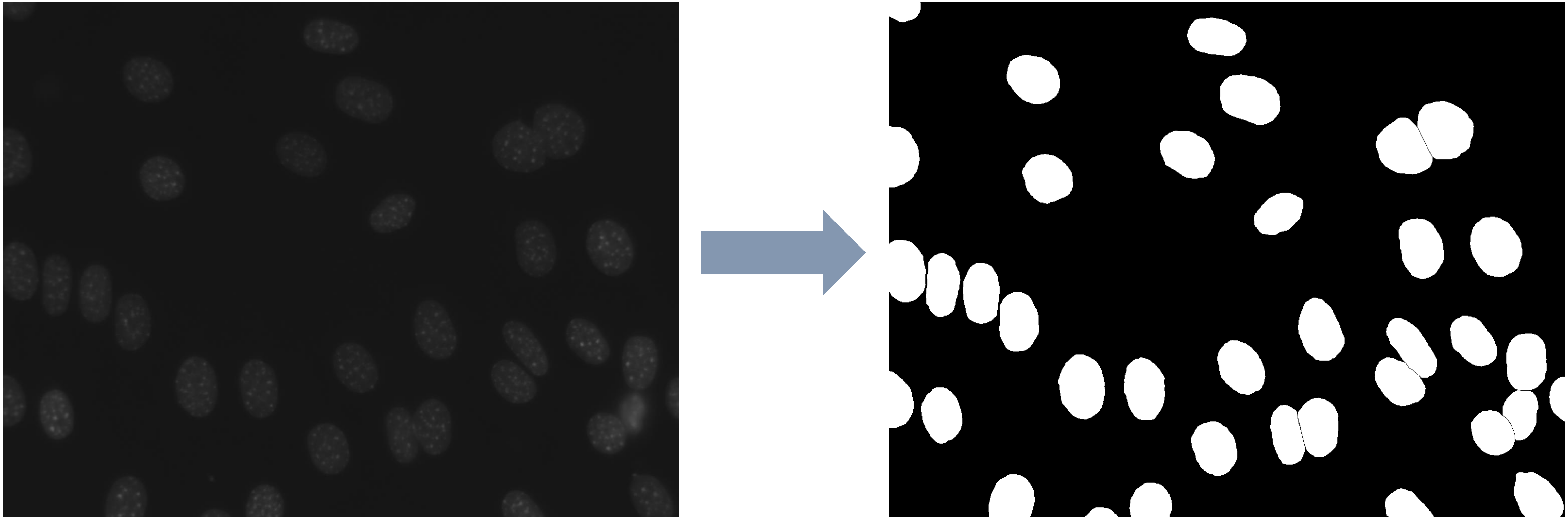
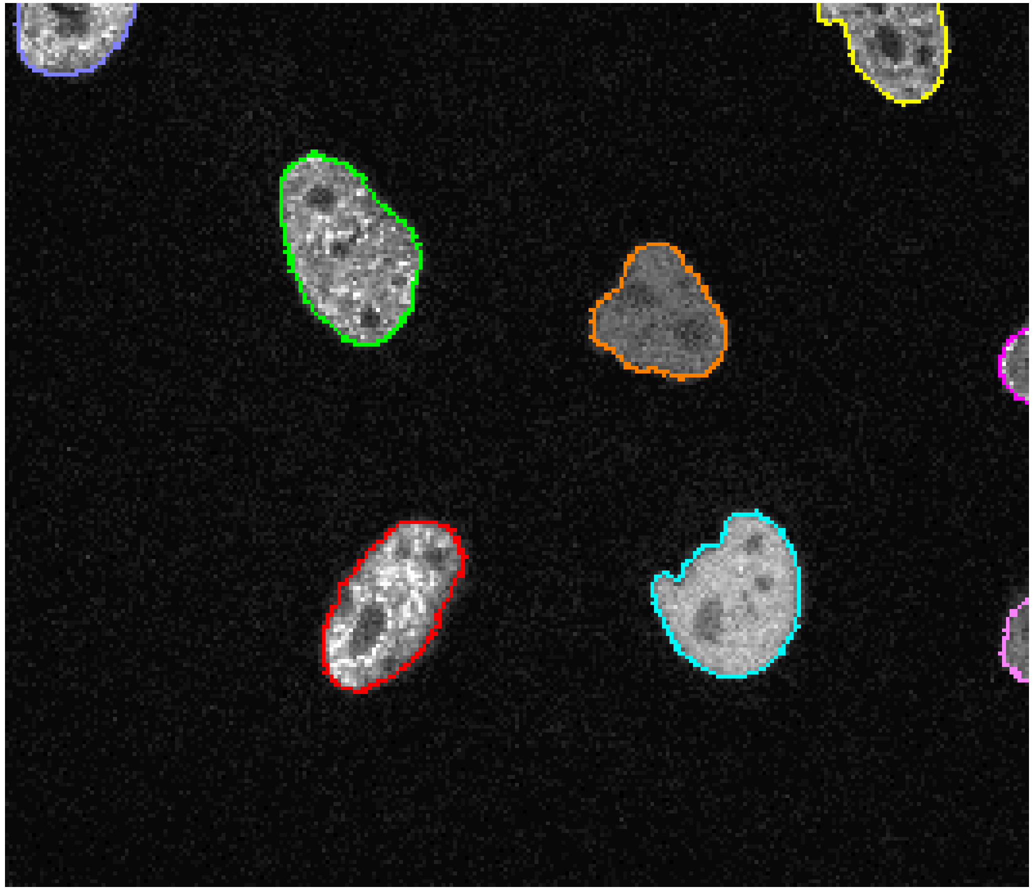
Image segmentation



Applications

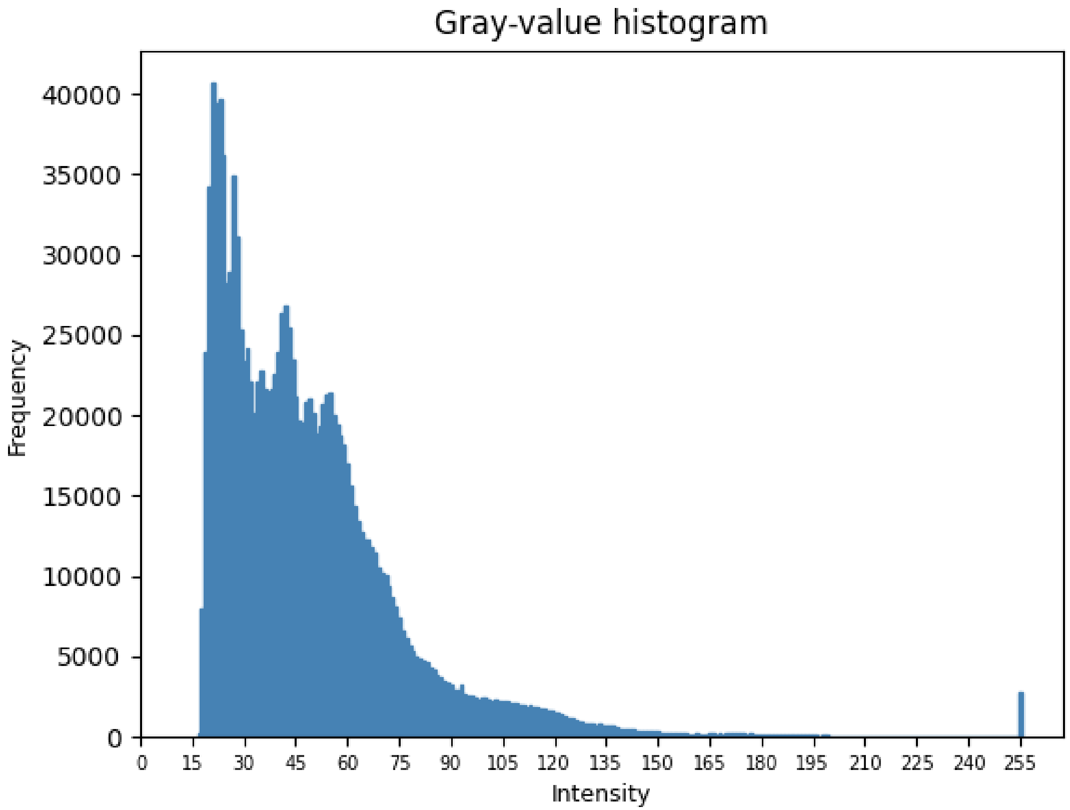
- High-throughput cytometry
  - Cell size
  - Cell counting
  - Cell-cycle determination

- Cell tracking



Methods

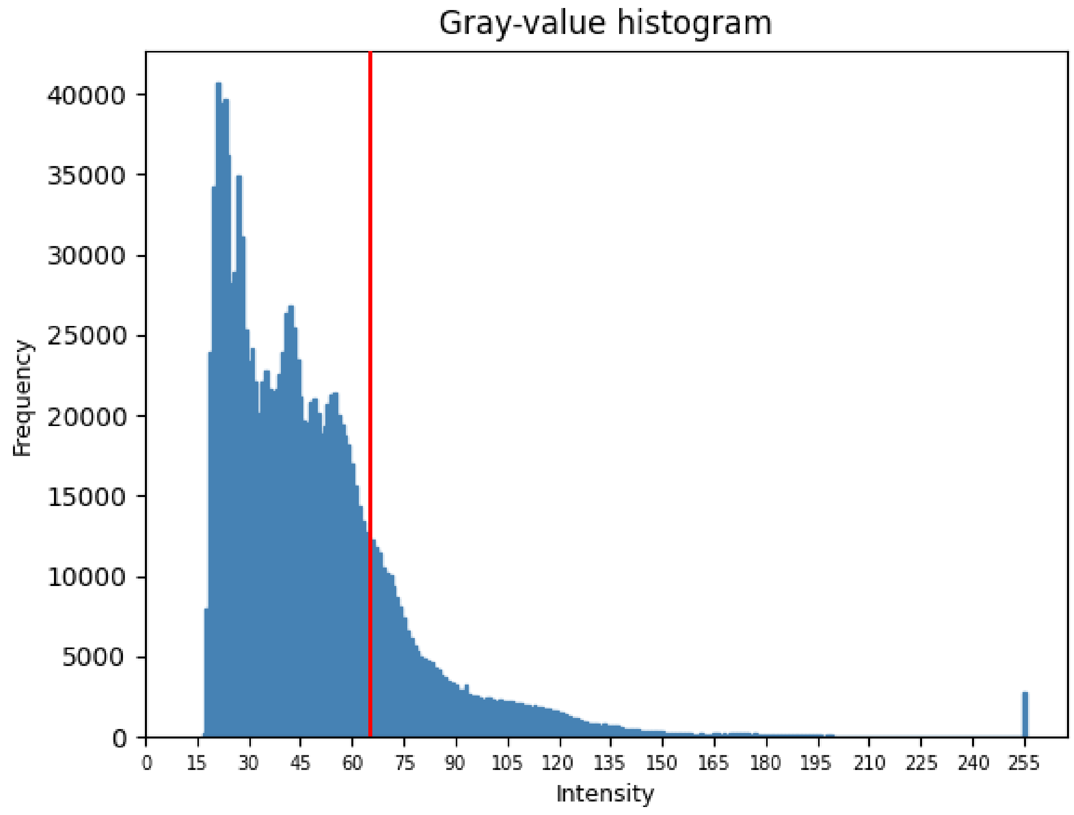
- Thresholding → Intensity clipping
- Region growing
- Machine learning
- etc.



