

Cell nuclei segmentation

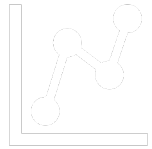
Implementation and evaluation of Otsu thresholding

Topic 01 Team 03
Franka Begall
Kira Effenhauser
Selina Geiger
Anna-Lena Schulz

Supervisor: PD Dr. Karl Rohr, Dr. Leonid
Kostykin, Kerem Celikay, Janis Meyer

Tutor: Hannah Winter

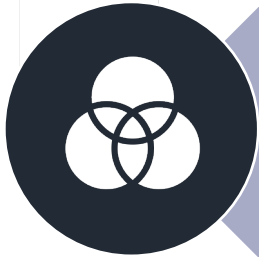
Software requirements



Otsu thresholding

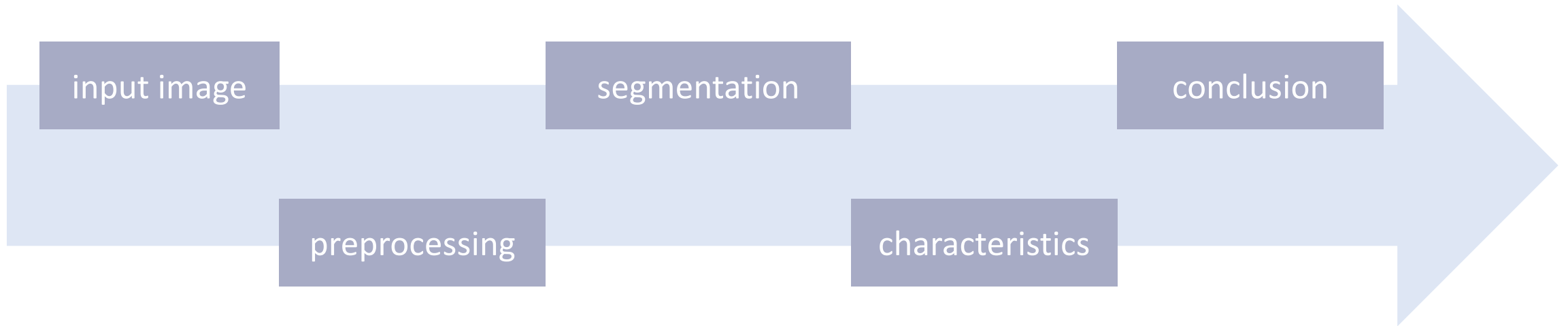


Local thresholding

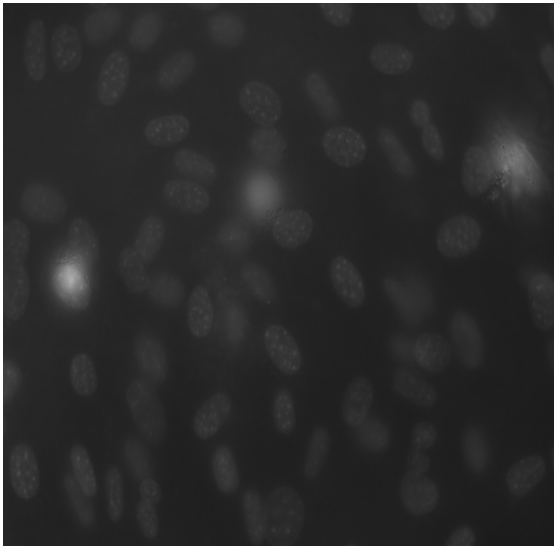
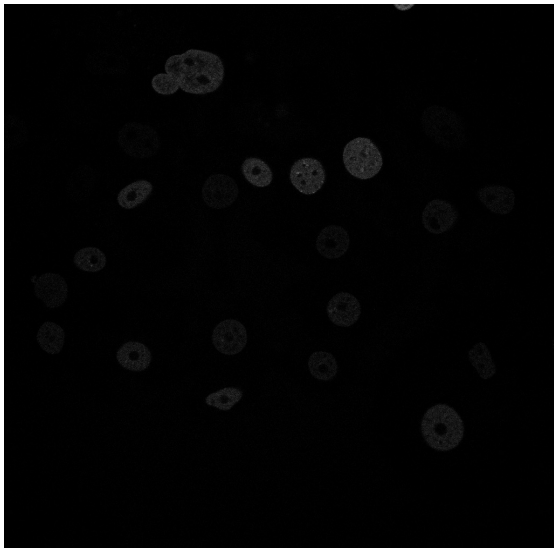


Dice score

Procedure



Datasets



	N2DH-GOWT1	N2DL-HeLa	NIH3T3
Organism	mouse (<i>Mus musculus</i>)	human (<i>homo sapiens</i>)	mouse (<i>Mus musculus</i>)
Cell type	embryonic stemm cells	epithelial cells of cervical cancer	embryonic fibroblast cells
Staining	Oct4-GFP	H2b-GFP	Hoechst
Microscope	timelapse confocal microscopy	Olympus lx81 microscope used for live imaging of fluorescently labeled chromosomes	fluorescence microscopy

Input image

Preprocessing

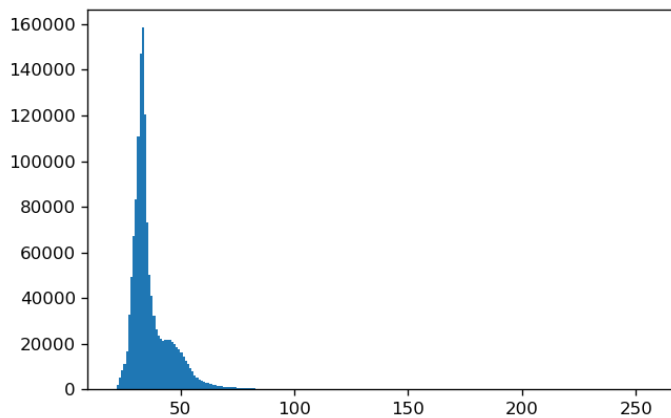
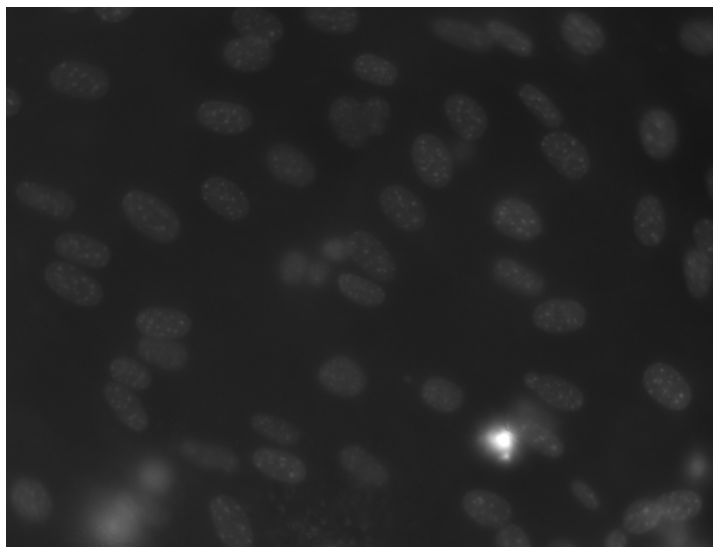
Segmentation

