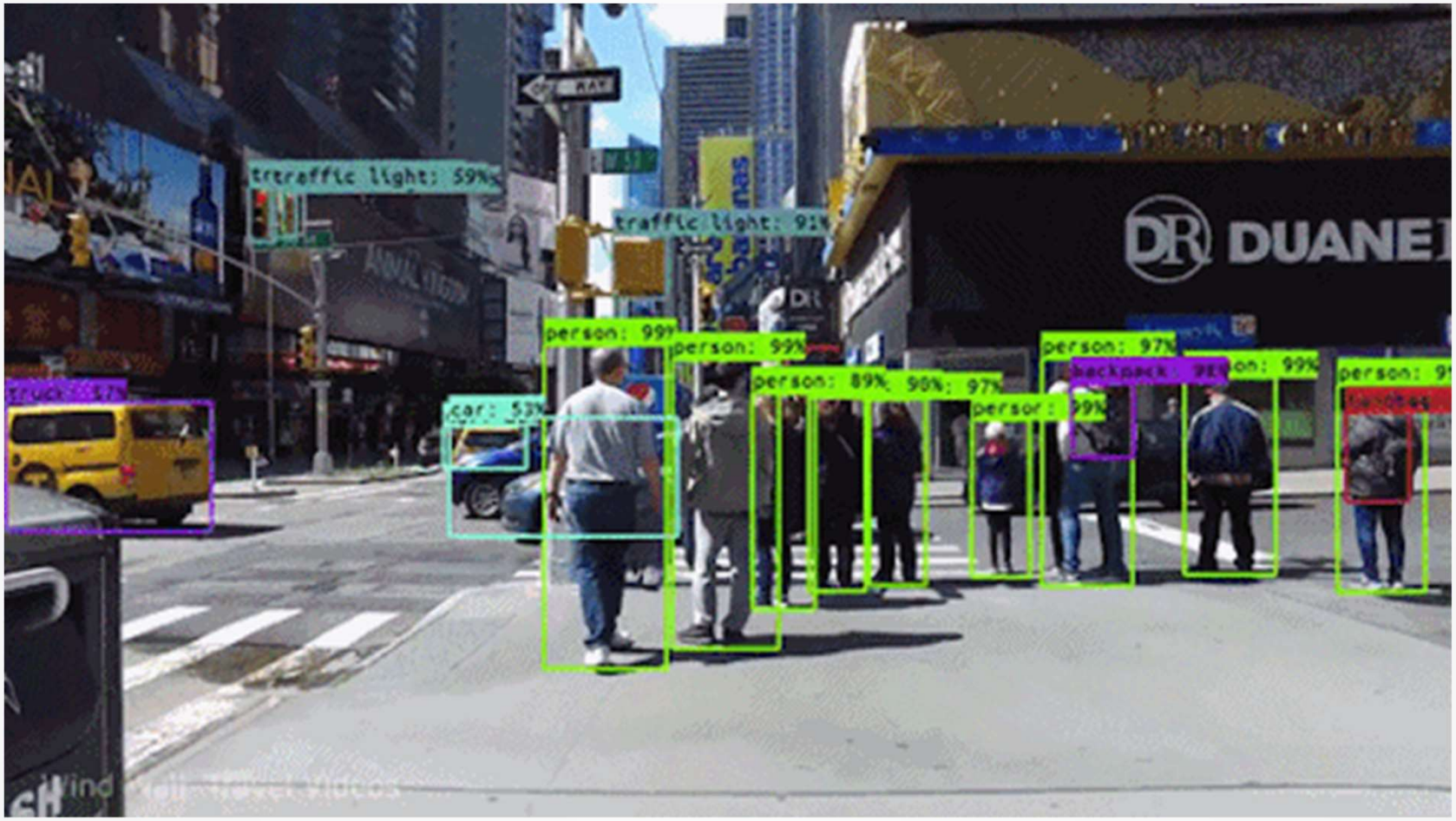
https://www.ttnews.com/articles/analysis-self-driving-cars-need-standardized-software-avoid-crashes

Self-driving cars



https://spectrum.ieee.org/computing/embedded-systems/bringing-big-neural-networks-to-selfdriving-cars-smartphones-and-drones