Methods

- Thresholding → Intensity clipping

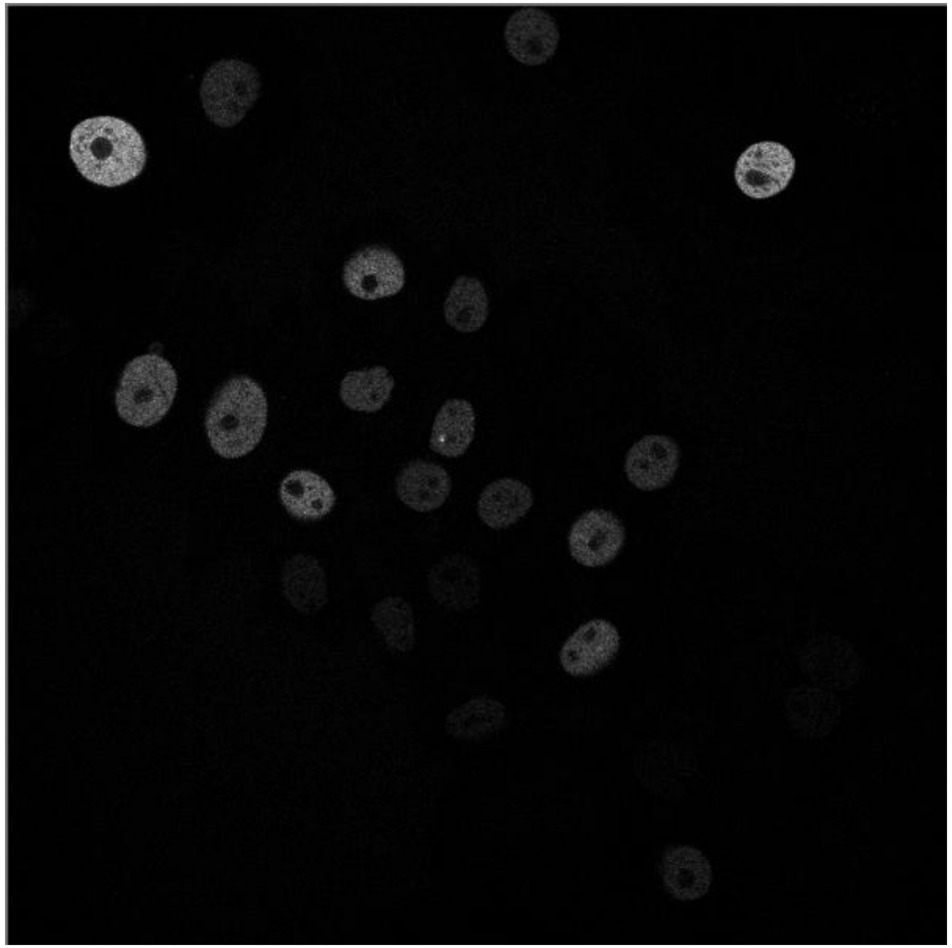
Otsu thresholding



Procedure

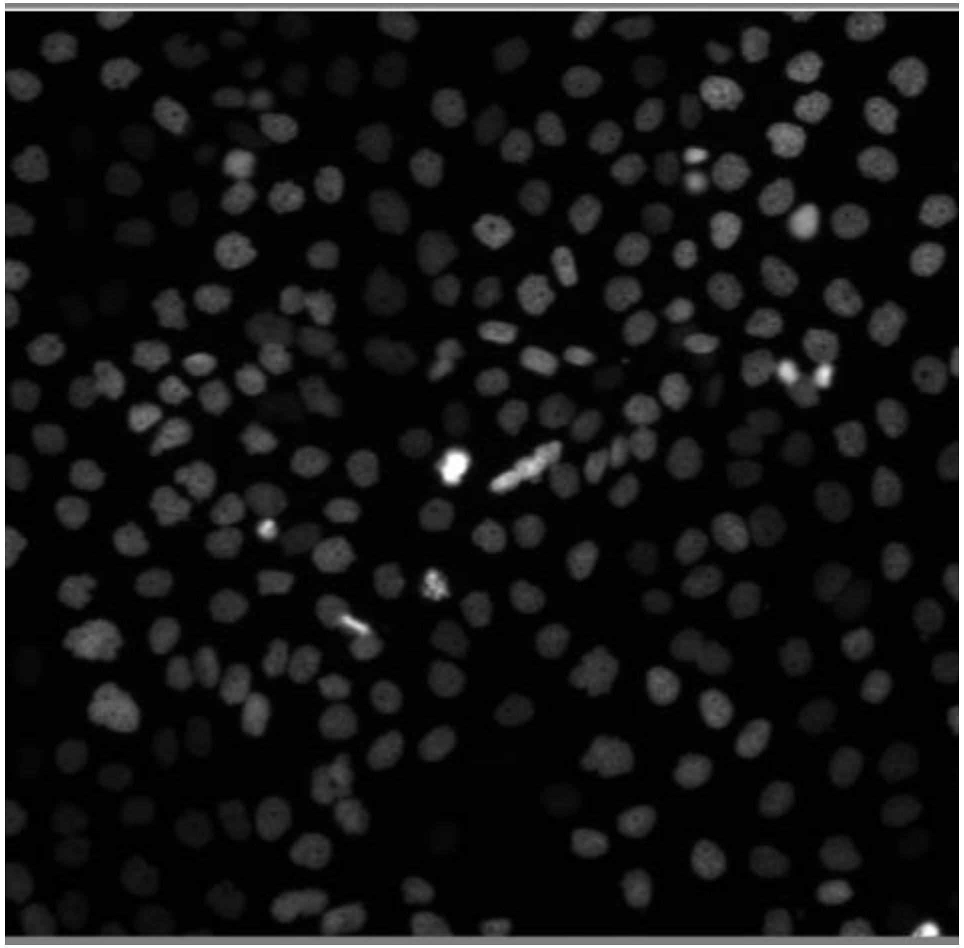

Input
Pre-processing
Otsu Thresholding
Dice score
Comparison

Input 1: N2DH-GOWT1 cells



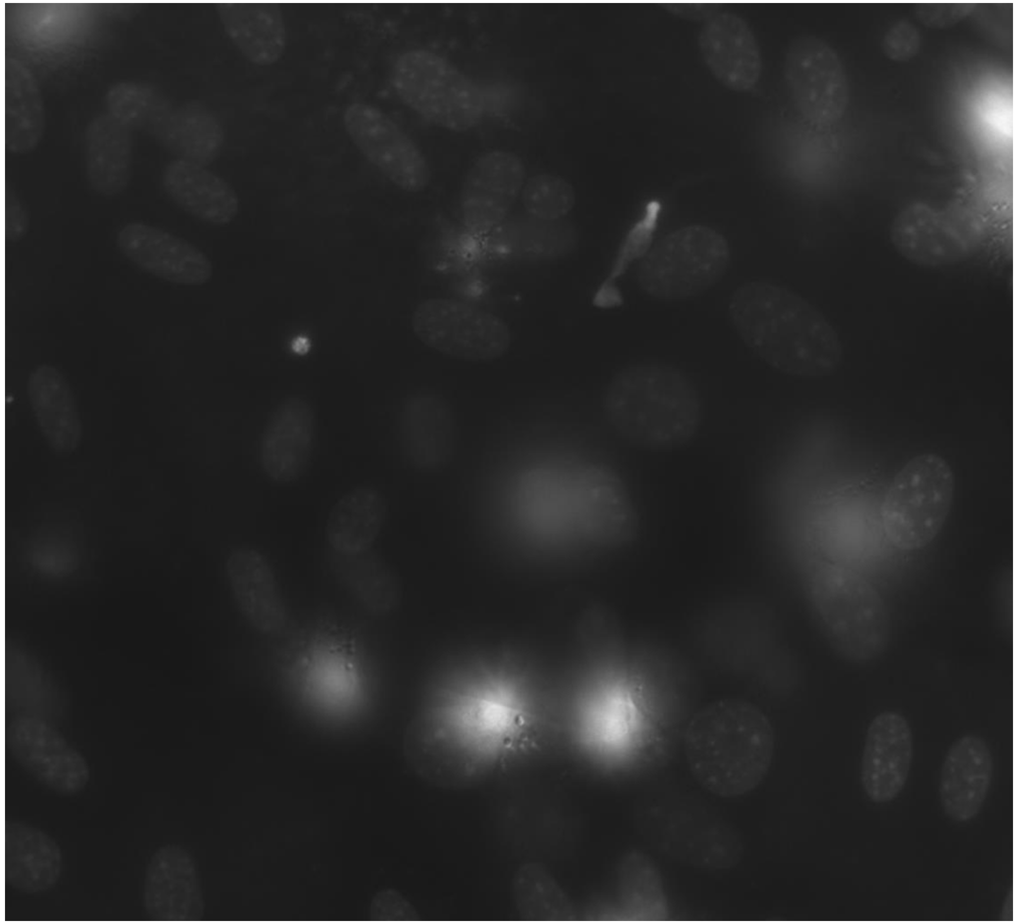
- GFP-Gowt1 mouse embryonic stem cells
- Time-laps confocal microscopy and GFP-staining
- Investigate genomic integrity of the cells
- Challenges:
  - Brightness of cells varies
  - Some cells hardly visible