Characteristics

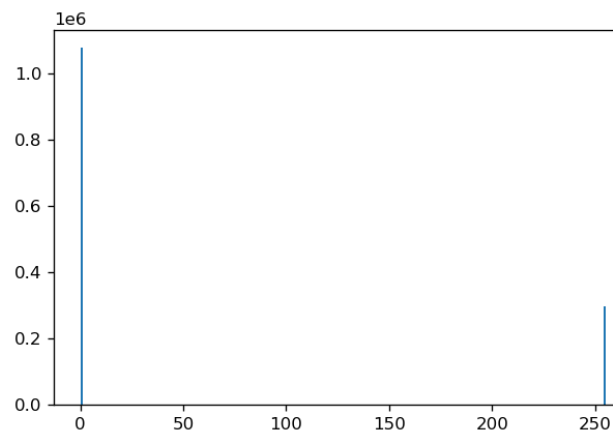
Conclusion

Image types

Image



Ground Truth



Input image

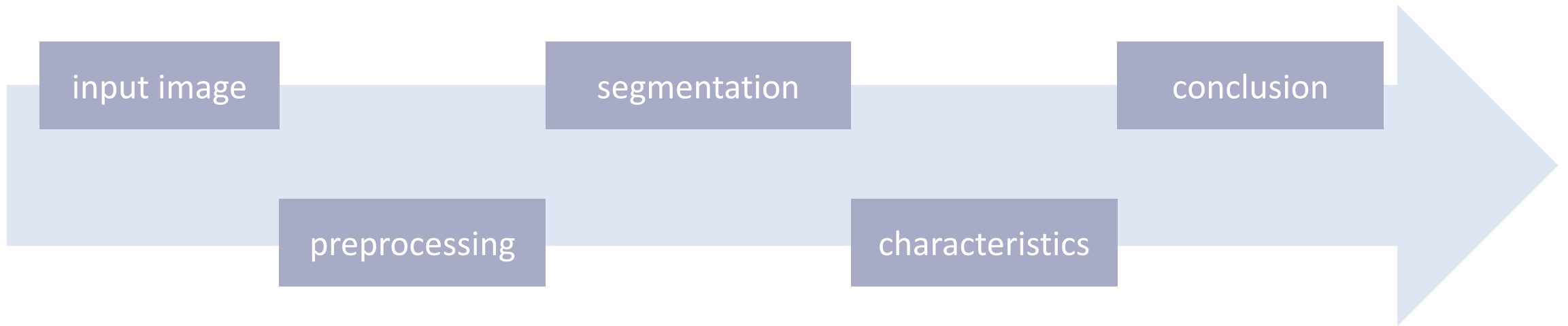
Preprocessing

Segmentation

Characteristics

Conclusion

Procedure

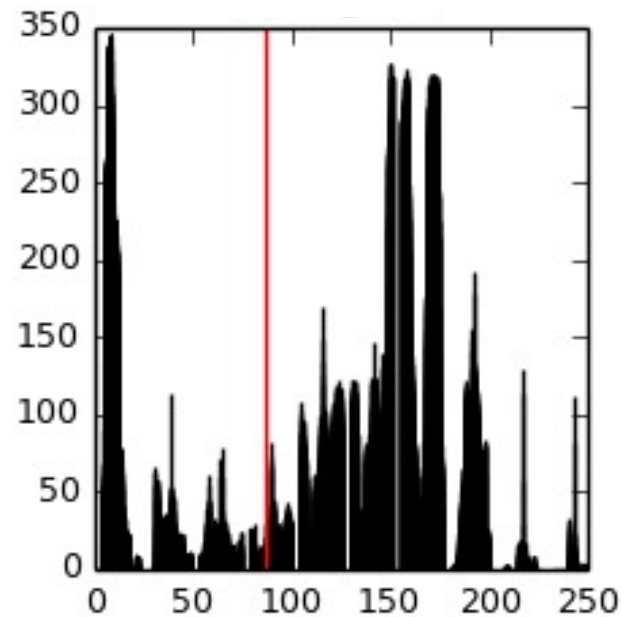


Otsu thresholding

what we need:



what we work with:



what we get:



http://sharky93.github.io/docs/gallery/auto_examples/plot_otsu.html

Input image

Preprocessing

Segmentation

Characteristics

Conclusion

Otsu thresholding

Determines a threshold automatically for image segmentation

minimize the within class variance

$$\sigma_w^2(t) = w_1(t)\sigma_1^2(t) + w_2(t)\sigma_2^2(t)$$

σ : variance of class

w : probabilities of the two classes divided by a threshold

t : threshold

maximize the between class variance

$$\sigma_b^2(t) = w_1(t)w_2(t)[\mu_1(t) - \mu_2(t)]^2$$

σ : variance of class

w : probabilities of the two classes divided by a threshold

t : threshold

μ : mean of class

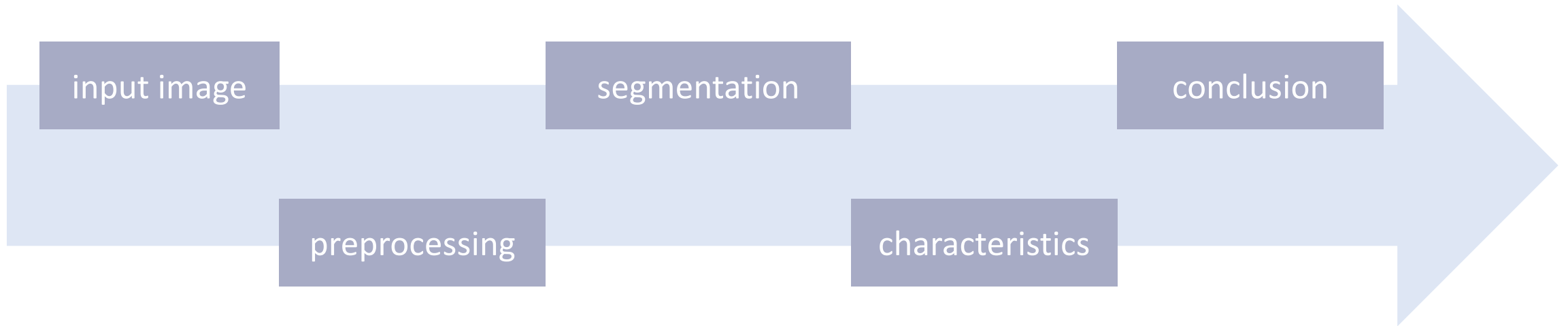
Otsu thresholding

- 1 Process the input grayscale image
- 2 Generate a distribution of the pixel intensities (histogram)
- 3 Calculate the threshold value t
- 4 Replace the pixels values with white ($>t$) and black ($<t$)