Object identification



https://towardsdatascience.com/how-do-self-driving-cars-see-13054aee2503

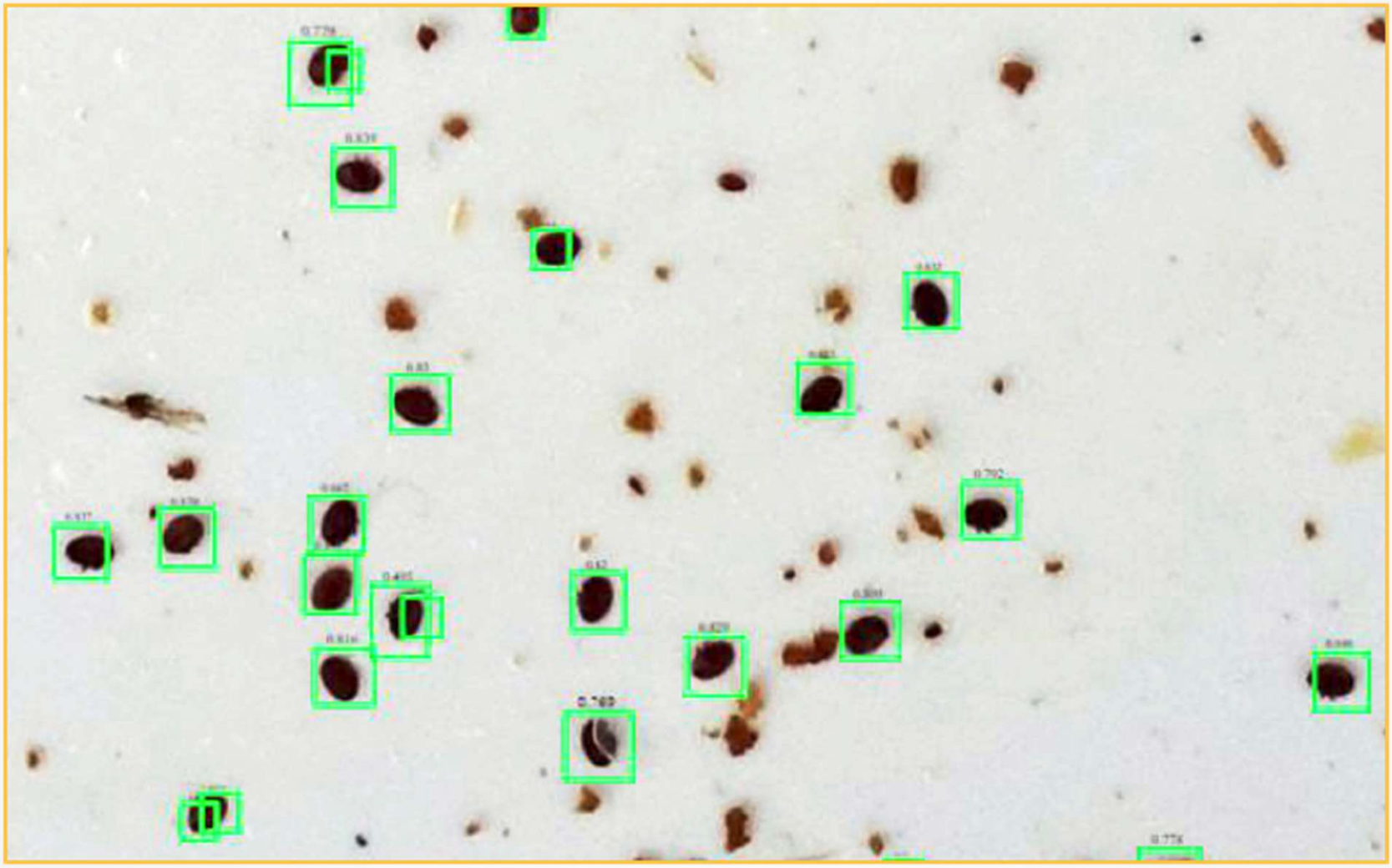
Agriculture

Varroa mites are parasitic mites that feed on honeybees

Can spread and destroy hives

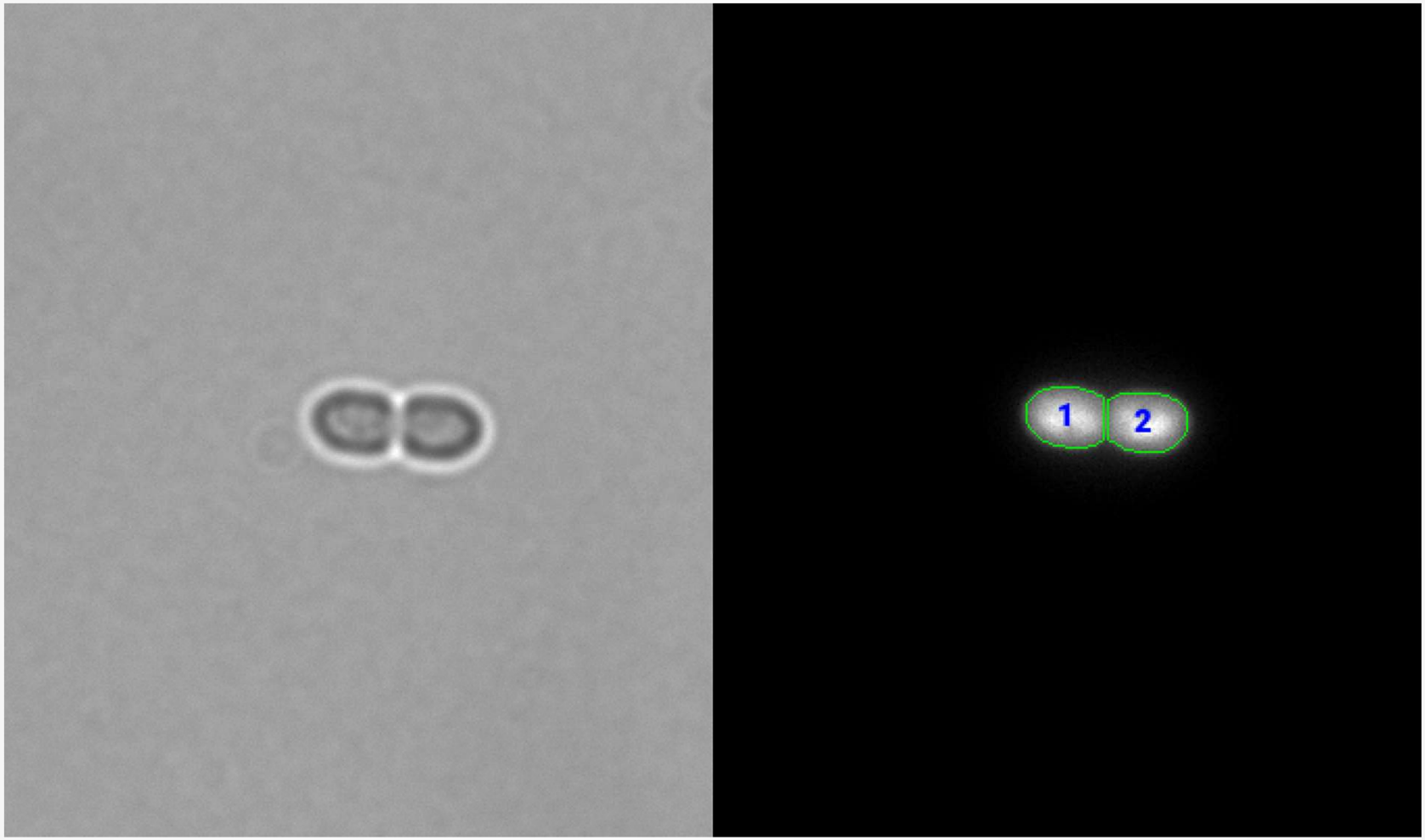
Science magazine

Monitoring mite load – There's an app for that!



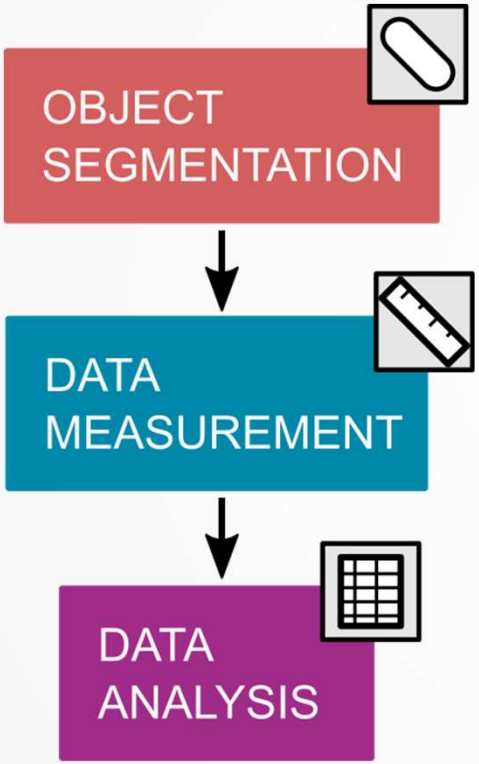
https://www.apizoom.app/

Cell microscopy

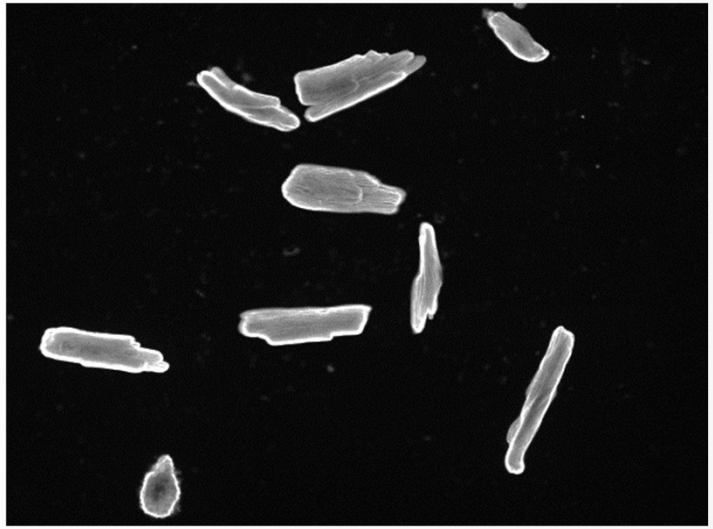


Outline

Image analysis pipeline

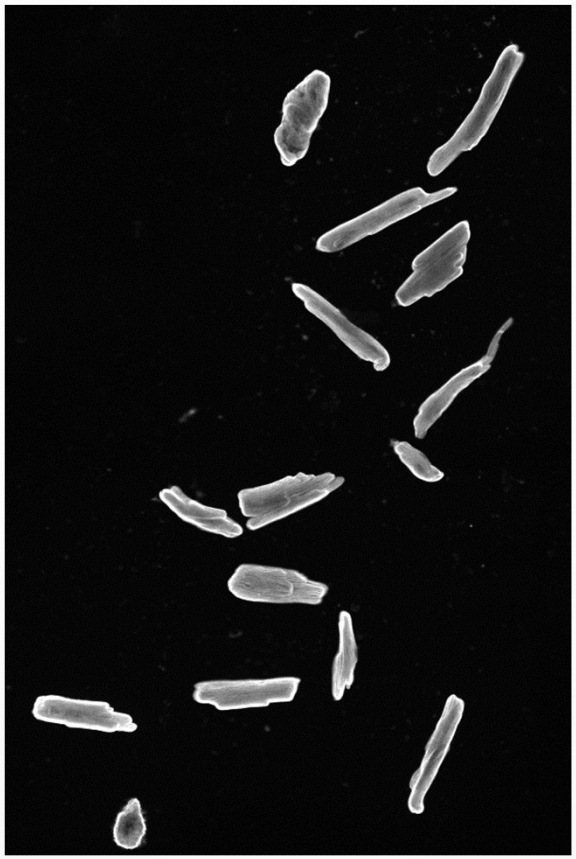


Identify and count the number of cells in an image



Read and display image

```
I = imread('cells.tif');
imshow(I);
```



The matrix as an image

- Image data is stored as a matrix
- Pixel values (matrix elements) are proportional to the amount of light arriving at the camera

