

Whole-Slide Image Analysis and Quantitative Pathology with QuPath

Thierry Pécot

Research Engineer

Biosit SFR UMS CNRS 3480 – Inserm 018

CZI Imaging Scientist



WHOLE-SLIDE IMAGE SIZE

- A **slide size** is generally **25 mm x 75 mm**

WHOLE-SLIDE IMAGE SIZE

- A **slide size** is generally **25 mm x 75 mm**
- A **20X** magnification gives a pixel width of **0.45 μm**

WHOLE-SLIDE IMAGE SIZE

- A **slide size** is generally **25 mm x 75 mm**
- A **20X** magnification gives a pixel width of **0.45 μm**
- If the **whole-slide image** is scanned:
 - $(25\text{e-}3 * 75\text{e-}3) / (0.45\text{e-}6)^2 = \mathbf{9\ 259\ 259\ 259\ pixels}$

WHOLE-SLIDE IMAGE SIZE

- A **slide size** is generally **25 mm x 75 mm**
- A **20X** magnification gives a pixel width of **0.45 μm**
- If the **whole-slide image** is scanned:
 - $(25\text{e-}3 * 75\text{e-}3) / (0.45\text{e-}6)^2 = \mathbf{9\ 259\ 259\ 259\ pixels}$
- For a **whole-slide image** with **1 channel** of fluorescence:
 - $9\ 259\ 259\ 259 * 1\ byte = \mathbf{9.26\ GB}$
- For a **H&E(S) whole-slide image** with **3 channels** (RGB):
 - $9\ 259\ 259\ 259 * 1\ byte * 3 = \mathbf{27.78\ GB}$

WHOLE-SLIDE IMAGE SIZE

- A **slide size** is generally **25 mm x 75 mm**
- A **20X** magnification gives a pixel width of **0.45 µm**
- If the **whole-slide image** is scanned:
 - $(25\text{e-}3 * 75\text{e-}3) / (0.45\text{e-}6)^2 = \mathbf{9\ 259\ 259\ 259\ pixels}$
 - For a **whole-slide image** with **1 channel** of fluorescence:
 - $9\ 259\ 259\ 259 * 1\ byte = \mathbf{9.26\ GB}$
 - For a **H&E(S) whole-slide image** with **3 channels** (RGB):
 - $9\ 259\ 259\ 259 * 1\ byte * 3 = \mathbf{27.78\ GB}$



! not possible to load the whole data in the RAM memory

WHOLE-SLIDE IMAGE SIZE

- A **slide size** is generally **25 mm x 75 mm**
- A **20X** magnification gives a pixel width of **0.45 µm**
- If the **whole-slide image** is scanned:
 - $(25\text{e-}3 * 75\text{e-}3) / (0.45\text{e-}6)^2 = 9\,259\,259\,259 \text{ pixels}$
 - For a **whole-slide image** with **1 channel** of fluorescence:
 - $9\,259\,259\,259 * 1 \text{ byte} = 9.26 \text{ GB}$
 - For a **H&E(S) whole-slide image** with **3 channels** (RGB):
 - $9\,259\,259\,259 * 1 \text{ byte} * 3 = 27.78 \text{ GB}$