Input 2: N2DL-HeLa cells



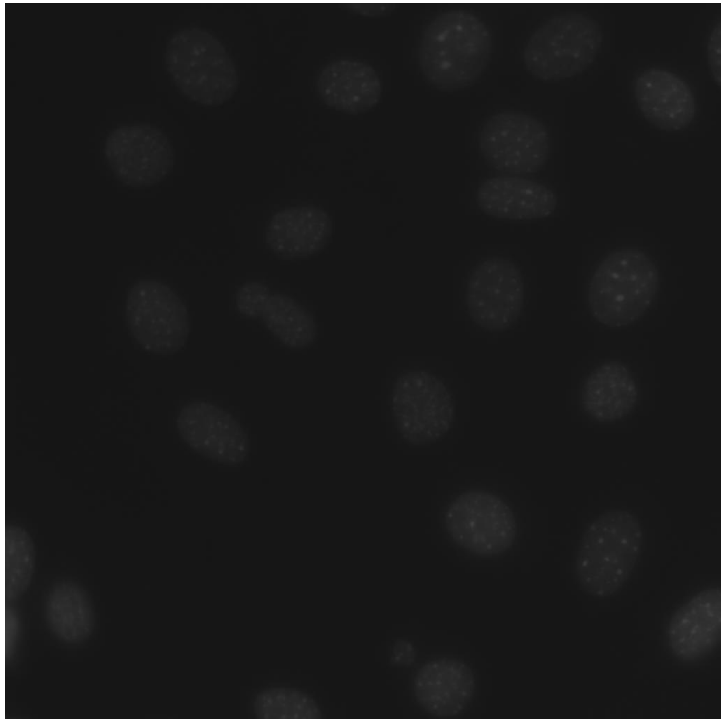
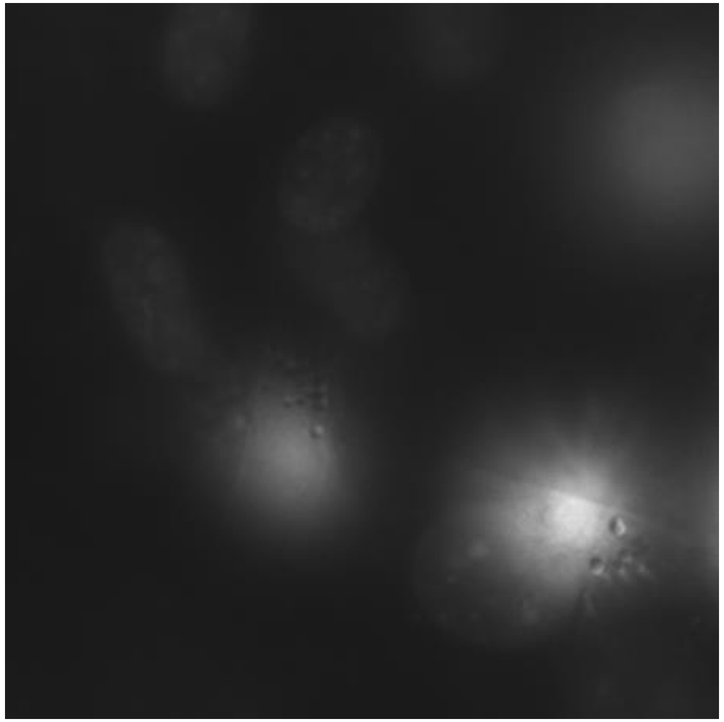
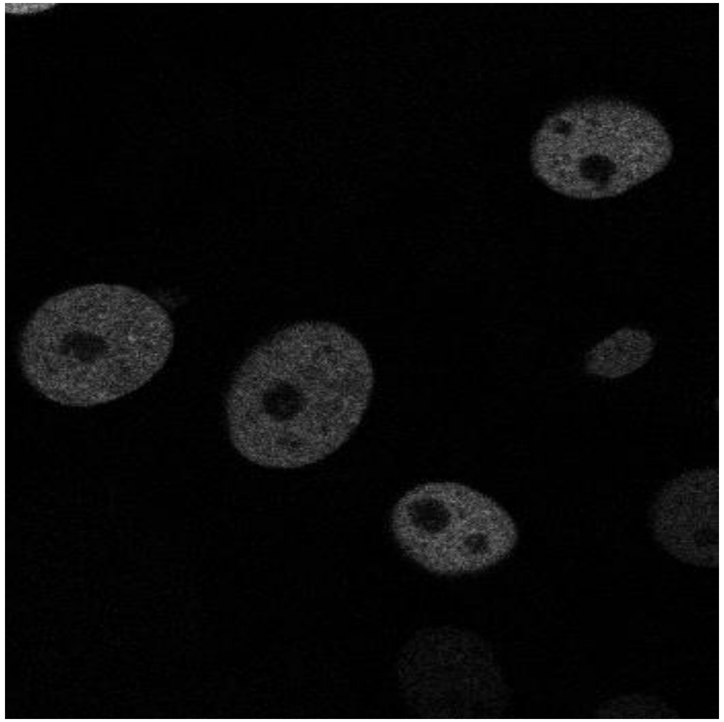
- human epithelial cells of cervical cancer
- live imaging of fluorescently labelled chromosomes
- Phenotypic profiling of the human genome
- Challenges:
  - Some cells hardly can be seen
  - Not easy to distinguish background and cells
  - Not much contrast

Input 3: NIH3T3 cells



- Several mouse embryonic fibroblast cells
- Fluorescence microscopy images
- Evaluation of image analysis pipelines
- Challenges:
  - Visible debris: light spots
  - Nuclei vary in brightness

Problems

Low contrast	Reflections	Random noise
		

Preprocessing

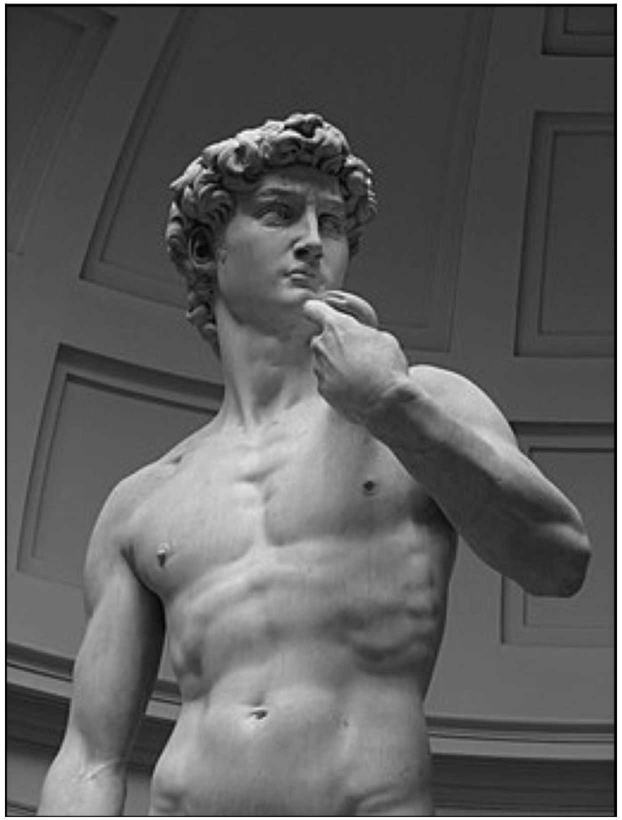
Solutions:

- Random noise

→ **Gauss filter, median filter**

- Reflections

- Low contrast



Original

Gauss filter (σ = 3)

Preprocessing

Solutions:

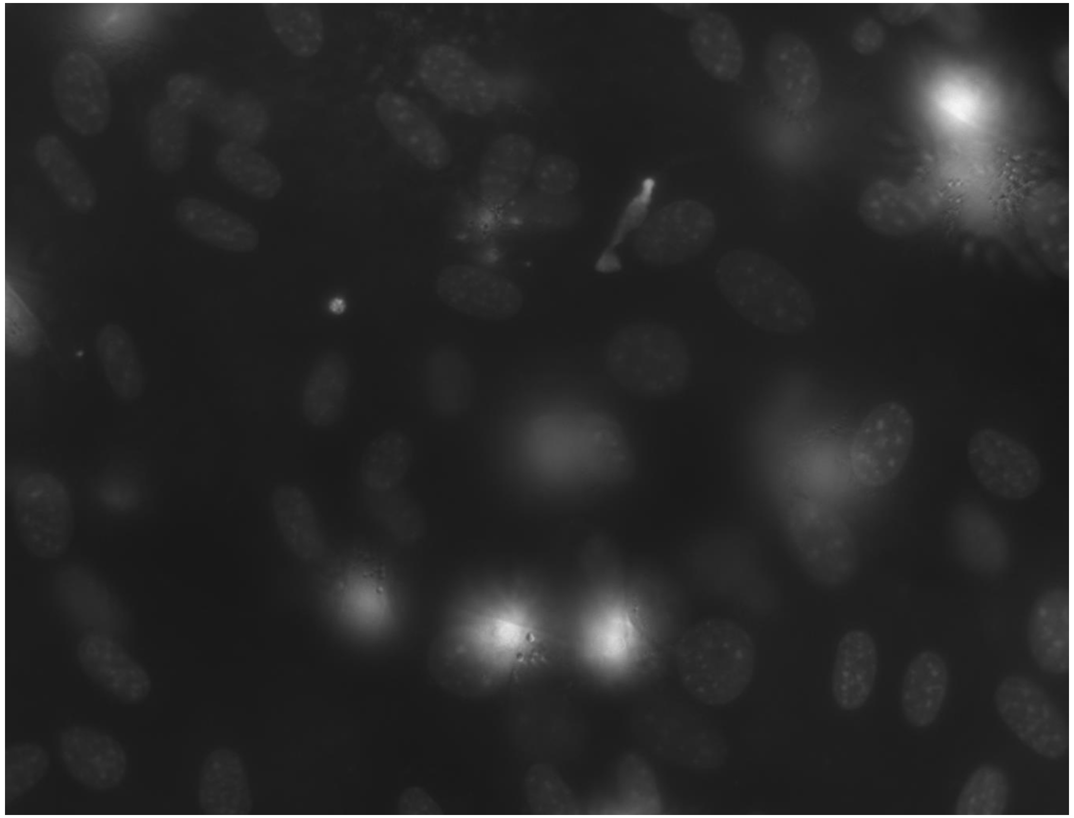
- Random noise

→ **Gauss filter, median filter**

- Reflections

→ **Thresholding**

- Low contrast



Preprocessing

Solutions:

- Random noise

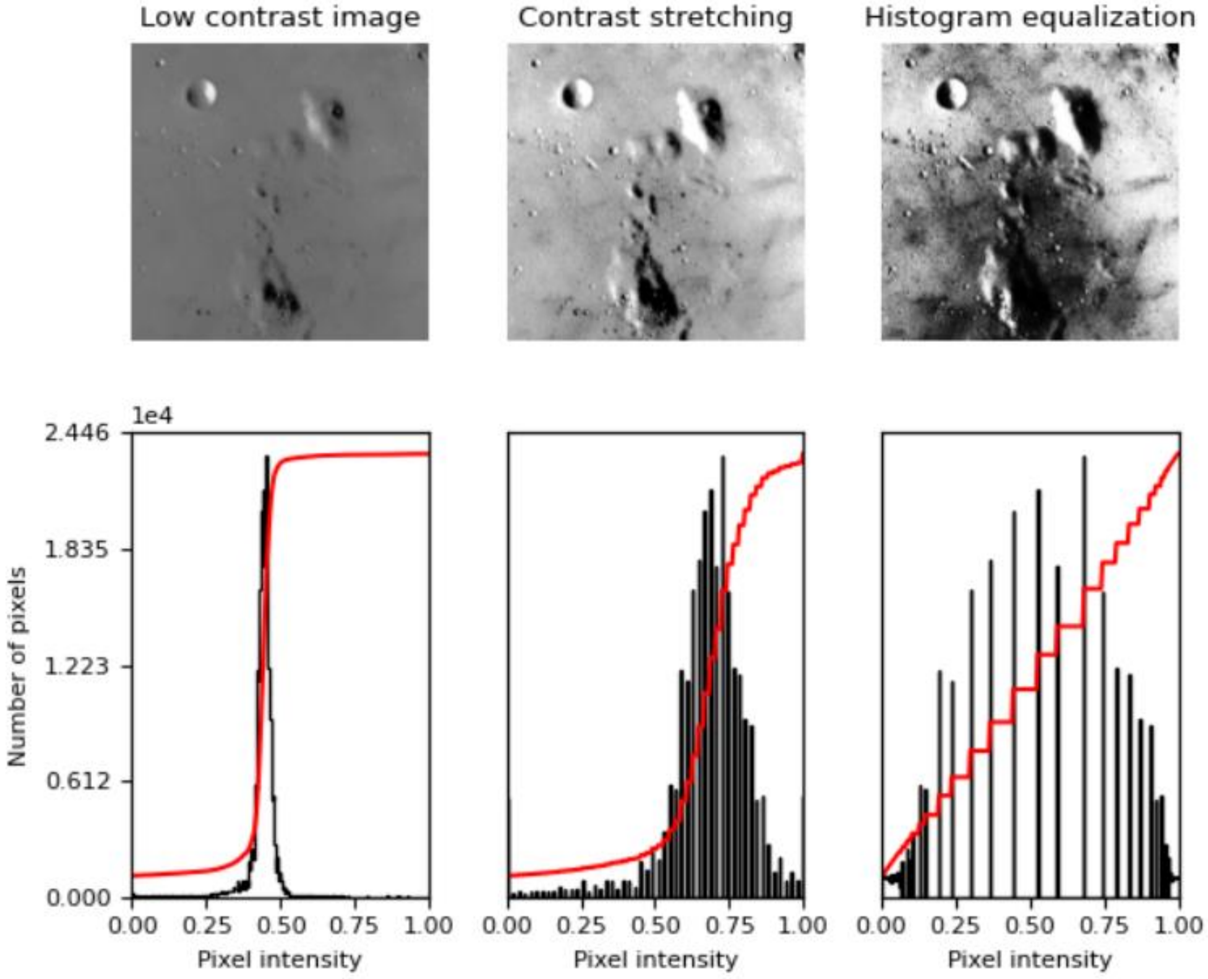
→ **Gauss filter, median filter**

- Reflections

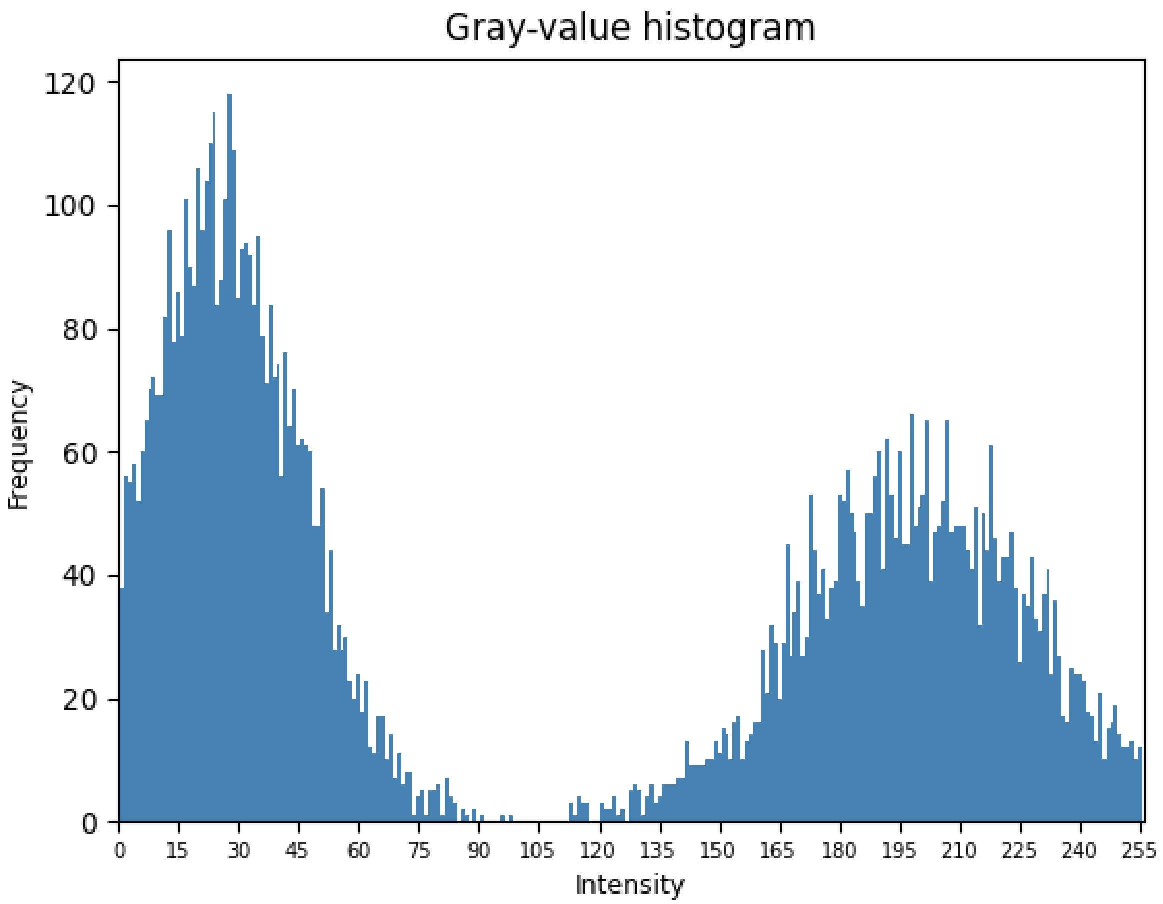
→ **Thresholding**

- Low contrast

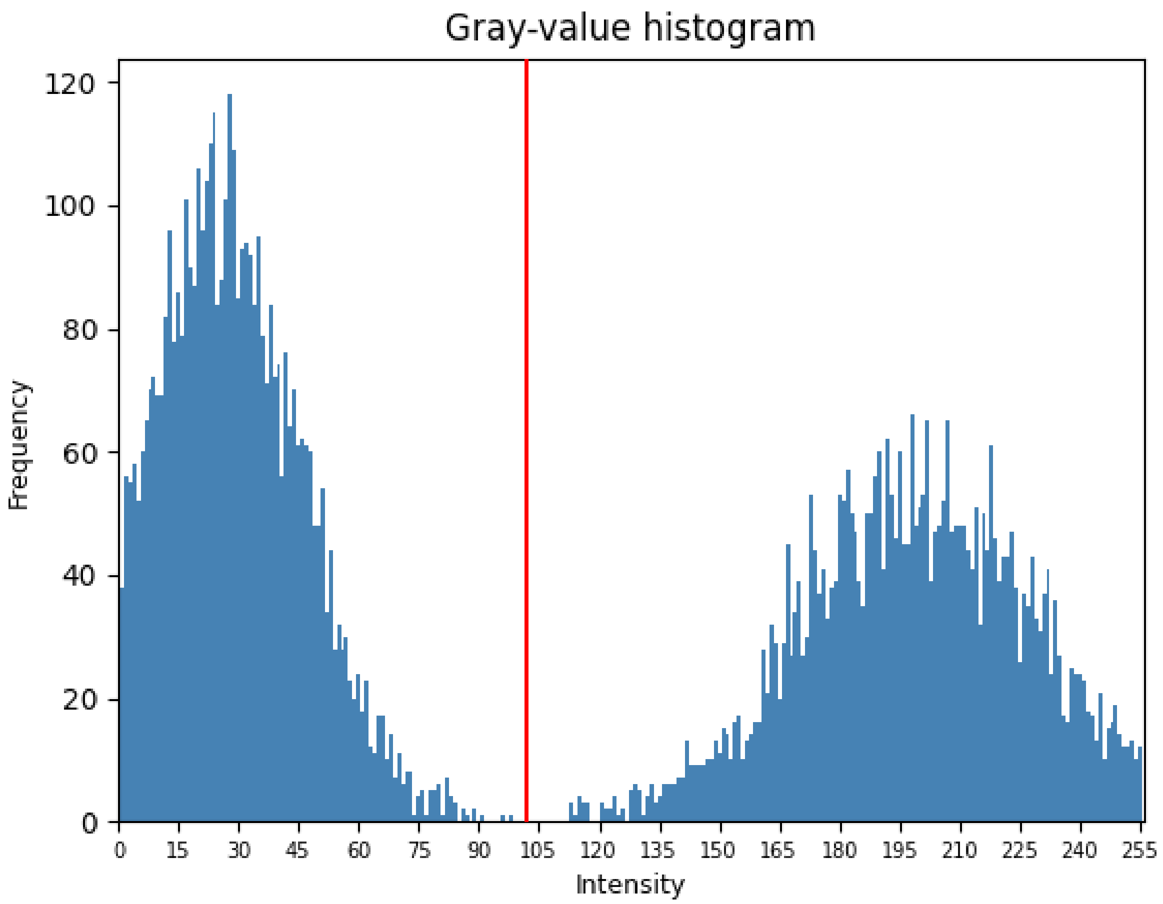
→ **Histogram equalization**



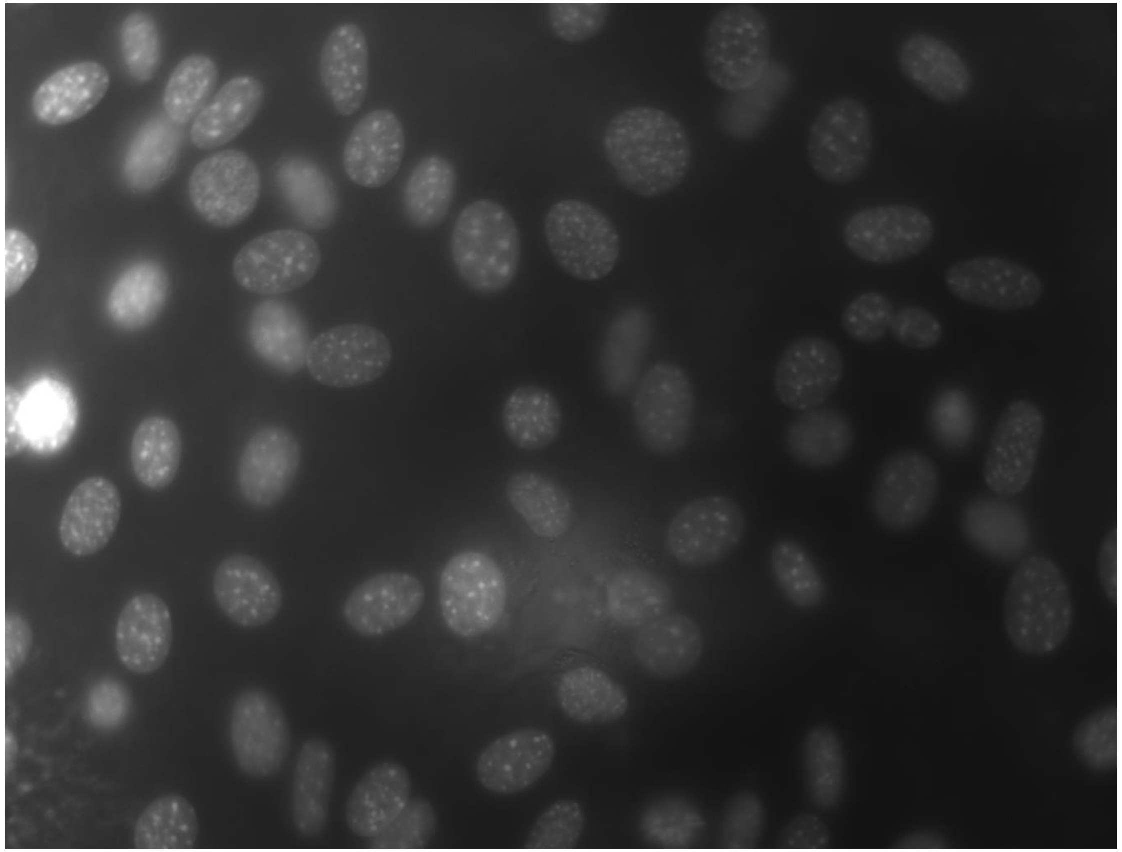
Otsu Thresholding