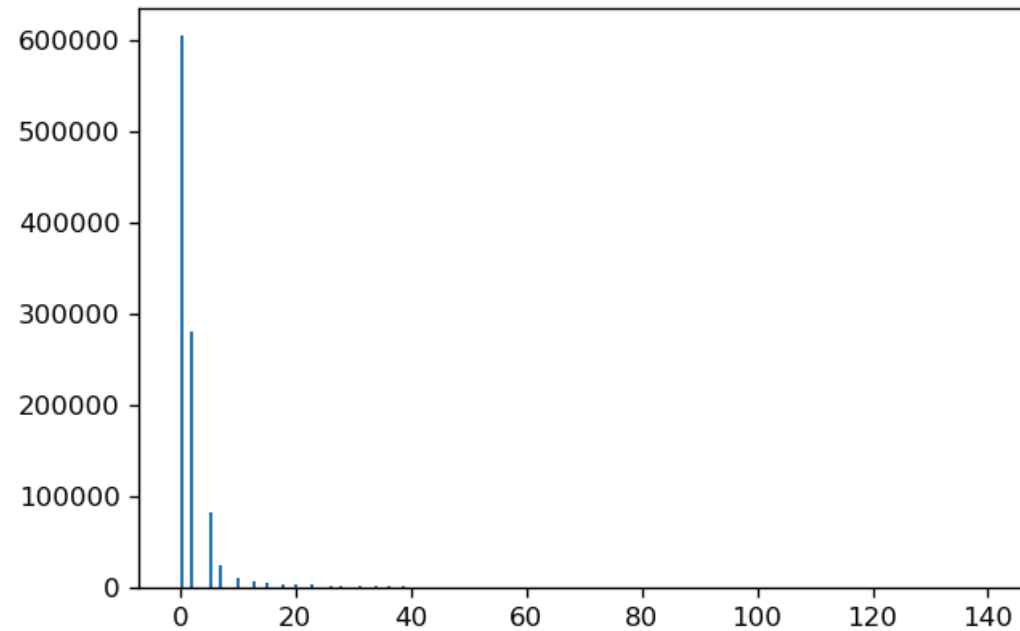
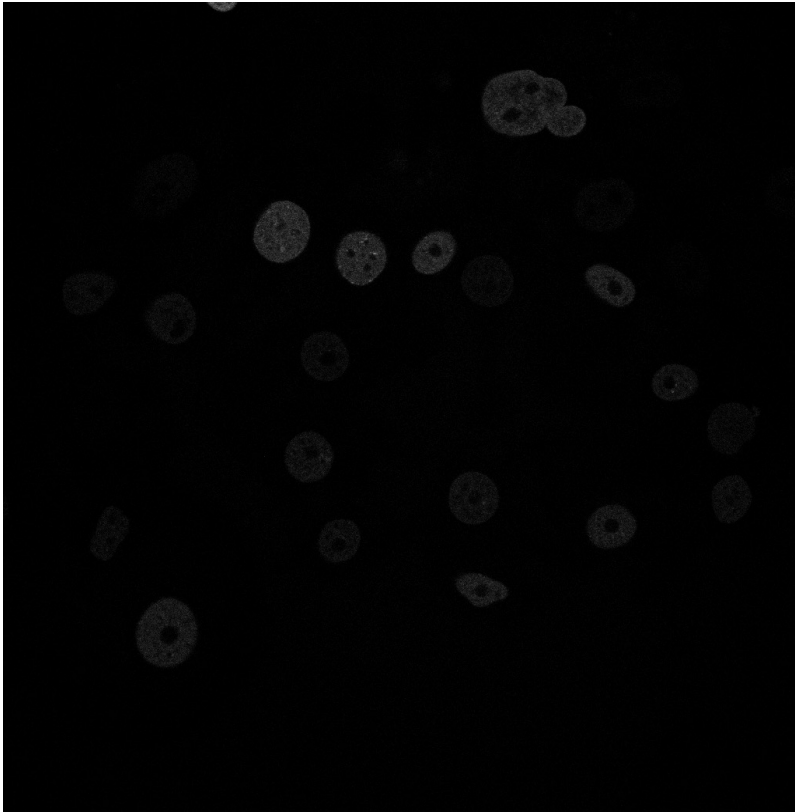
Conditions for optimal Otsu thresholding

- high contrast between objects and background
- bimodal distribution
- minimal image noise
- similar intensities of objects