Cameras are made up of pixels which sense light

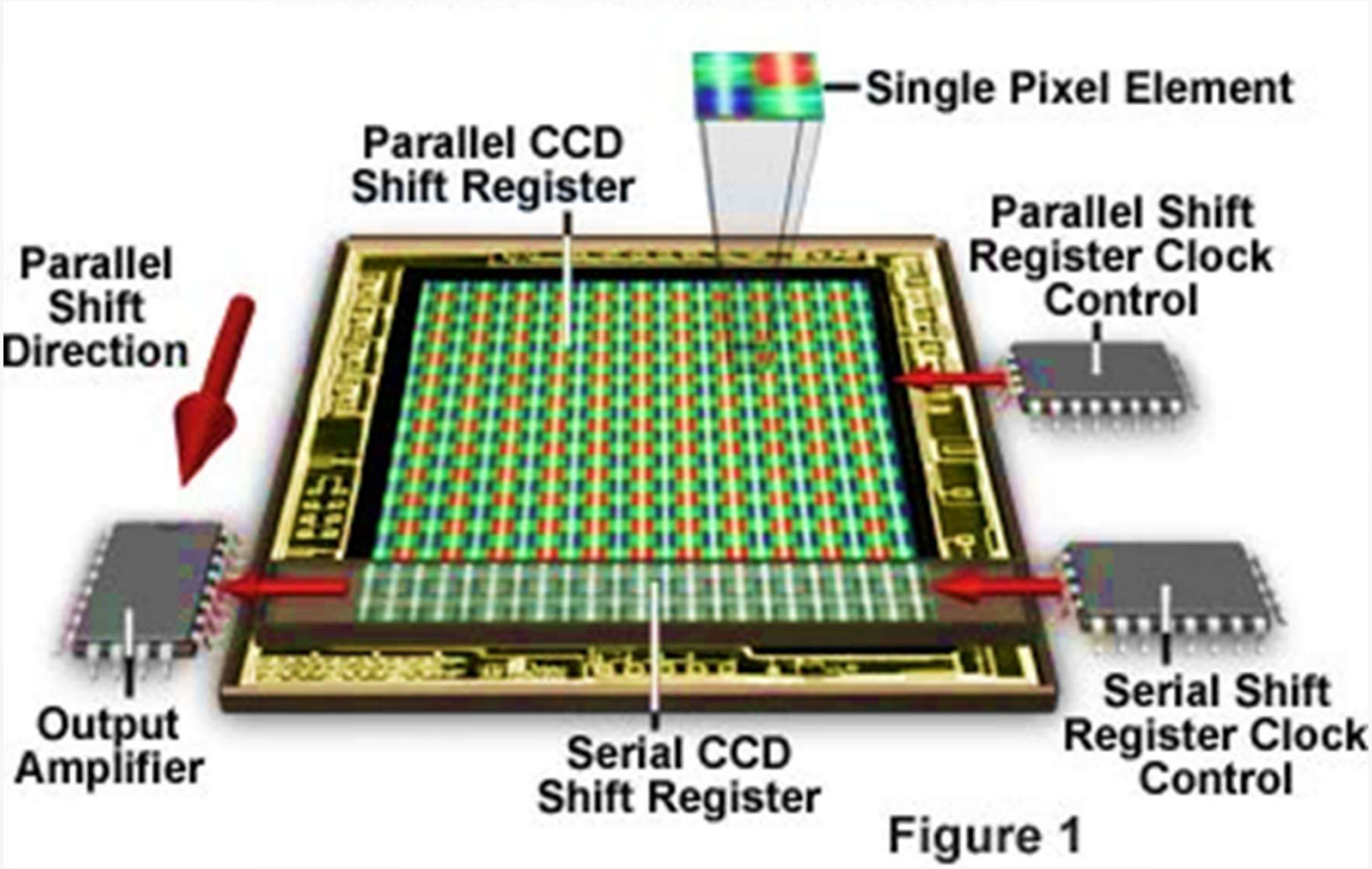
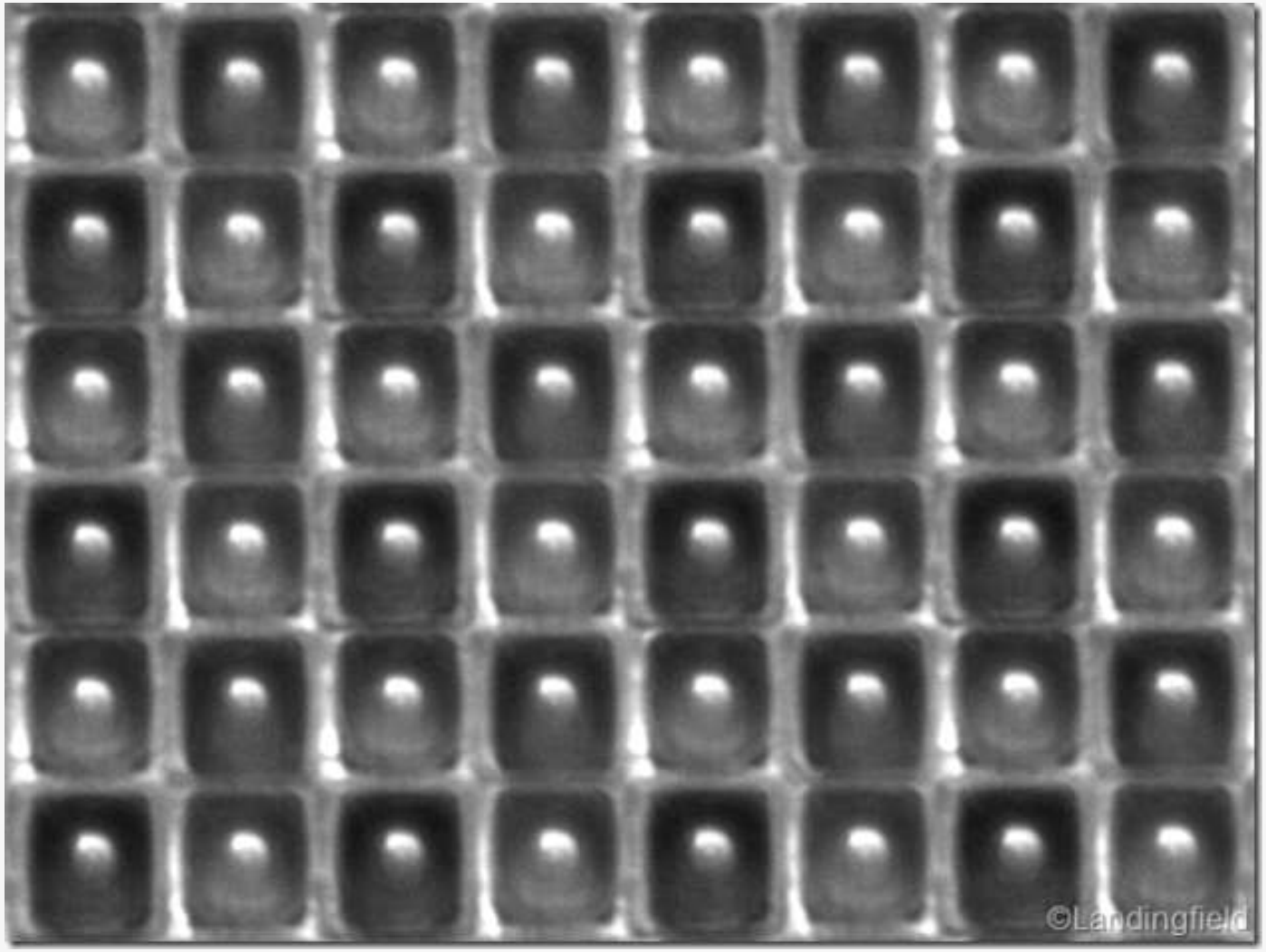


Figure 1

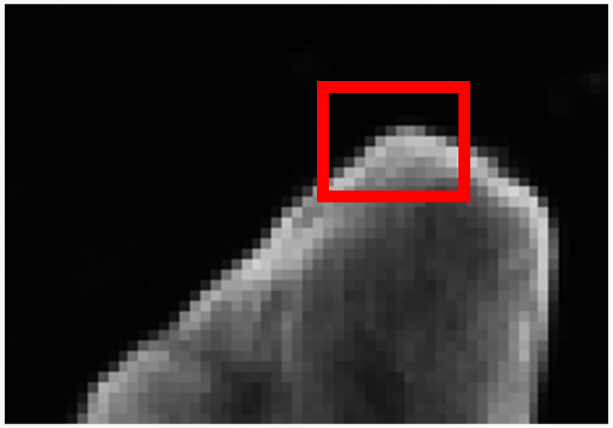
https://www.olympus-lifescience.com/en/microscope-resource/primer/digitalimaging/concepts/fullframe/

Cameras are made up of pixels which sense light



https://landingfield.wordpress.com/2013/02/06/peeping-into-pixel-a-micrograph-of-cmos-sensor/

Pixel value is proportional to intensity



717	903	621	709	898	719	642	717	748
603	852	736	674	854	857	699	774	862
680	771	845	656	656	795	851	855	826
743	794	804	872	819	750	735	703	831
654	801	787	690	760	765	847	764	717
720	713	830	712	760	742	840	727	893
826	827	1465	2639	2836	2455	1392	801	909
1367	5355	15745	29835	33954	26919	10861	5200	2880
10010	35722	45549	45090	48515	48938	48414	40945	23469
39876	47697	40529	40393	42619	46166	45452	46076	51689
43298	41279	37058	39235	38086	33828	36430	42236	48575
35926	33456	36935	32915	29291	30213	33151	34151	36410
30744	34123	28316	28607	25909	26563	27494	23572	24449
29009	27604	25771	24170	21809	22445	19099	16297	18891
27202	24527	20556	21673	20800	19050	13170	14654	15201
23379	20439	18702	21495	17883	13002	12585	13834	13534
23566	20047	16644	19202	16166	13060	11006	13946	12904
20230	18094	15817	15470	14530	13522	9928	13409	13351
15872	14687	15497	15713	13172	11391	11165	12301	12499

The higher the value, the more light arrived at that pixel

The matrix as an image

- Image data is stored as a matrix
- Pixel values (matrix elements) are proportional to the amount of light arriving at the camera
- You can use matrix commands to index portions of the image or to get the measurement from a specific pixel

```
Icrop = I(1:100, 1:100)
Ipixel = I(25, 30)
```

Image data type

- Typically image data is stored as unsigned integers
- Important because unsigned integers have no negative numbers and no decimal places

5/2 = 3

- Convert to double before doing math (function `double`)