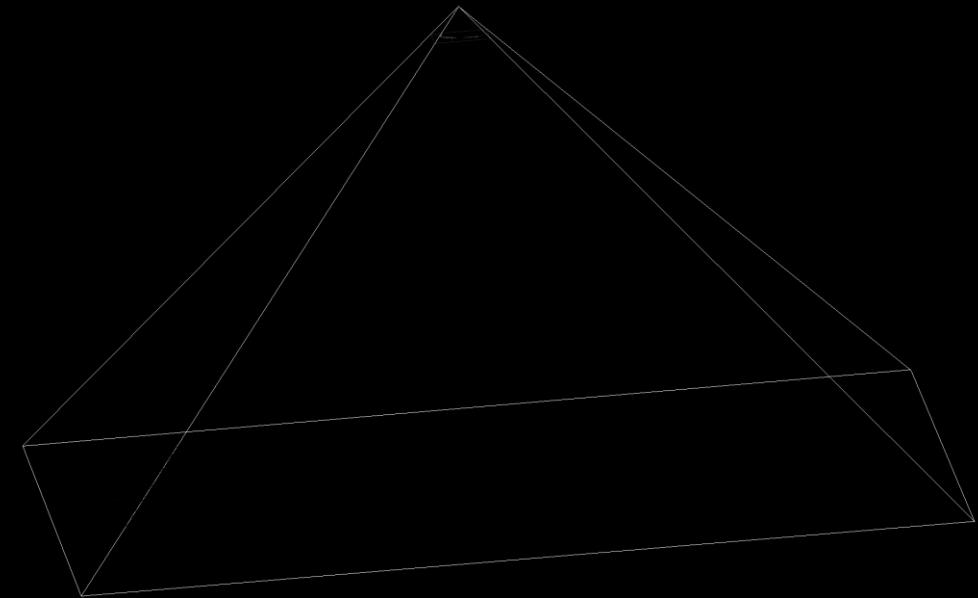
not possible to load the whole data in the **RAM memory**