Procedure

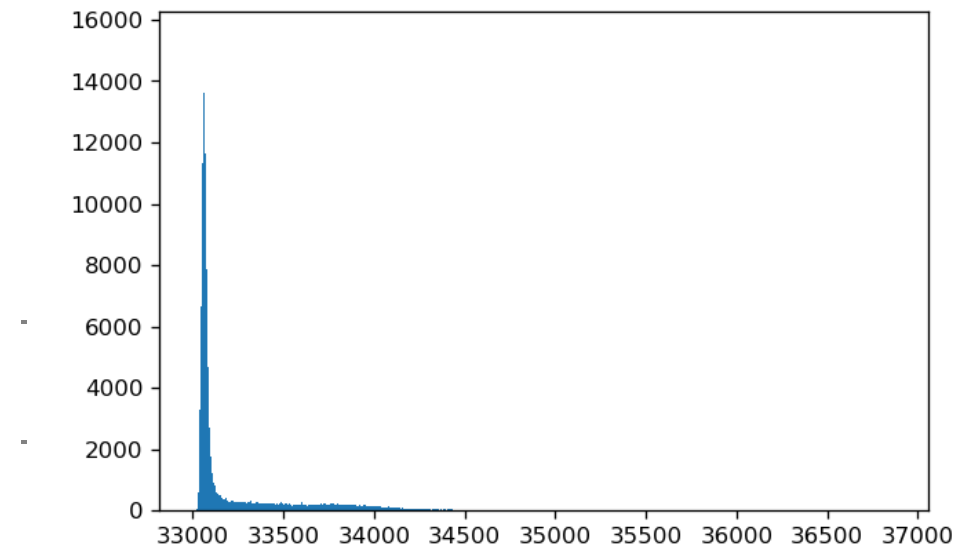


Challenges: N2DH-GOWT1



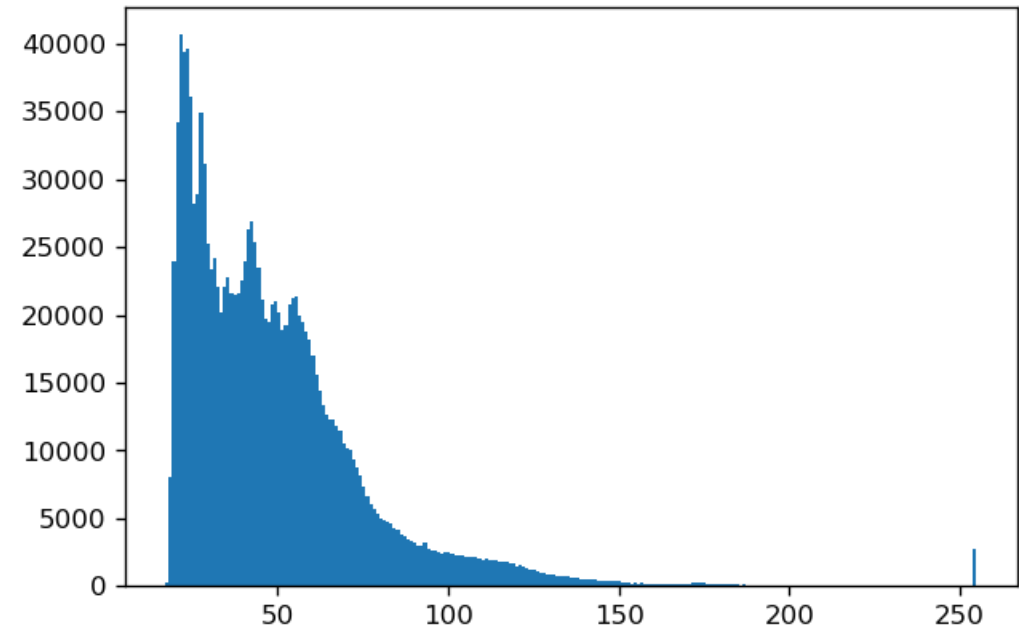
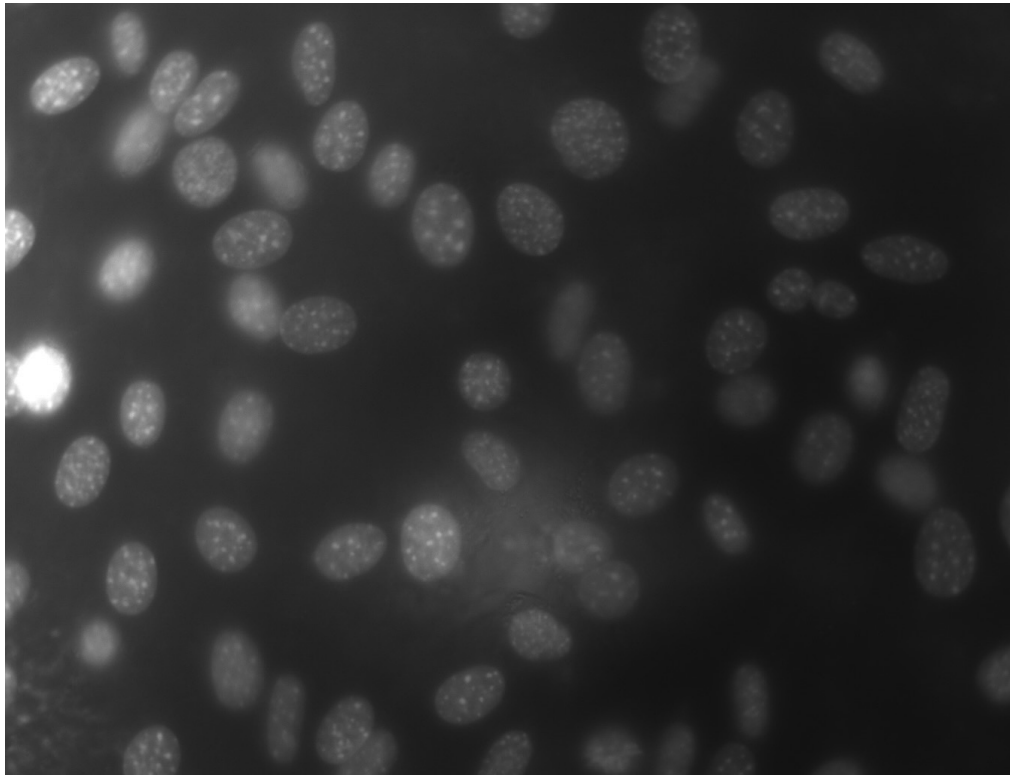
- noise
- low contrast

Challenges: N2DL-HeLa



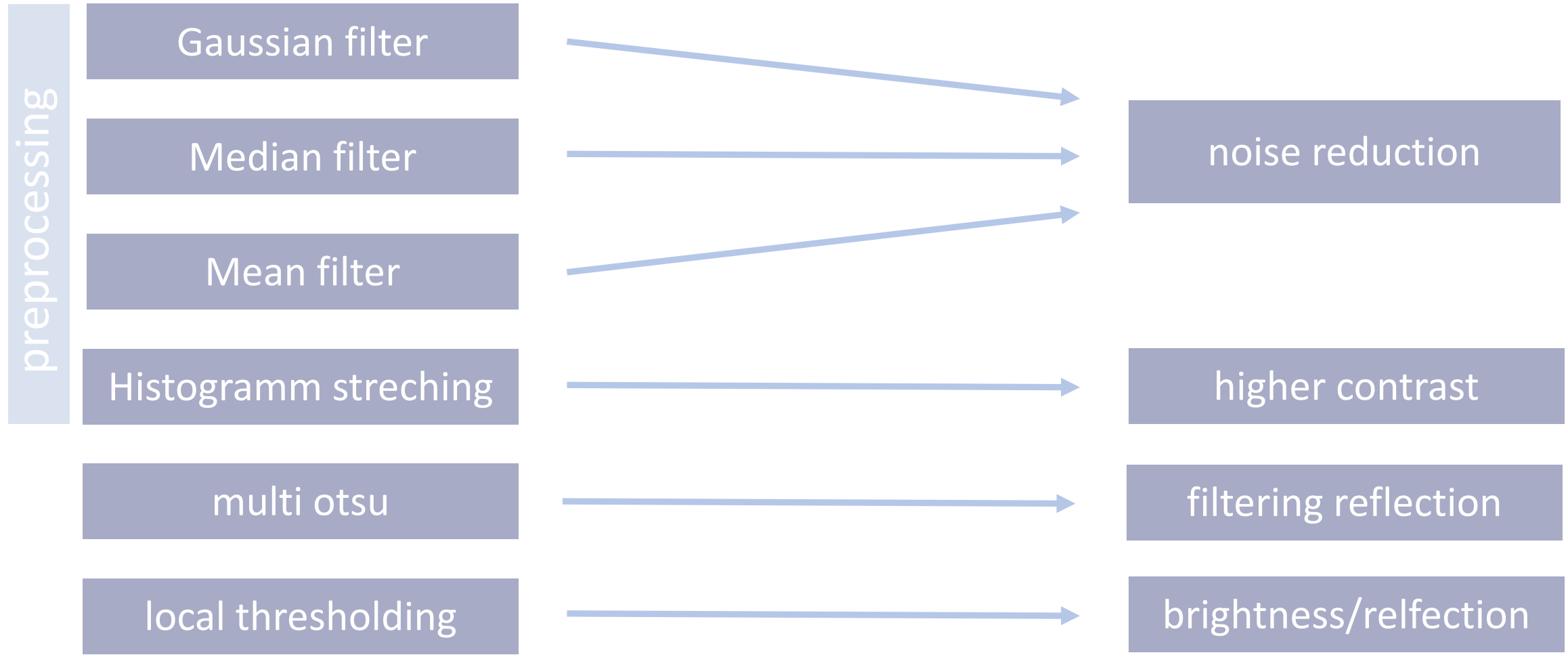
- low contrast

Challenges: NIH3T3



- varying brightness
- reflections

How to master these challenges?