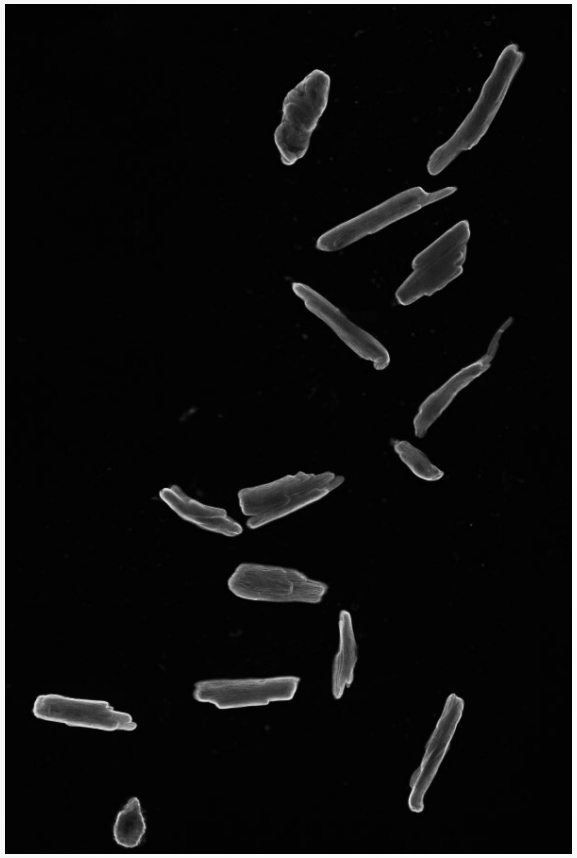
Unsigned integers

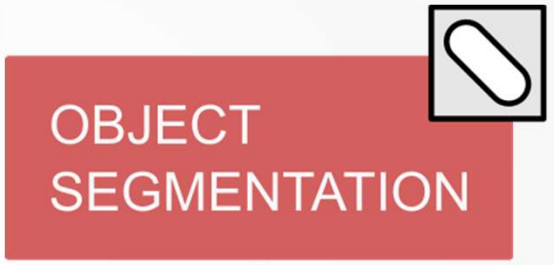
- `uint16` means unsigned 16-bit integer
- Minimum value = 0 (unsigned)
- Maximum value = $2^N - 1$

$$2^{16} - 1 = 65535$$

- Values above or below these extremes are capped

The image analysis pipeline

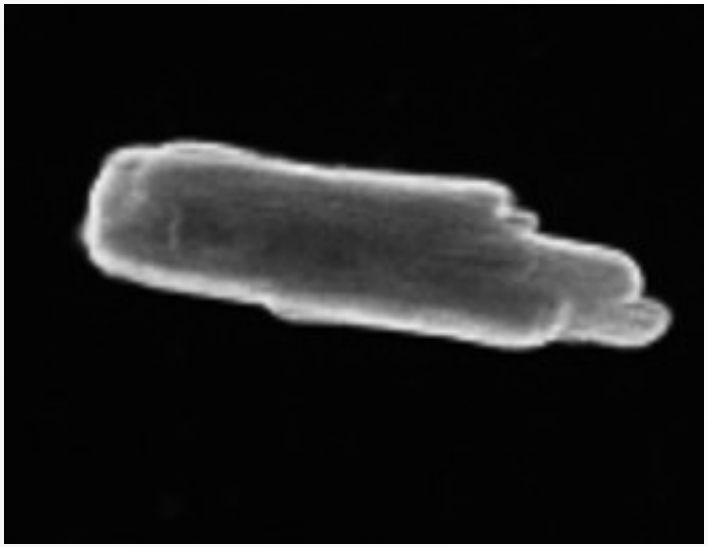


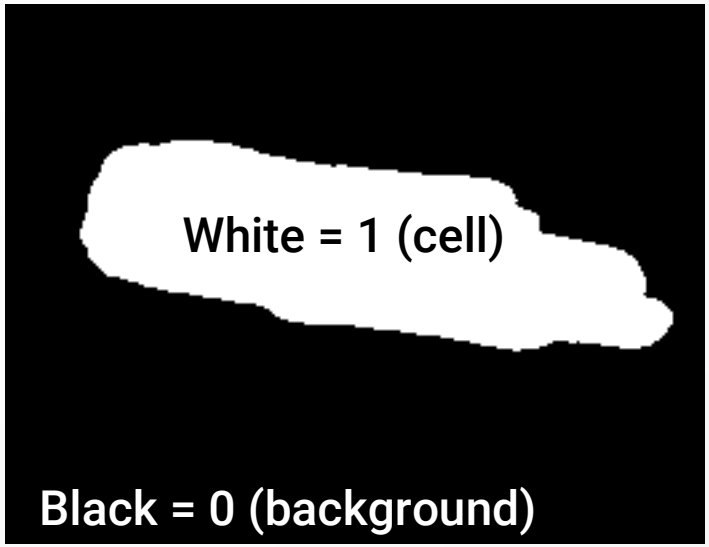


Object segmentation

- Segmentation is the process of identifying objects in an image

- Output is a "mask" – a logical array where true = cell, false = background





Mask

Object segmentation

- Segmentation is the process of identifying objects in an image
- Output is a "mask" – a logical array where true = cell, false = background
- Typically performed by using intensity thresholding

Manual intensity threshold

- Enable data tips in the figure window
- Scroll around and choose an appropriate value

```
mask = I > threshold_value
imshow(mask)
```

Image intensity histogram

- `histogram(I, 100)` plots a histogram with 100 bins
- `ylim([0 2500])` sets limits on the y-axis

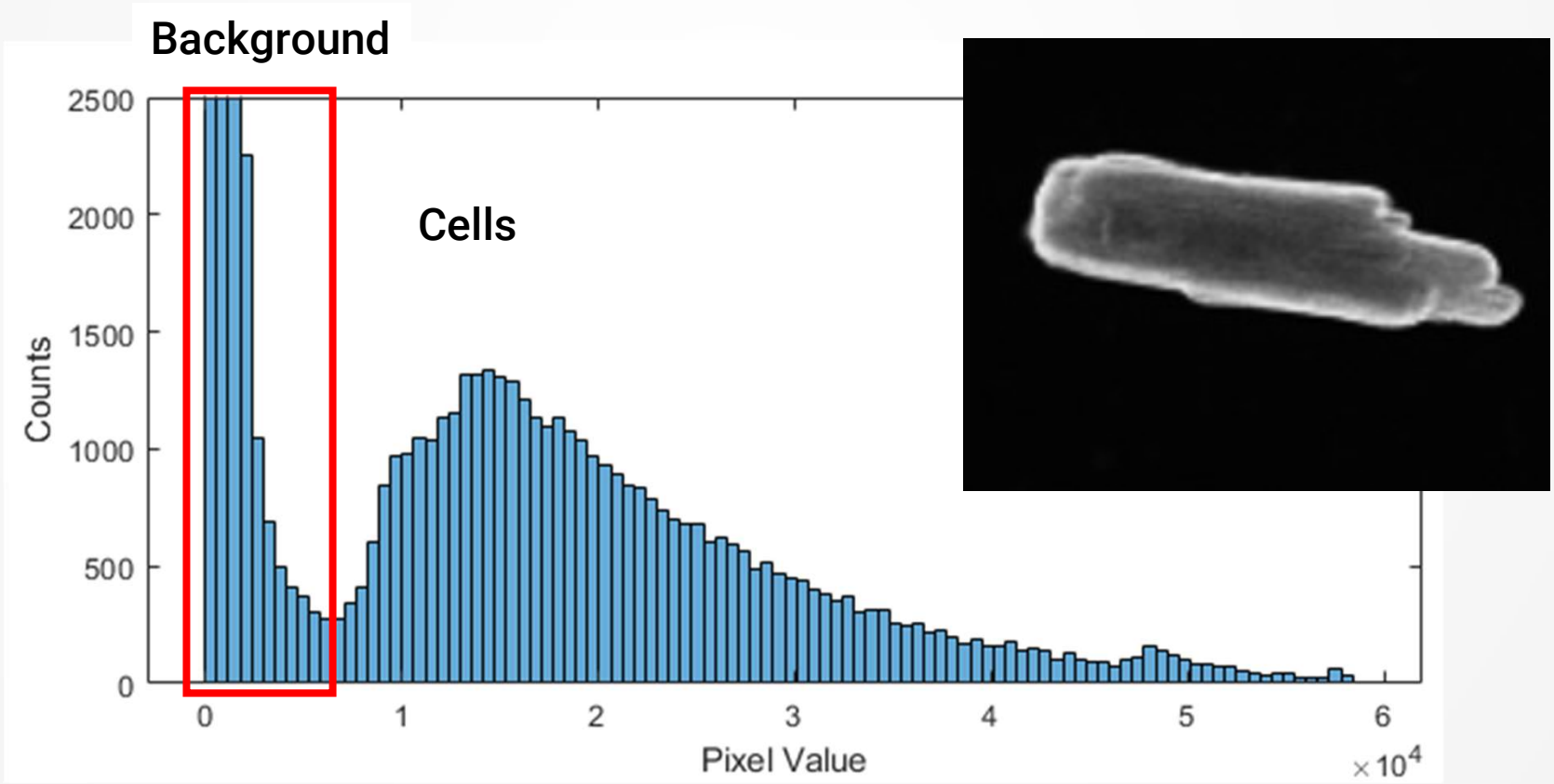
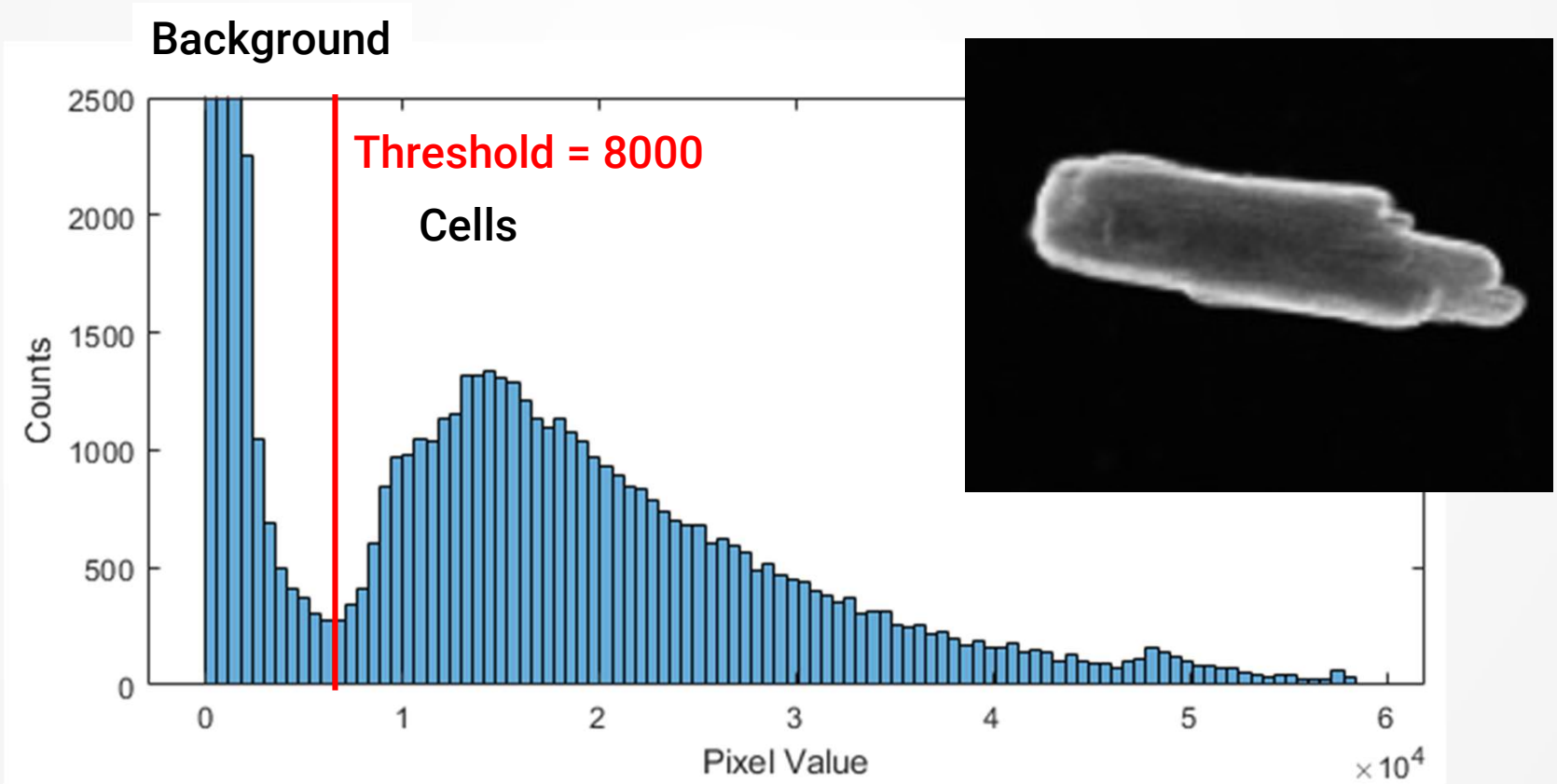