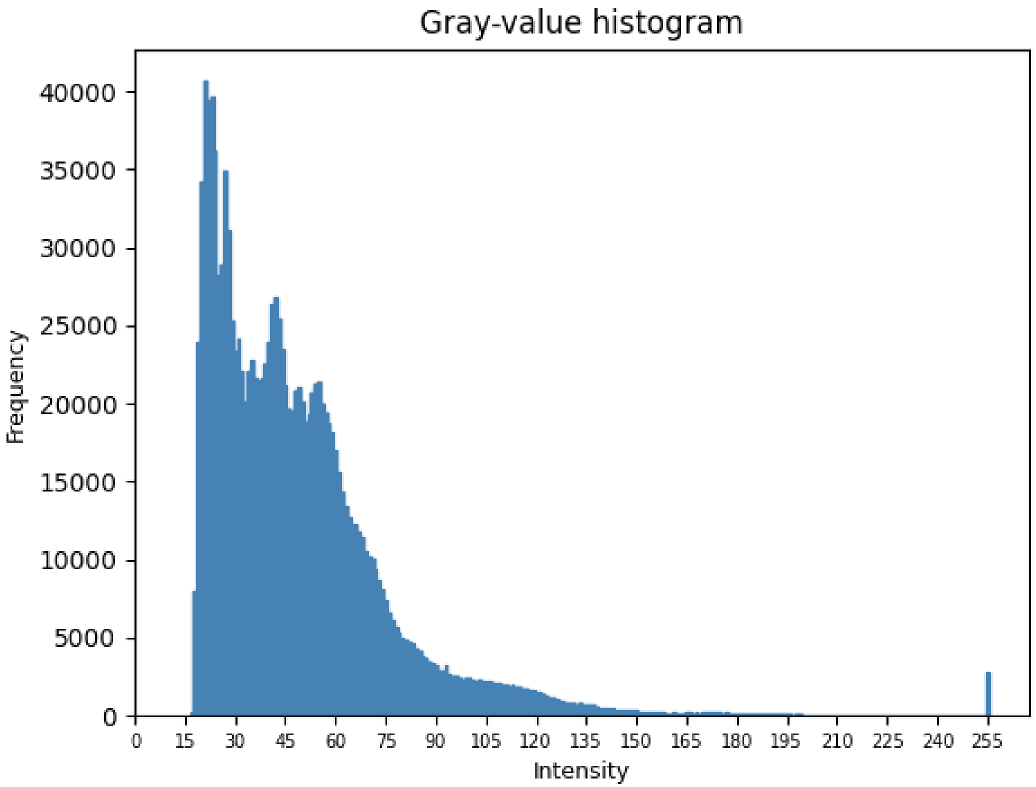


Otsu Thresholding



Otsu Thresholding





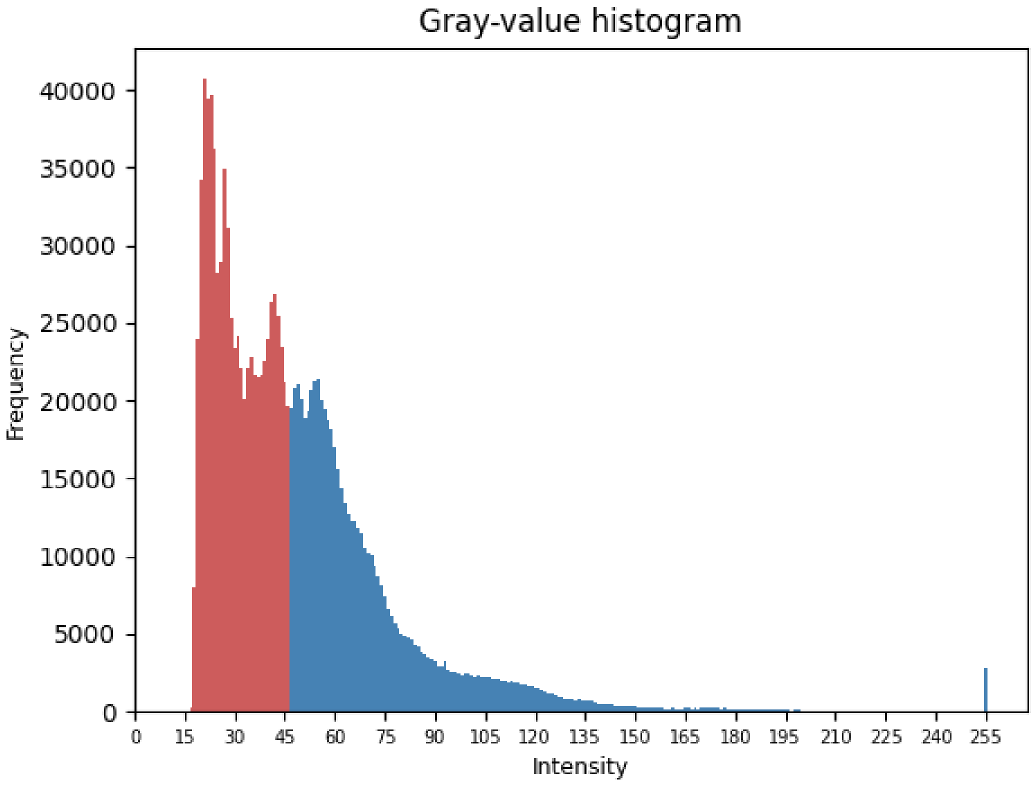
Threshold value k $\in [0,255]$

Otsu Thresholding

Between-class variance

$$\sigma_B = \omega_0 \omega_1 (\mu_1 - \mu_0)^2$$

$\omega_{0,1}$ = probability of class occurrence
$\mu_{0,1}$ = mean intensity values