Input image

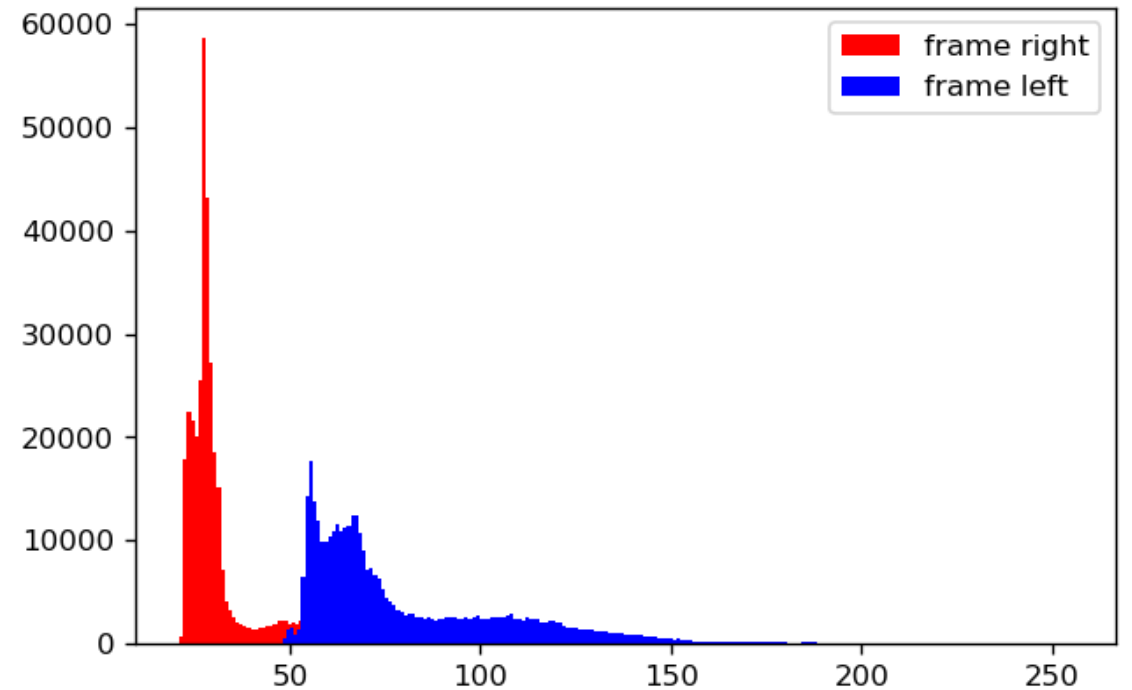
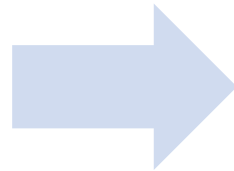
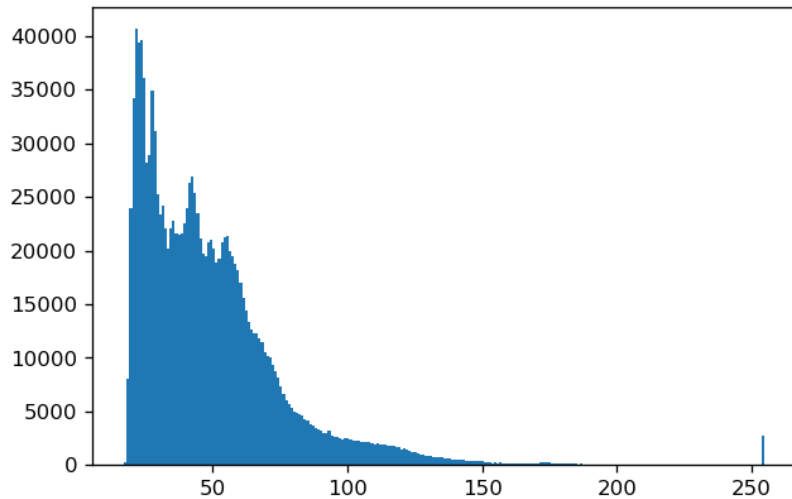
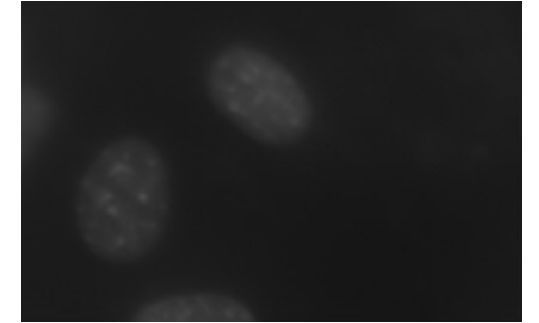
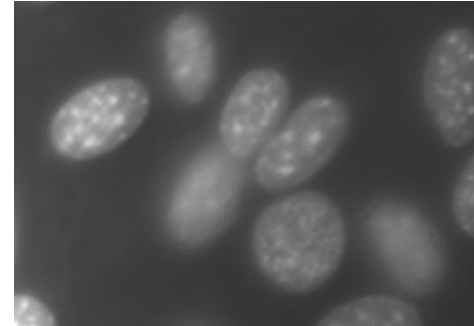
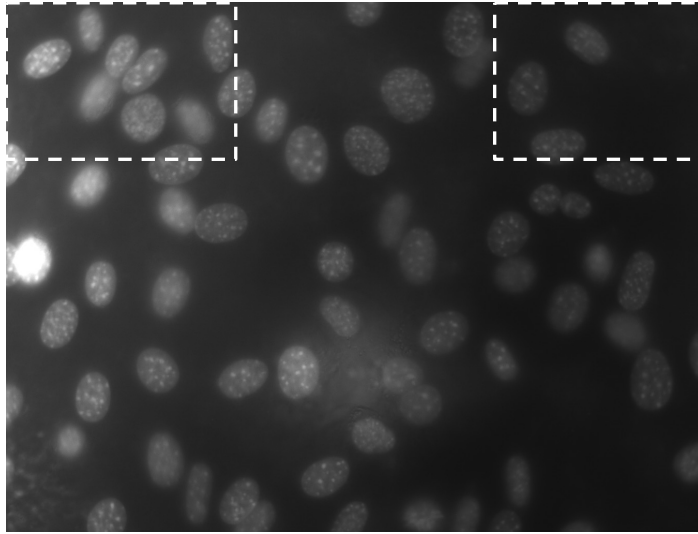
Preprocessing