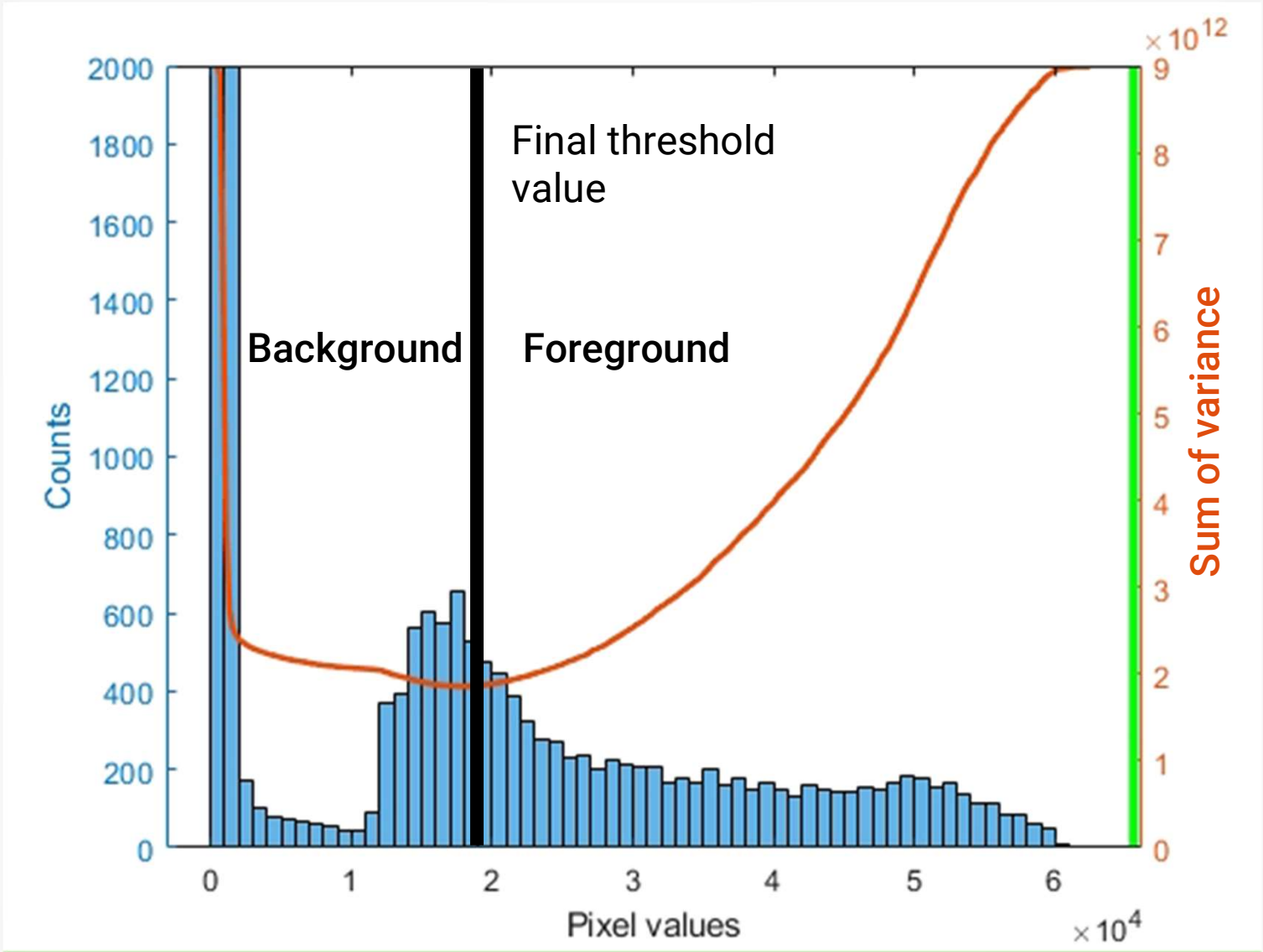
Image intensity histogram

- `histogram(I, 100)` plots a histogram with 100 bins
- `ylim([0 2500])` sets limits on the y-axis



Otsu's method

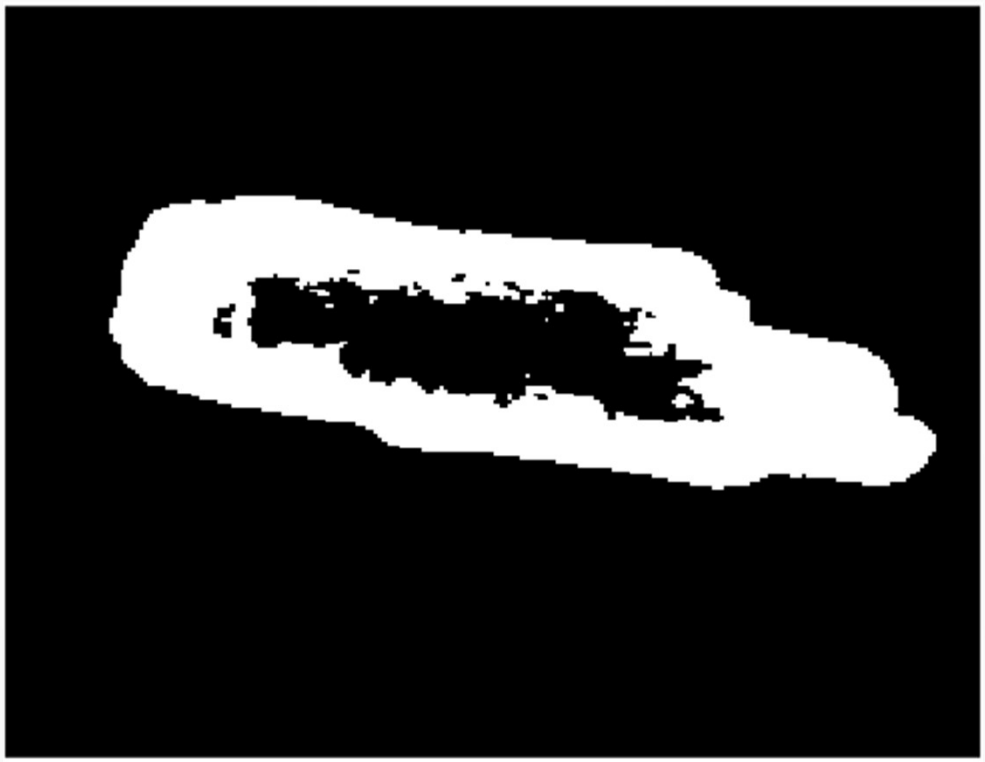
- Automatic threshold finding algorithm
- Uses the image intensity histogram to select the "optimal" threshold