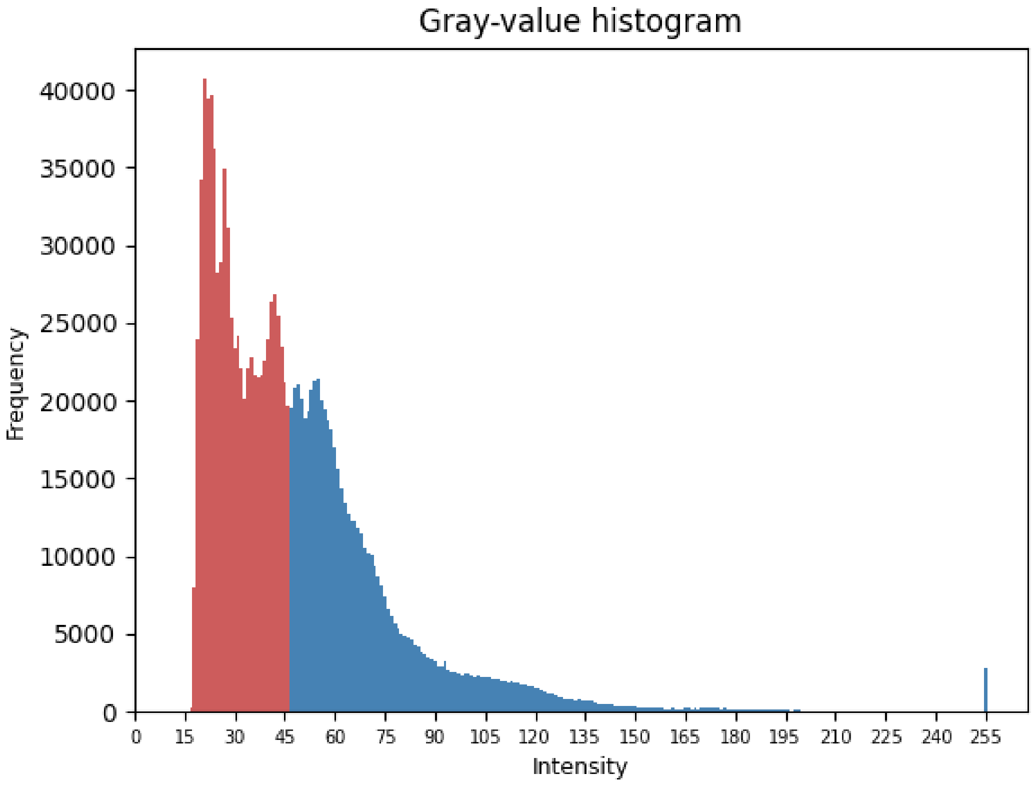
Threshold value k $\in [0,255]$

Otsu Thresholding

Criterion measure

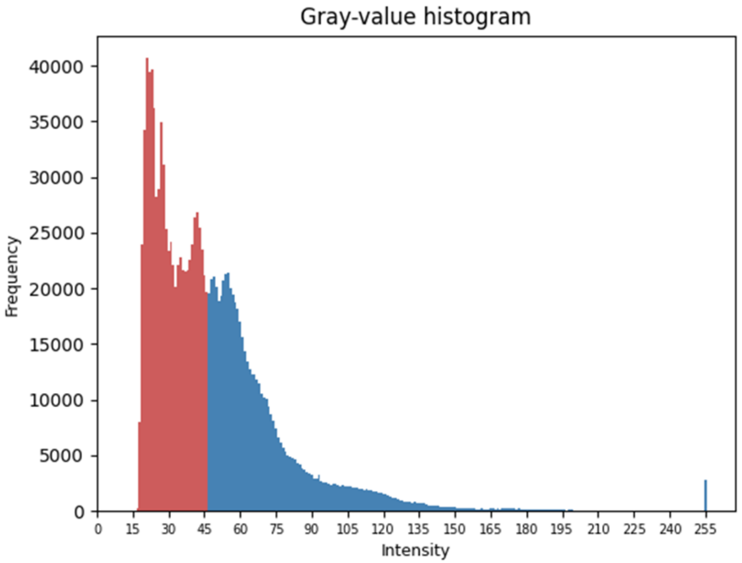
$$\eta(k) = \frac{{\sigma_B}^2(k)}{{\sigma_T}^2}$$

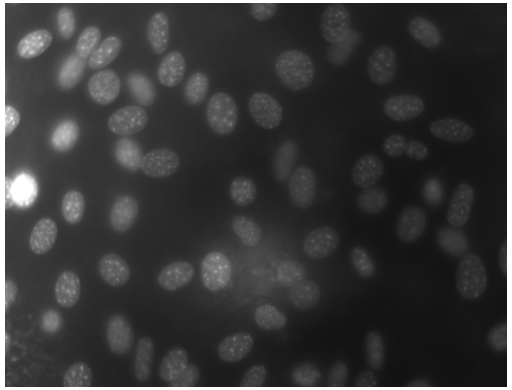
σ_B = between-class variance
σ_T = total variance
$\eta(k) \in [0,1]$



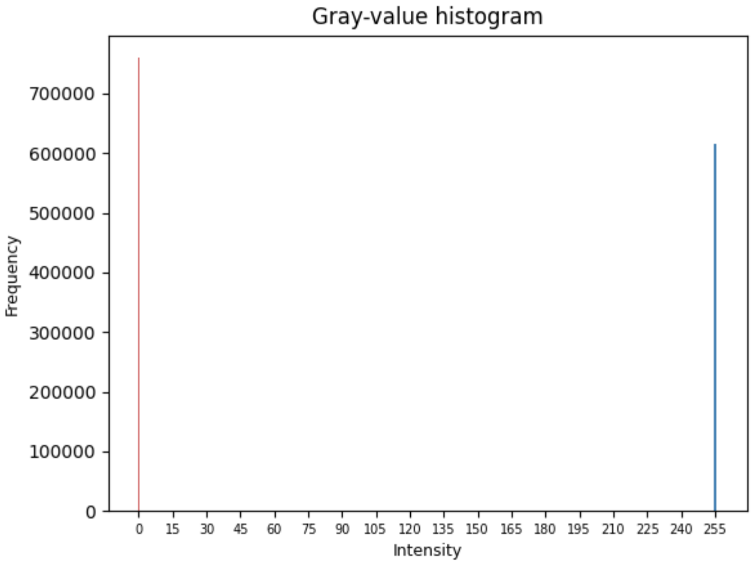
Threshold value k $\in [0,255]$

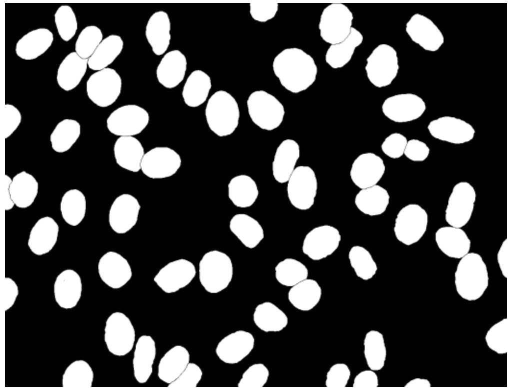
Image Clipping



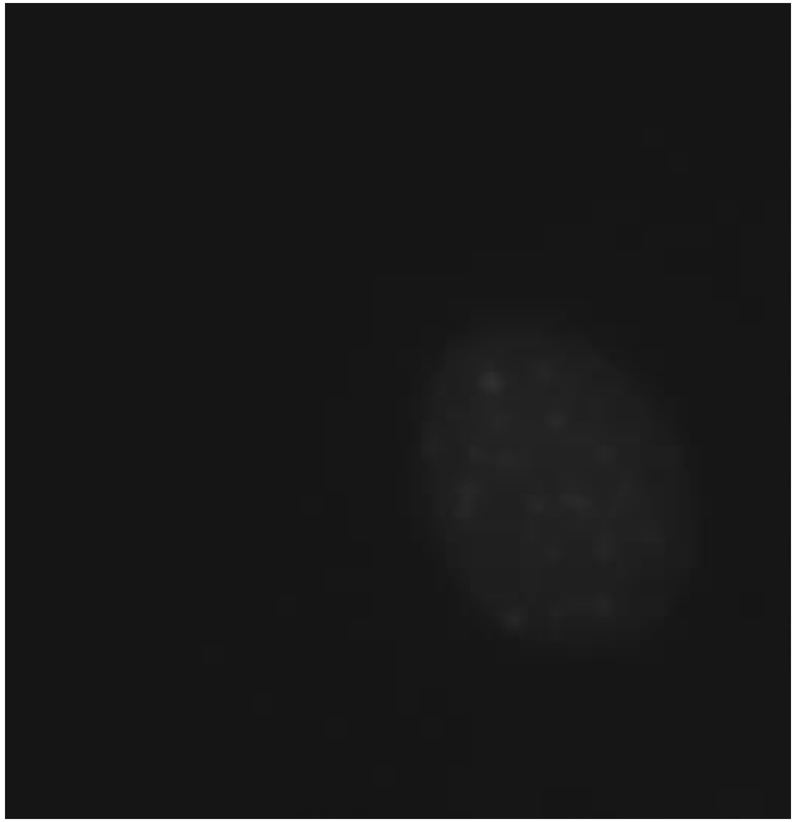


$$g_{\text{clip}}(x,y) = \begin{cases} 0 & \text{if } g(x,y) \leq k \\ 255 & \text{if } g(x,y) > k \end{cases}$$