Segmentation

Characteristics

Conclusion

Local thresholding



Input image

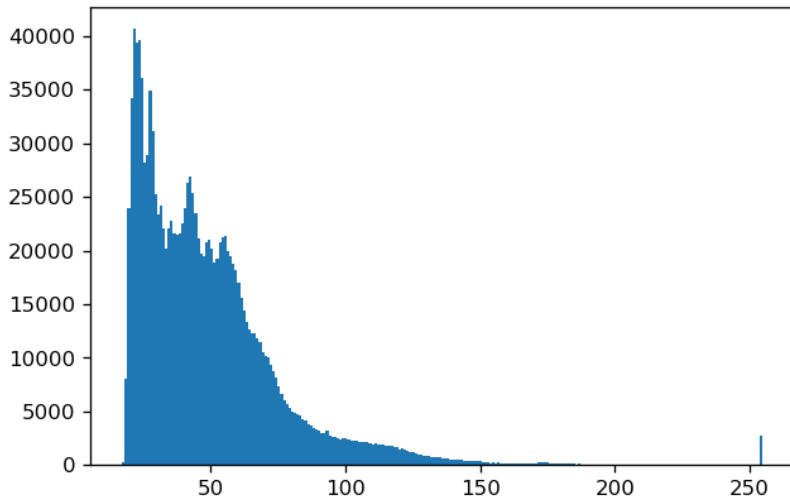
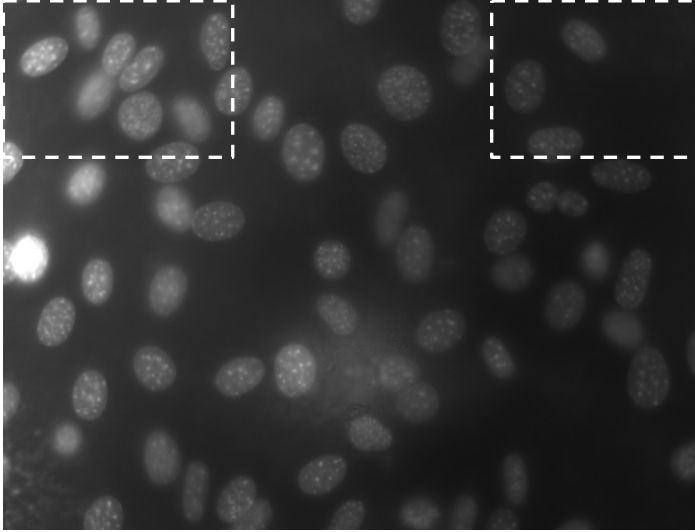
Preprocessing

Segmentation

Characteristics

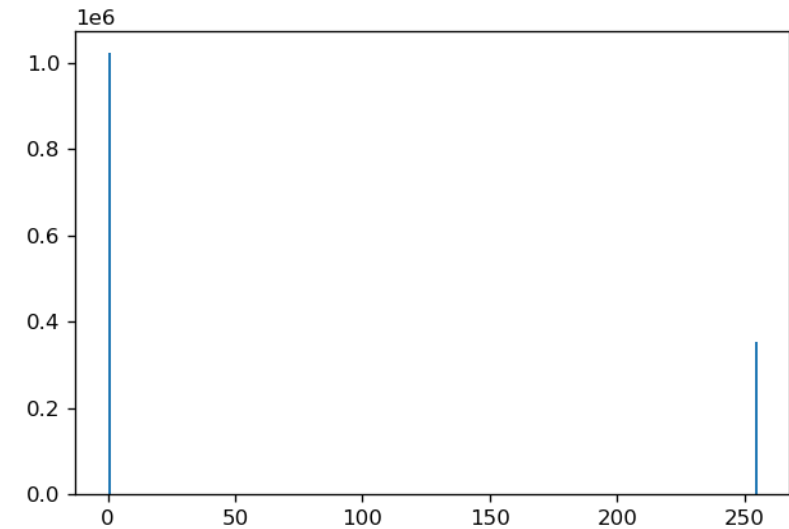
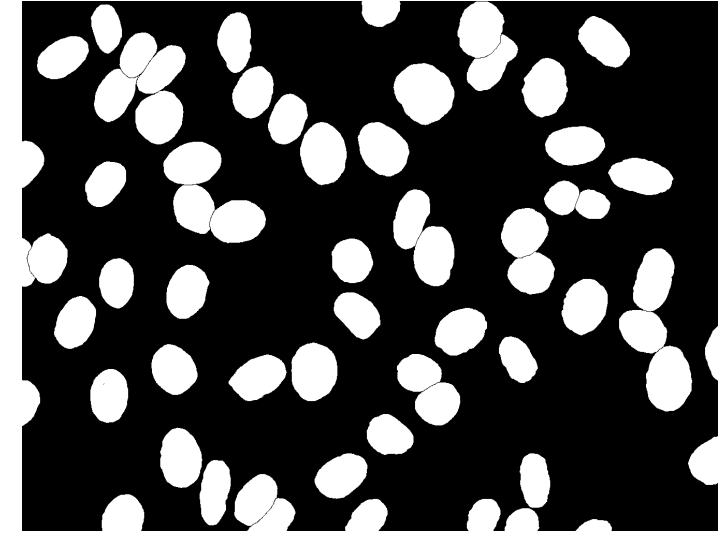
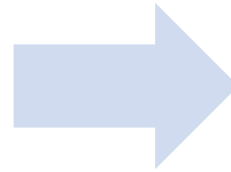
Conclusion