Sum of variance = (weighted) variance on left + (weighted) variance on right

Otsu's method

```
mask = imbinarize(I);
```



Otsu's method

```
mask = imbinarize(I);
mask = imfill(mask, 'holes');
```

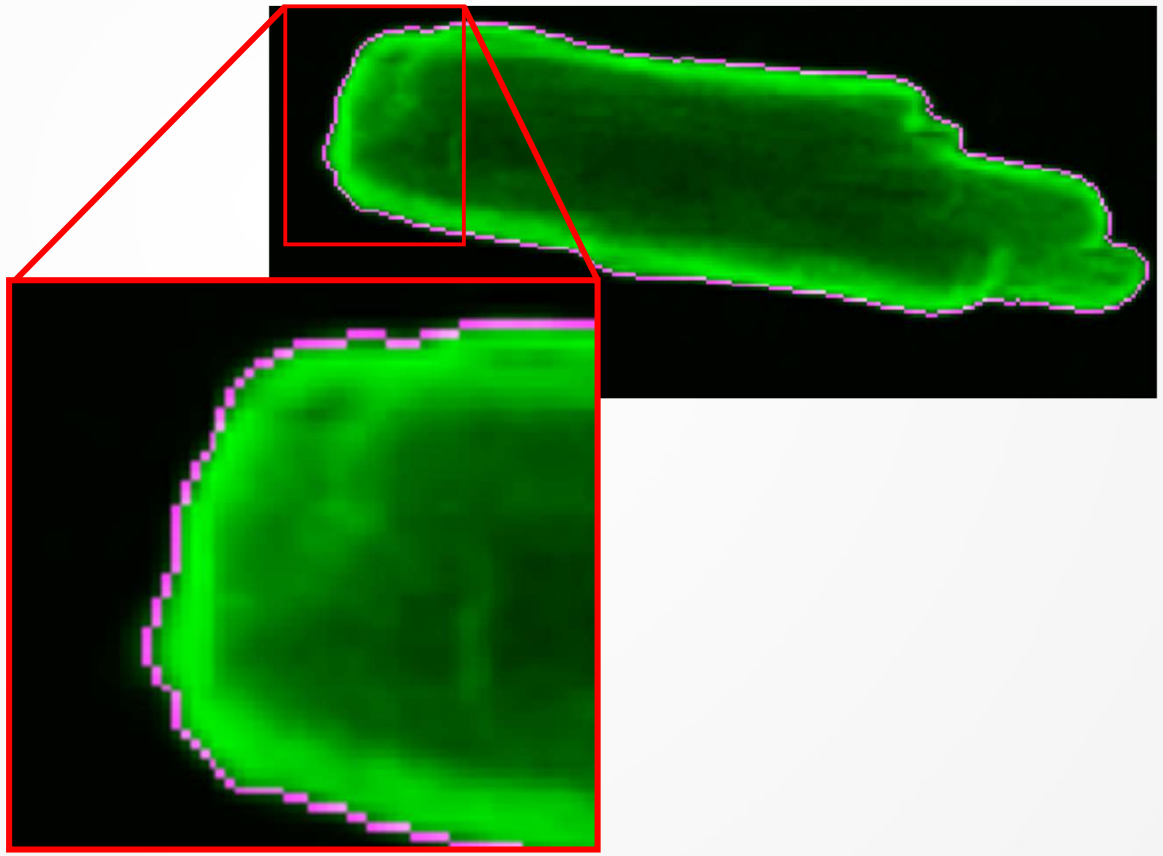
Holes are zeros surrounded by ones

Otsu's method

- Automatic threshold finding algorithm
- Uses the image intensity histogram to select the "optimal" threshold
- Caveat: You must have strong contrast between the object and background (i.e. cells must be brighter than the background)

Visualizing the mask

```
perimeter = bwperim(mask);
imshowpair(I, perimeter)
```

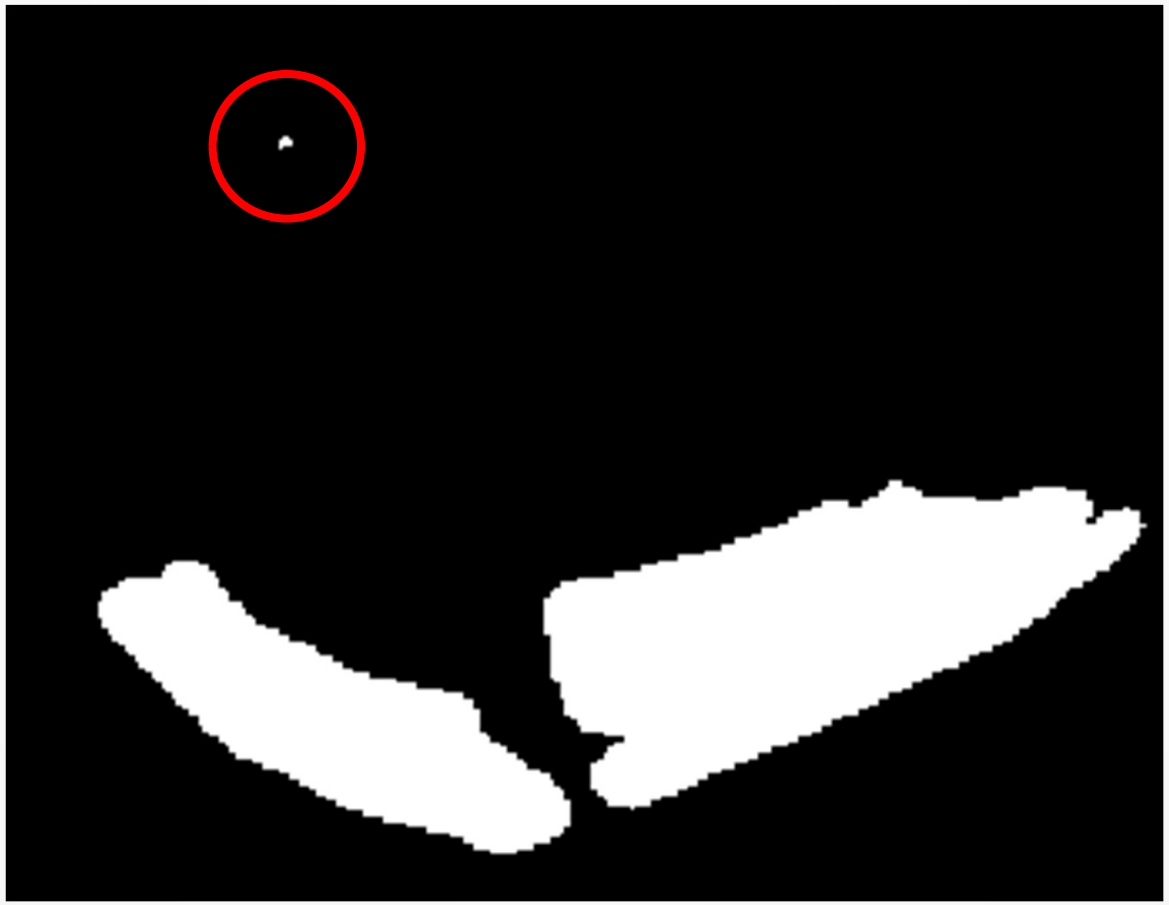


First image `I` is shown in green

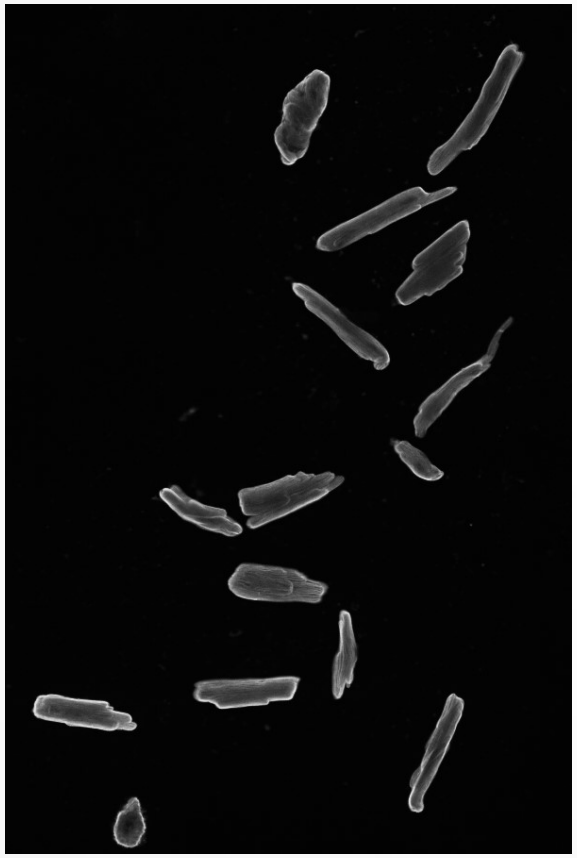
Second image `mask` is shown in magenta

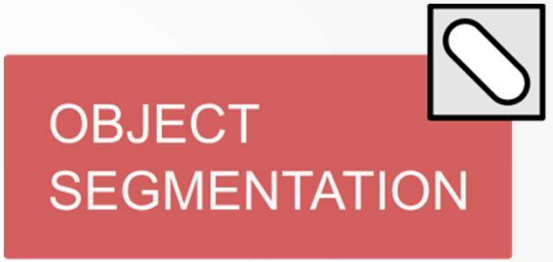
Removing small areas

- `bwareaopen(mask, 100)` removes true regions that are less than 100 pixels in size