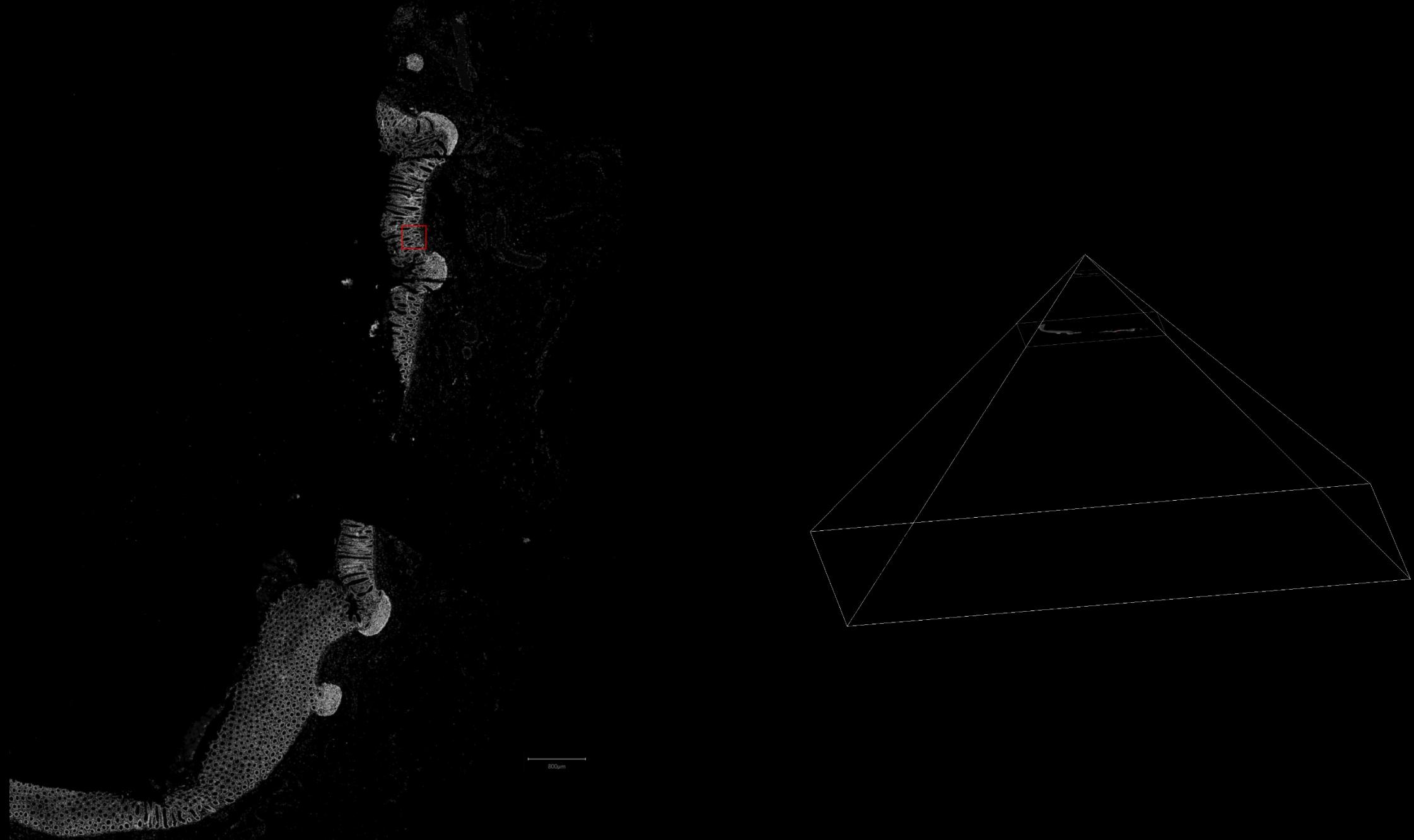
only load the **required information** in the RAM memory:

- Define **several resolutions** to create a **pyramidal representation**

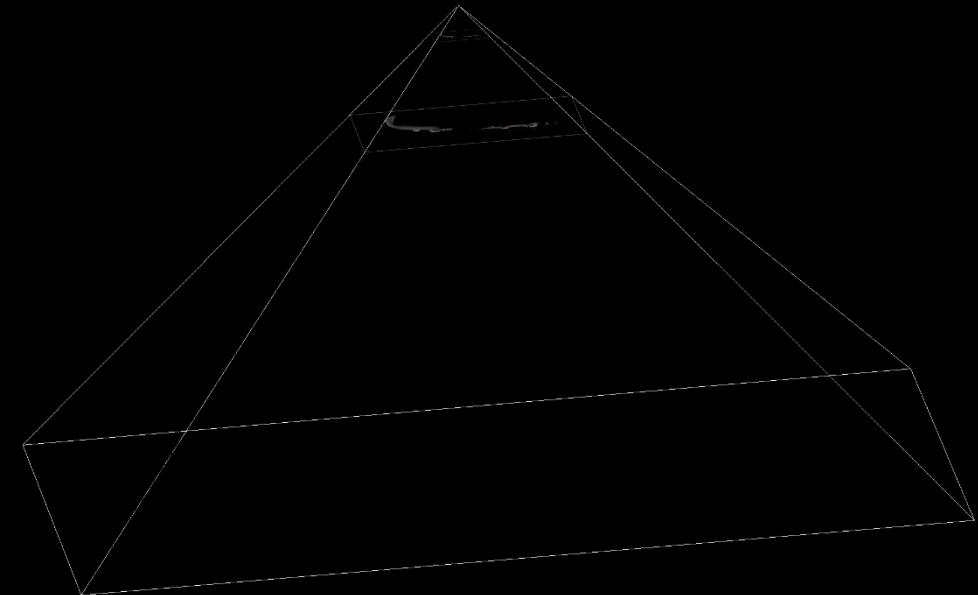
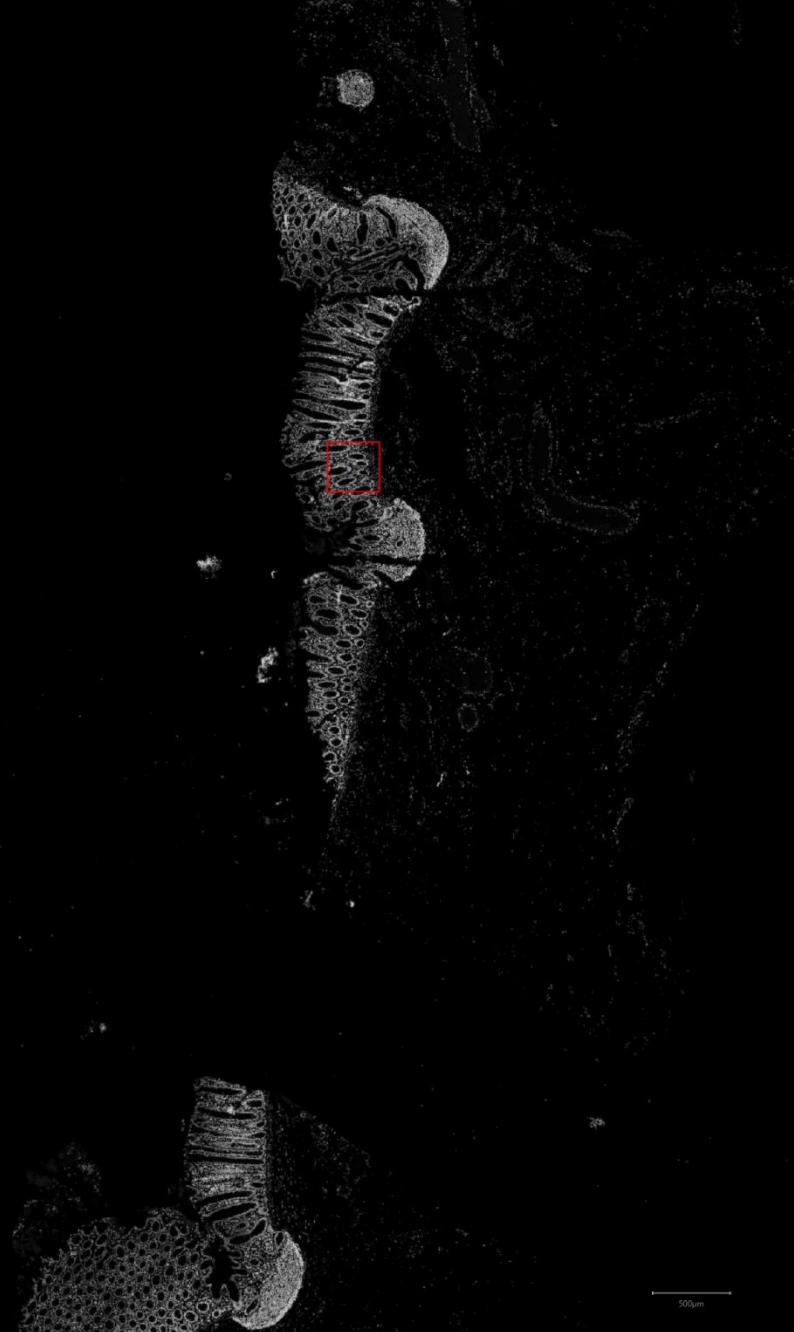
WHOLE SLIDE IMAGE – PYRAMIDAL REPRESENTATION