Local thresholding



Input image

Preprocessing

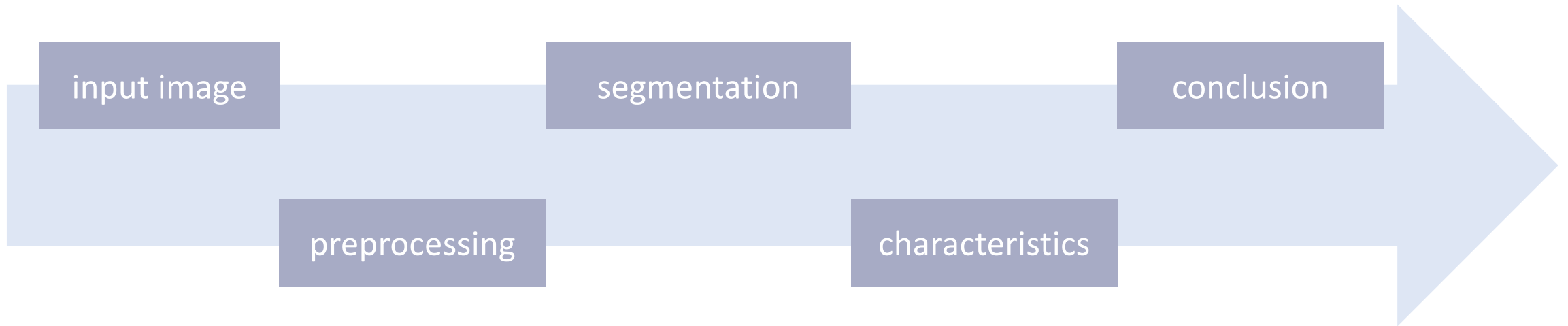


Characteristics

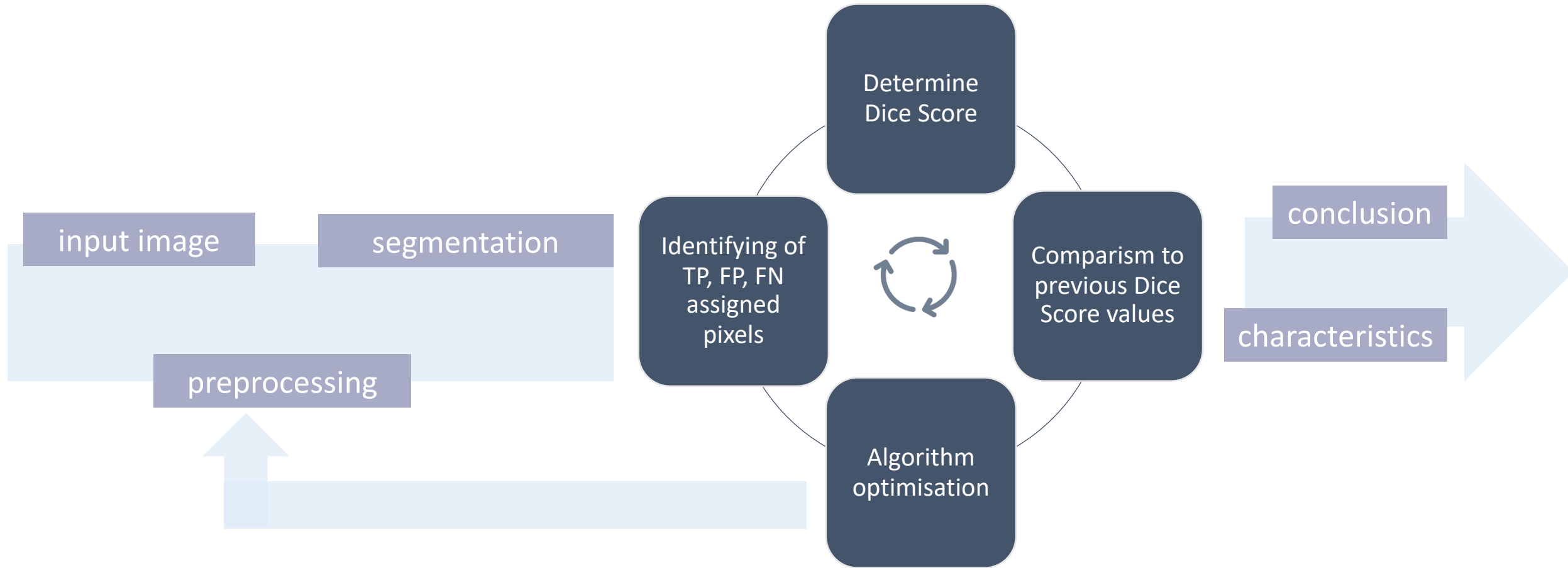
Conclusion

Segmentation

Procedure



Usage of Dice Score



Dice Score

Measure of similarity between two sets of data

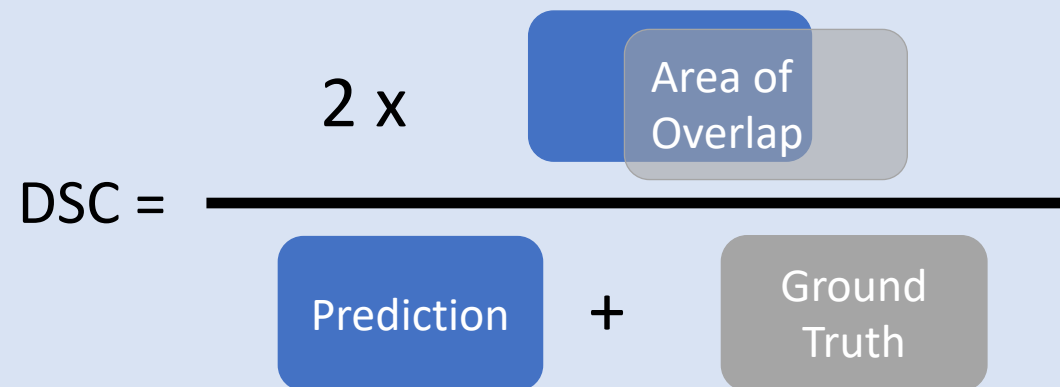


used to evaluate the similarity between a predicted segmentation mask and the ground truth segmentation mask