The image analysis pipeline





Measuring cell properties

- The Image Processing toolbox comes with a function called `regionprops`

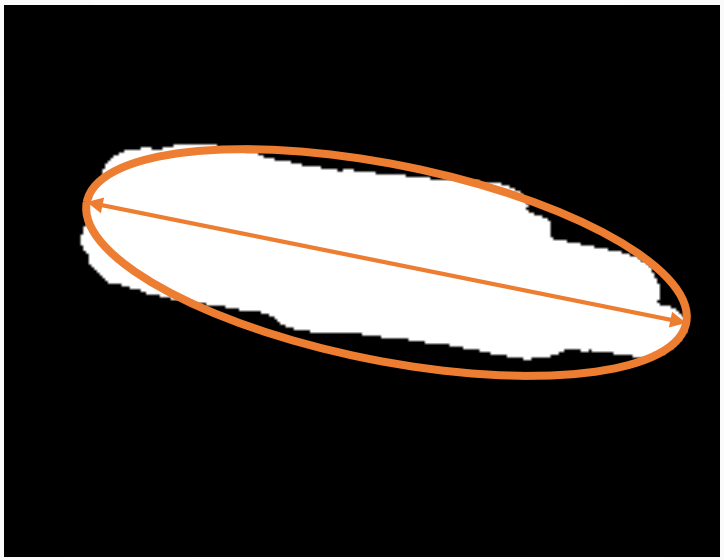
- `regionprops` can measure many things:
  - Area
  - Centroid (center coordinate)
  - Circularity
  - MajorAxisLength (length)
  - MinorAxisLength (width)
  - MeanIntensity
  - … more (see `help regionprops`)

Measuring cell properties

```
celldata = regionprops(mask, I, ...
   'MeanIntensity', 'MajorAxisLength')
```

Measuring length

- MajorAxisLength

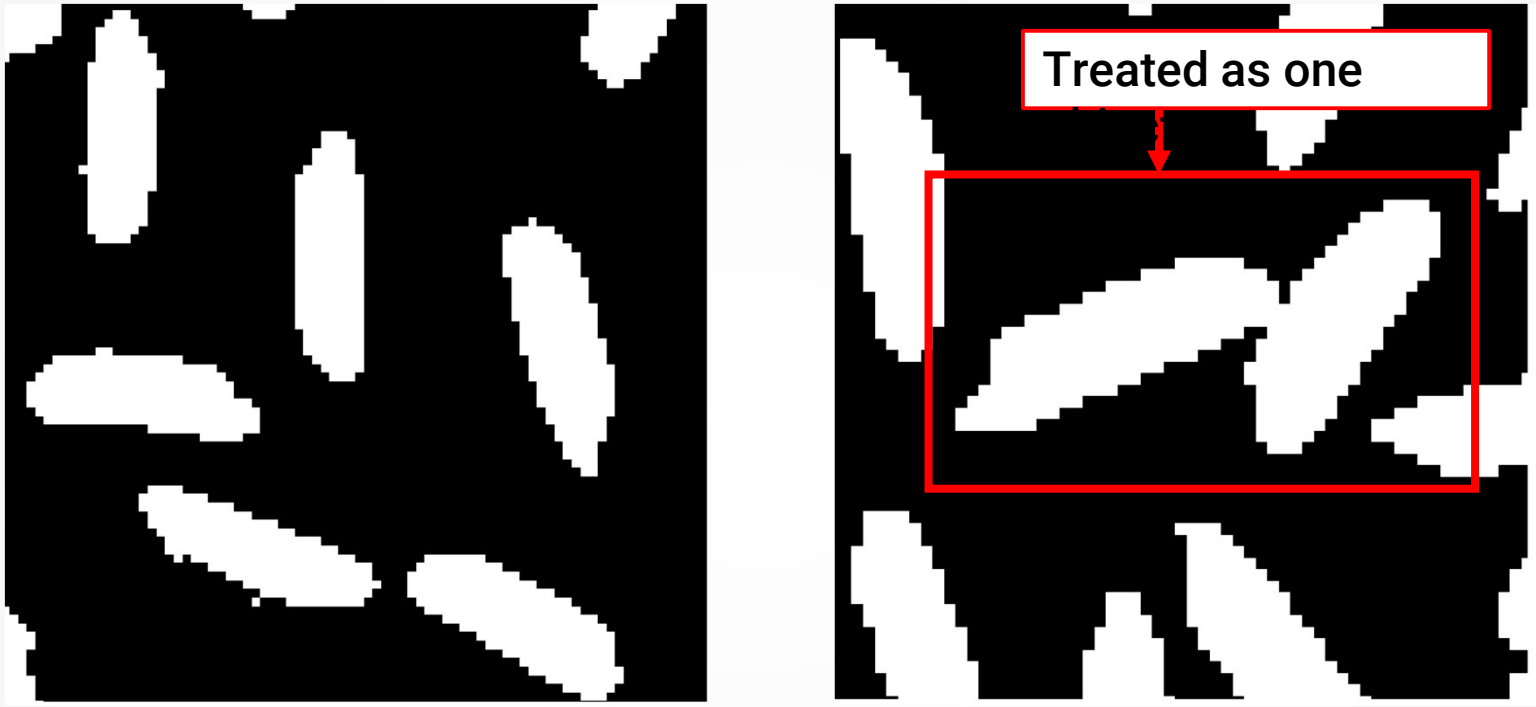


To figure out length, MATLAB fits an ellipse to the mask

Same method for MinorAxisLength and Orientation (angle of the object)

https://blogs.mathworks.com/steve/2010/07/30/visualizing-regionprops-ellipse-measurements/

regionprops treats unconnected regions as individual objects



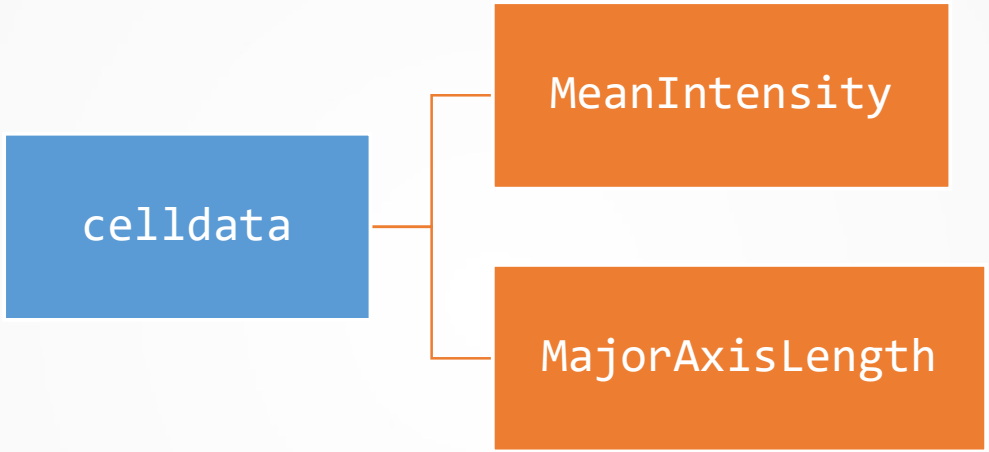
You can use techniques like watershedding and morphological operations to try to separate objects

https://www.mathworks.com/company/newsletters/articles/the-watershed-transform-strategies-for-image-segmentation.html

https://homepages.inf.ed.ac.uk/rbf/HIPR2/morops.htm

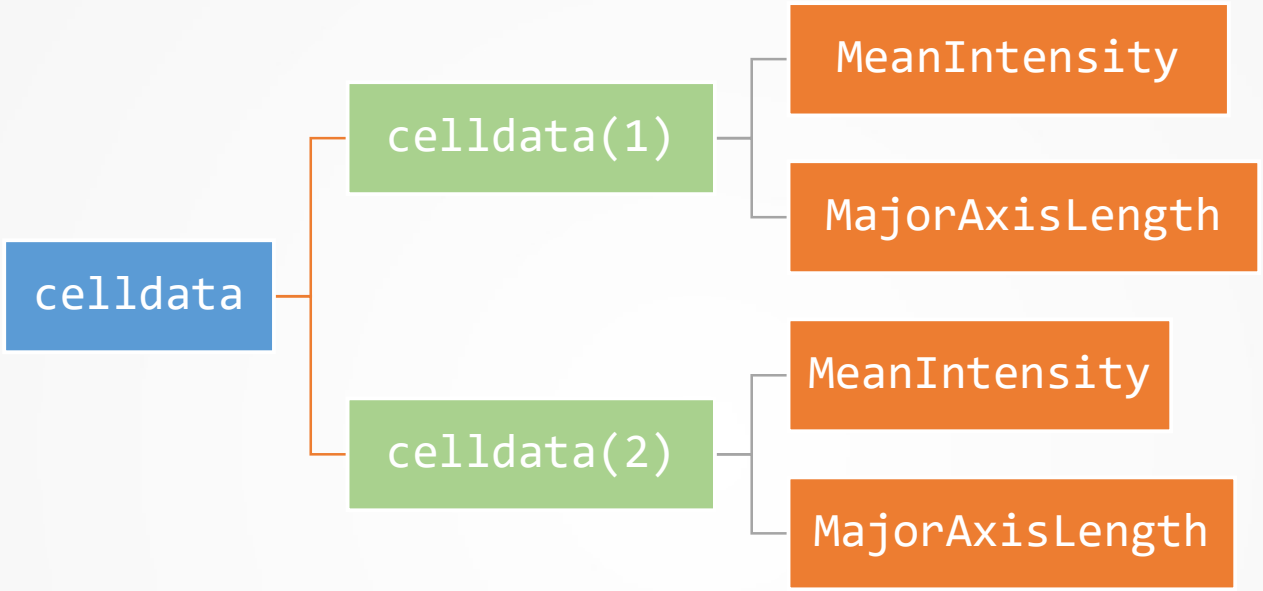
Structured Arrays (`struct`)