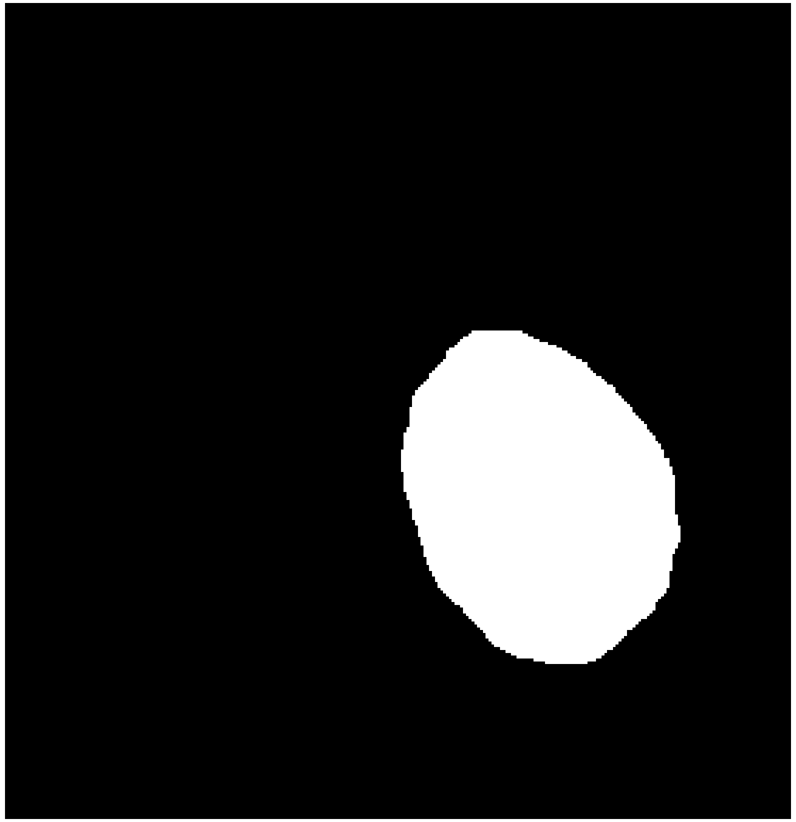




Implementation of the Dice Score



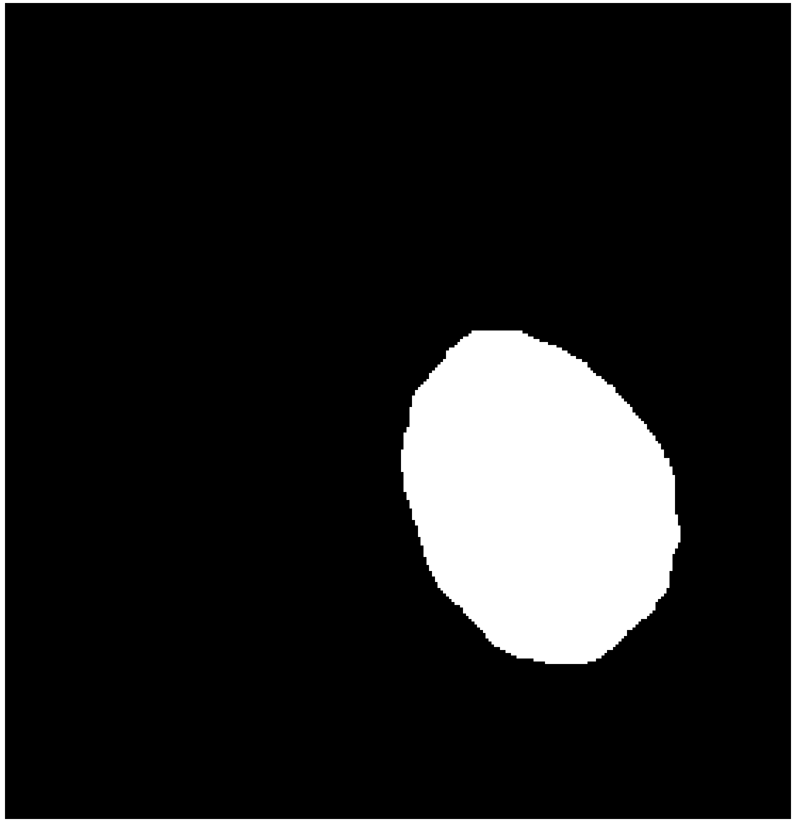
Original image



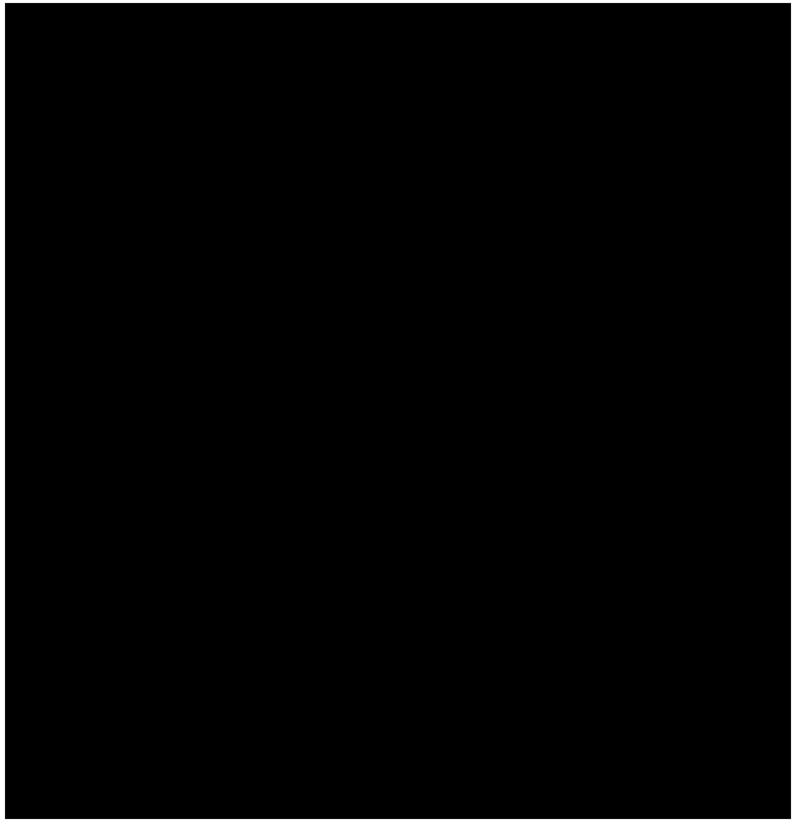
Ground truth

Implementation of the Dice Score

≈80 % accuracy



Ground truth



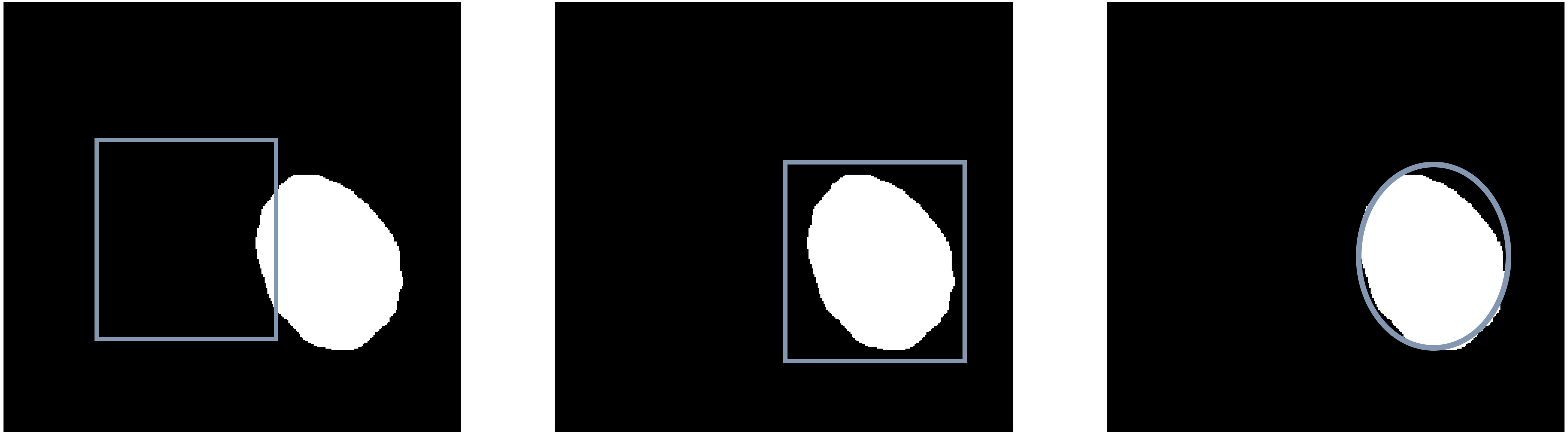
Prediction

Implementation of the Dice Score

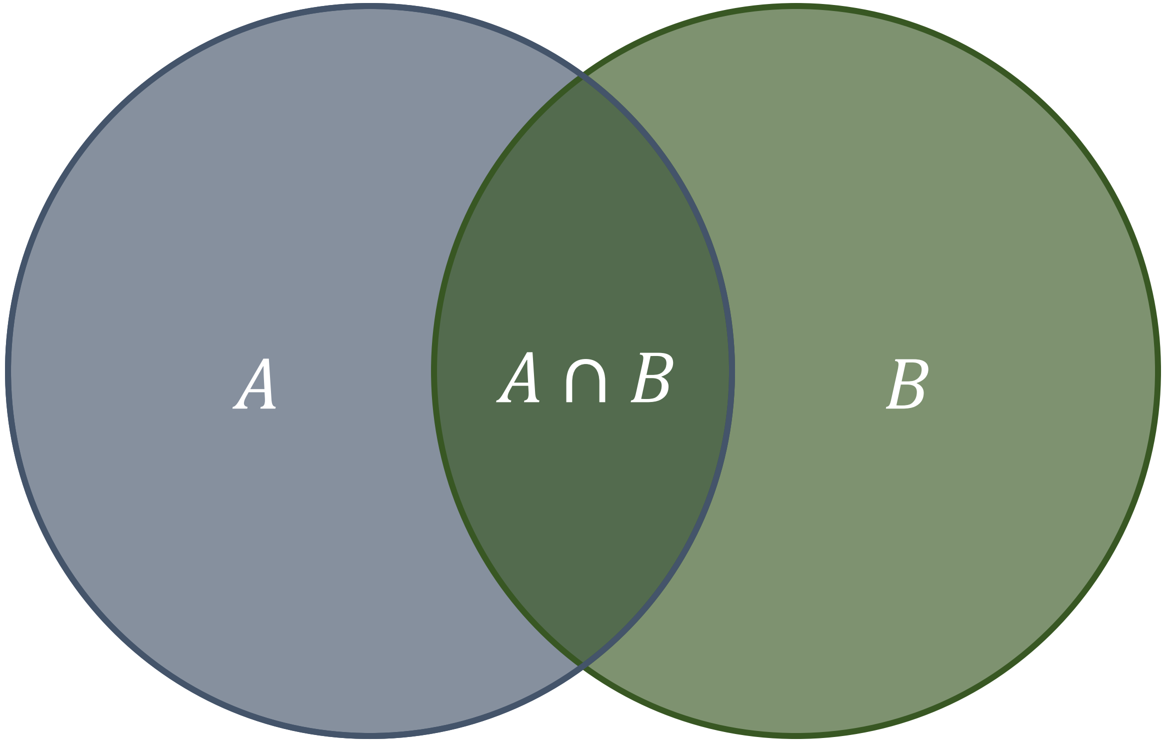
Bad prediction

Good prediction

Very good prediction

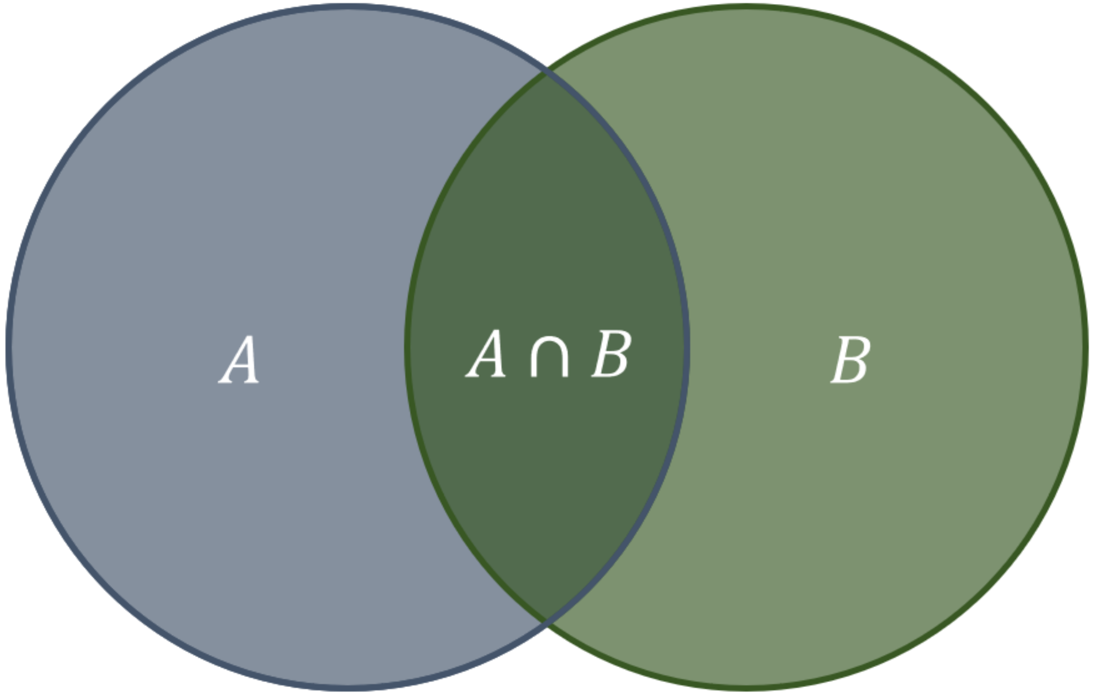


Implementation of the Dice Score



A: Predicted shape
B: Ground truth

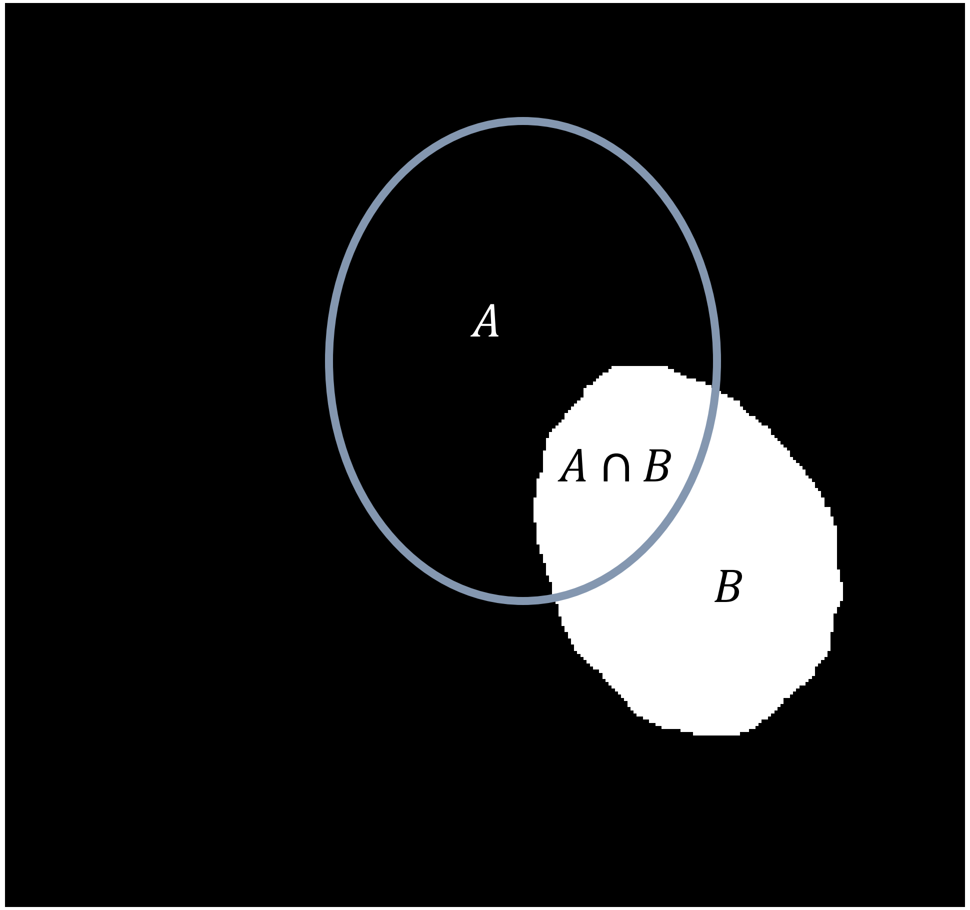
Implementation of the Dice Score



A: Predicted shape
B: Ground truth

$$DSC = \frac{2 \times A \cap B}{A + B}$$

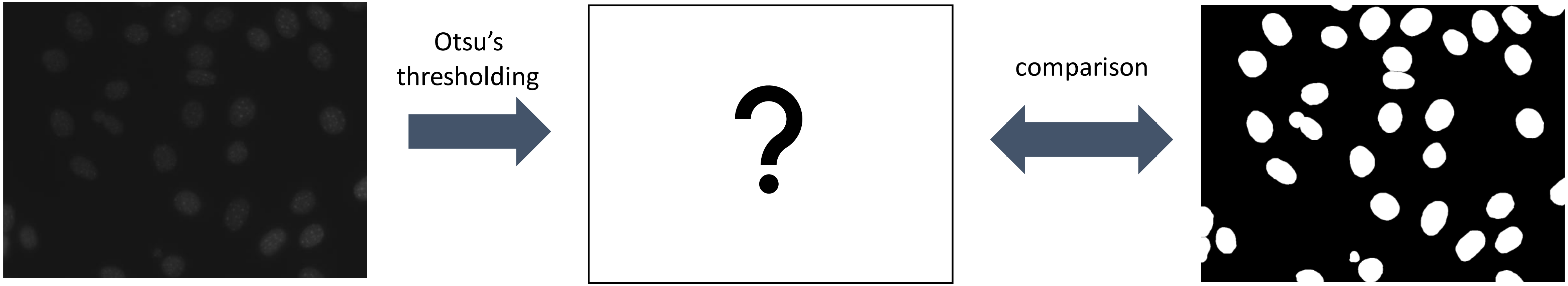
Implementation of the Dice Score



$$DSC = \frac{2 \times A \cap B}{A + B}$$

Implementation of the Dice Score

Our goal: compare ground truth images with our results



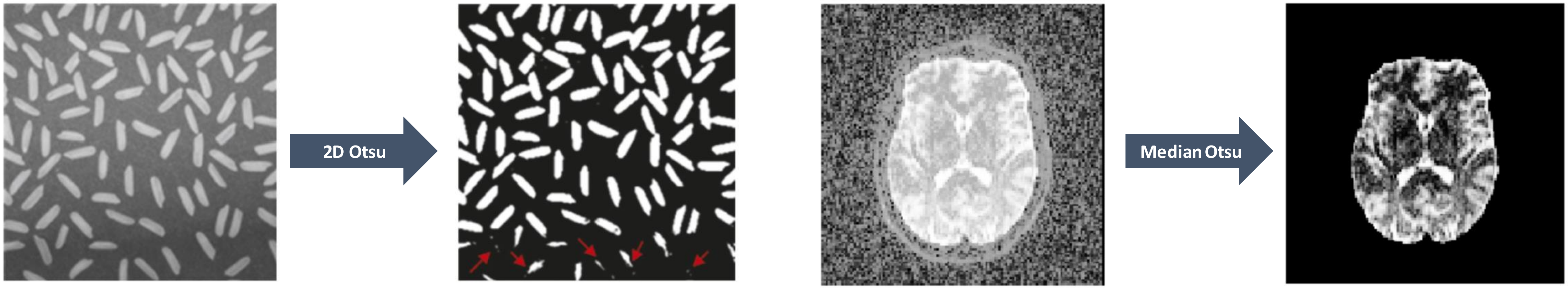
Original image

Our result

Ground truth

Further ideas

- 2D Otsu
- Median Otsu
- Algorithm for counting cells
- Algorithm for drawing cell trajectories



DIPY : Docs 0.16.0. - Brain segmentation with median_otsu (indiana.edu) and Jiangwa Xing, Pei Yang, Letu Qingge, "Robust 2D Otsu's Algorithm for Uneven Illumination Image Segmentation", *Computational Intelligence and Neuroscience*, vol. 2020, Article ID 5047976, 14 pages, 2020. https://doi.org/10.1155/2020/5047976

Timeline

Date	Milestone
Already done:	Researched Otsu thresholding and Dice scoring; Prepared project presentation
19.05.	Explore data with histograms and similar (All)
26.05. ✴	Code algorithms for Otsu thresholding (H, E) and Dice scoring (L, V)
02.06. ✴	Assemble the whole pipeline, test different preprocessing options (All)
09.06.	Research alternative evaluation methods
16.06. ✴	1. Test our pipeline on data from the BBBC 2. Implement alternative evaluation methods: • IoU, pixel accuracy (All) • Hausdorff metric (H, E) • NSD (L, V)
23.06.	Compare results with group 4.4 and 4.5
30.06. ✴	Complete report in Jupyter Notebook (All)
07.07. ✴	Complete final presentation (All)

Thank you for your attention!