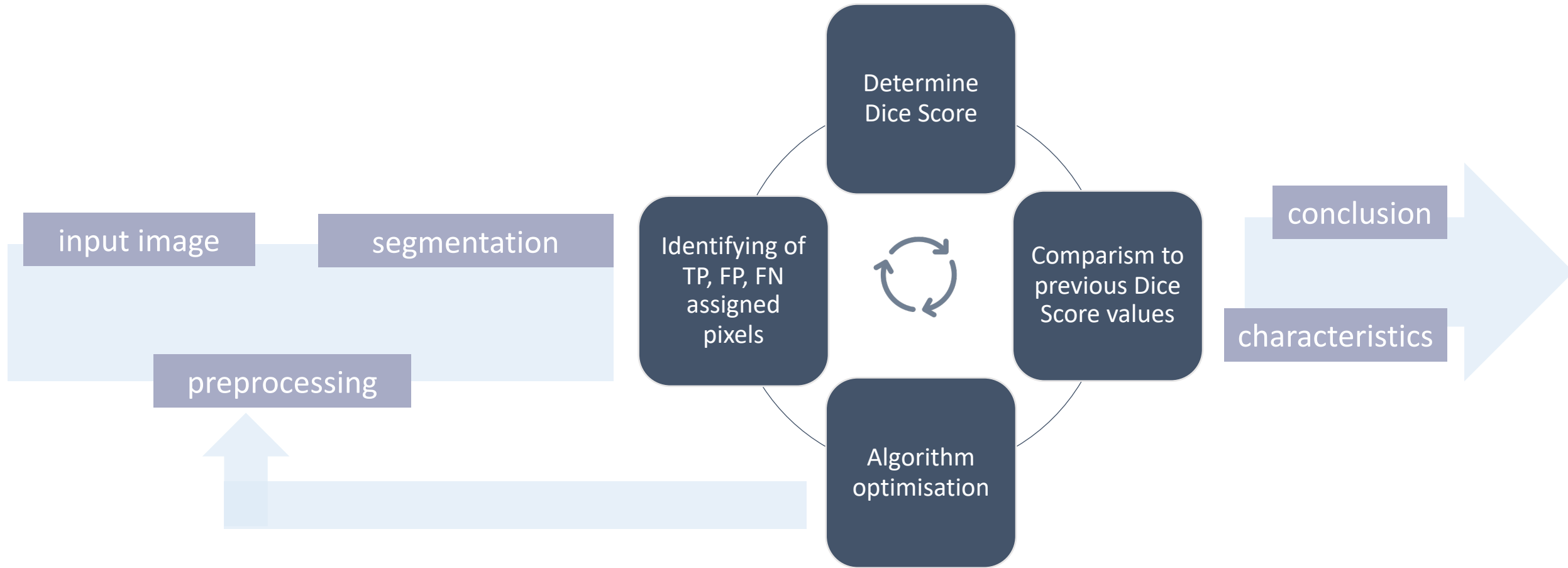
Boolean Data

$$\text{DSC} = \frac{2 \text{ TP}}{2 \text{ TP} + \text{FP} + \text{FN}}$$

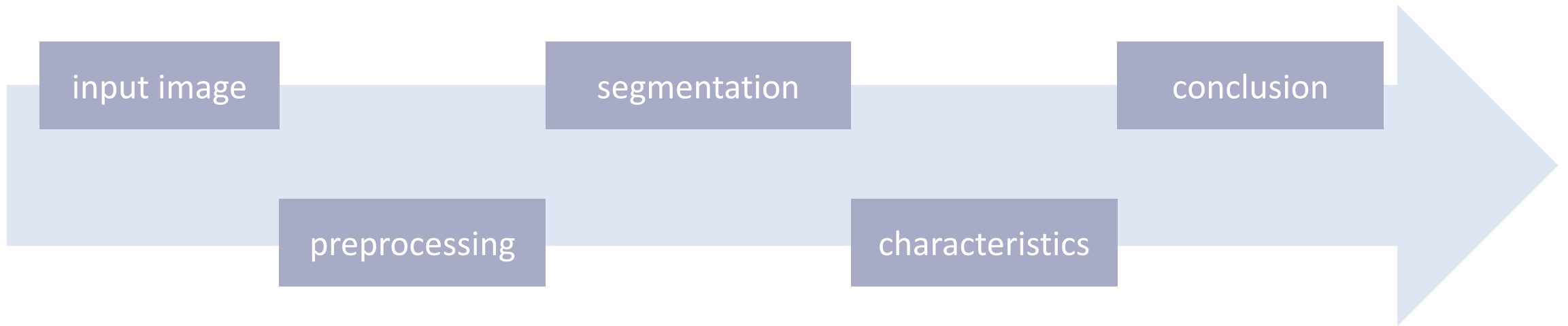
Discrete Data


$$\text{DSC} = \frac{2 \times \text{Area of Overlap}}{\text{Prediction} + \text{Ground Truth}}$$

Usage of Dice Score



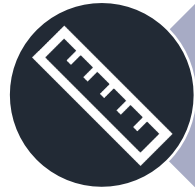
Procedure



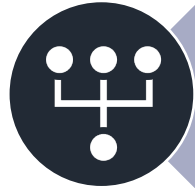
Additional ideas



Cell counting



Cell Size

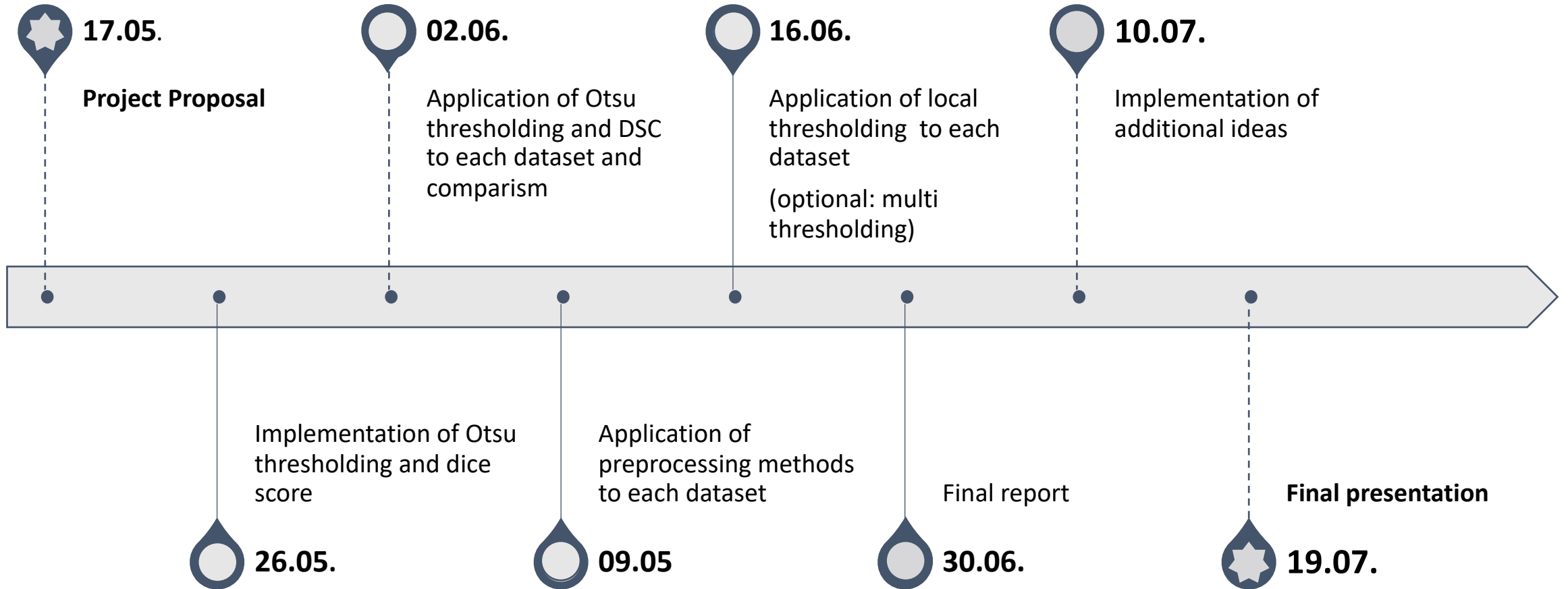


Cell arrangement



Other evaluation methods

Timeline



Python packages

- numpy (numerical python) → to work with numerical data
- skimage (scikit-image) → for image processing
- cv2 (OpenCV) → for solving computer vision problems
- matplotlib.pyplot → to visualize our data

→ and probably many more that we will discover while working on our project