- A `struct` is a basic MATLAB data type



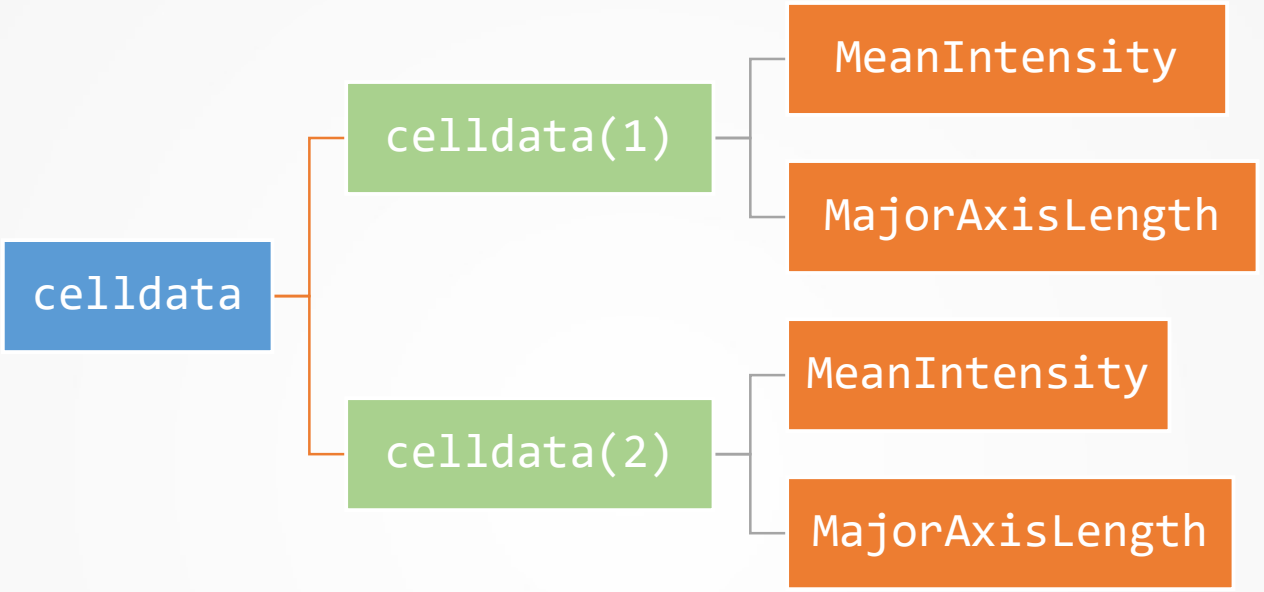
- Data is stored in fields
- Data in different fields can have different types and sizes
- Fieldnames are case-sensitive (`regionprops`: always uppercase first letter)

Multi-element structure arrays



- `regionprops` **outputs a multi-element structured array**
- **Every struct element has the SAME fields**

Accessing data from a struct



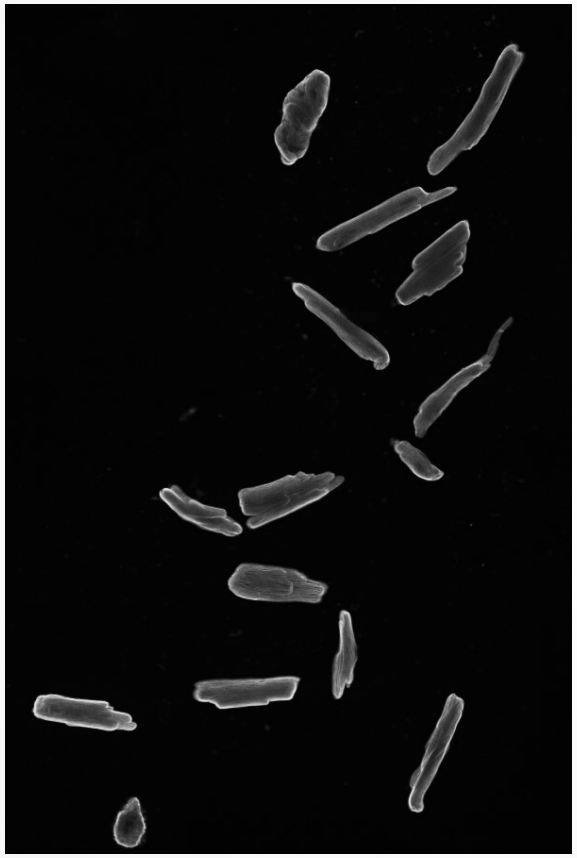

```
celldata(1).MeanIntensity
```

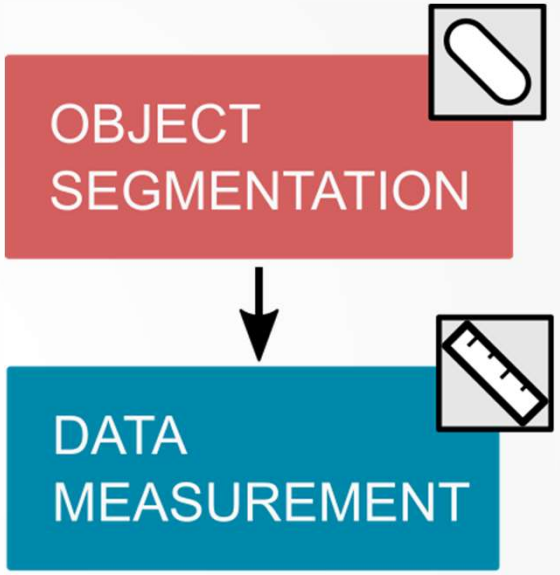
Measuring cell properties

- Number of cells = number of elements in the `celldata` struct

- `numCells = numel(celldata)`

The image analysis pipeline





Analyzing data

Compute the mean cell length

1. Concatenate (join) the values together using the function `cat`

```
lengths = cat(1, celldata.MajorAxisLength);
```

2. Compute the mean

```
mean(lengths)
```

Note: MATLAB converts the uint16 to double when computing the mean

Converting units

- `regionprops` returns values in pixels
- You need to know/calibrate distance per pixel
- For these images, 0.4596 µm/pixel
- So mean in microns = `mean(lengths) * 0.4596`

Visualizing the data

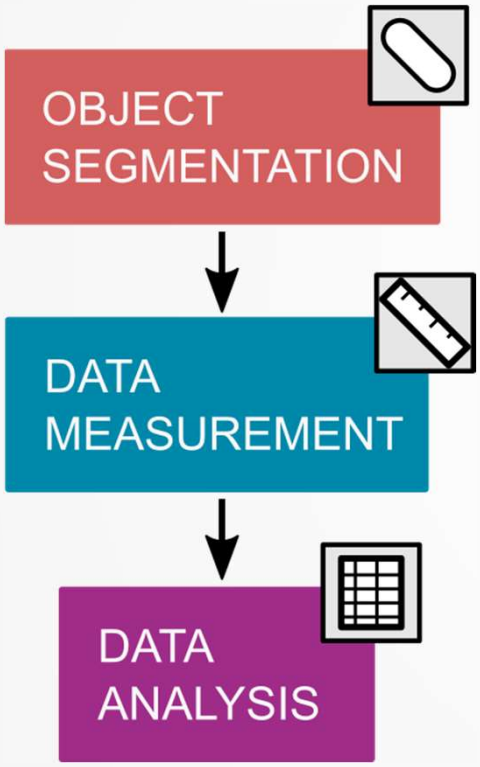
- Histograms

```
histogram(cat(1, celldata.MajorAxisLength), 30)
```

- Plots

```
plot(cat(1, celldata.MeanIntensity))
```

Summary



`imread` – **Read image**
`imshow` – **Display image**

`imbinarize` – **Otsu's method for segmentation**
`imfill` – **Fill in holes in mask**
`bwareaopen` – **Remove small areas in mask**

`regionprops` – **Measure data**
`cat` – **Concatenate (join) data**

`numel` – **Number of elements (cells)**
`mean` – **Compute mean**
`histogram`, `plot` – **Visualize data**

Resources

- Image analysis toolboxes I have developed:

  https://biof-git.colorado.edu/biofrontiers-imaging

- Lecture notes from Quantitative Optical Imaging (2019)

  https://jwtay.cc/teaching

- Workshop coming this summer @ JSCBB

  If interested, send me an email