WHOLE SLIDE IMAGE – PYRAMIDAL REPRESENTATION

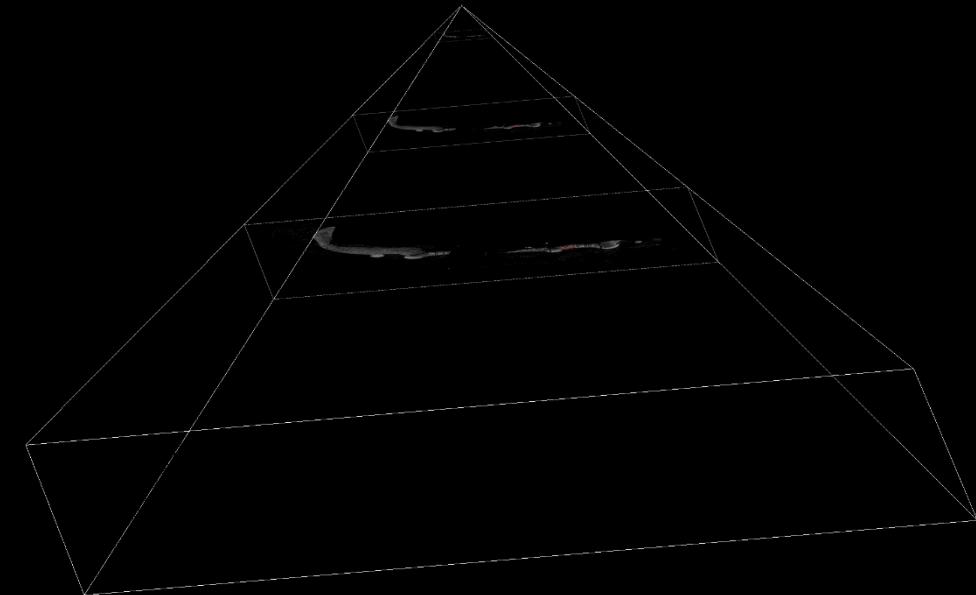
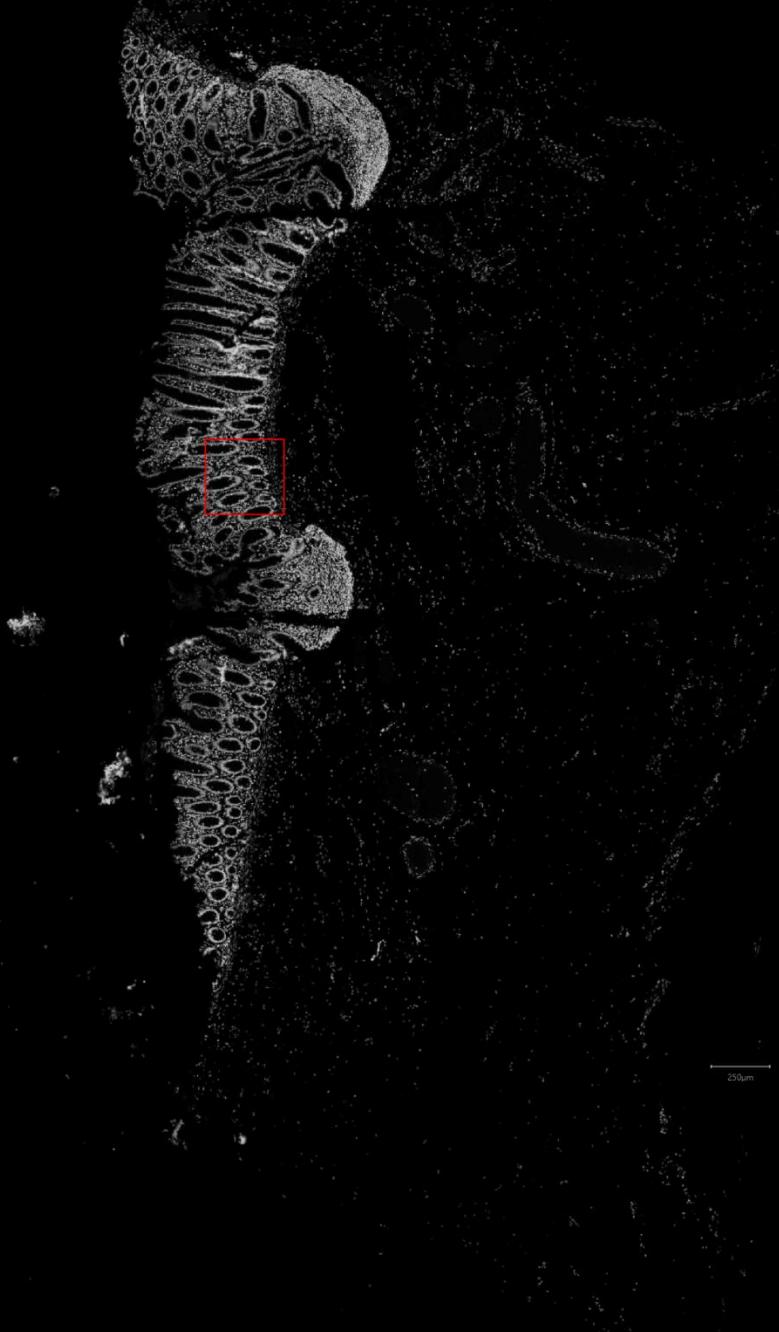


WHOLE SLIDE IMAGE – PYRAMIDAL REPRESENTATION

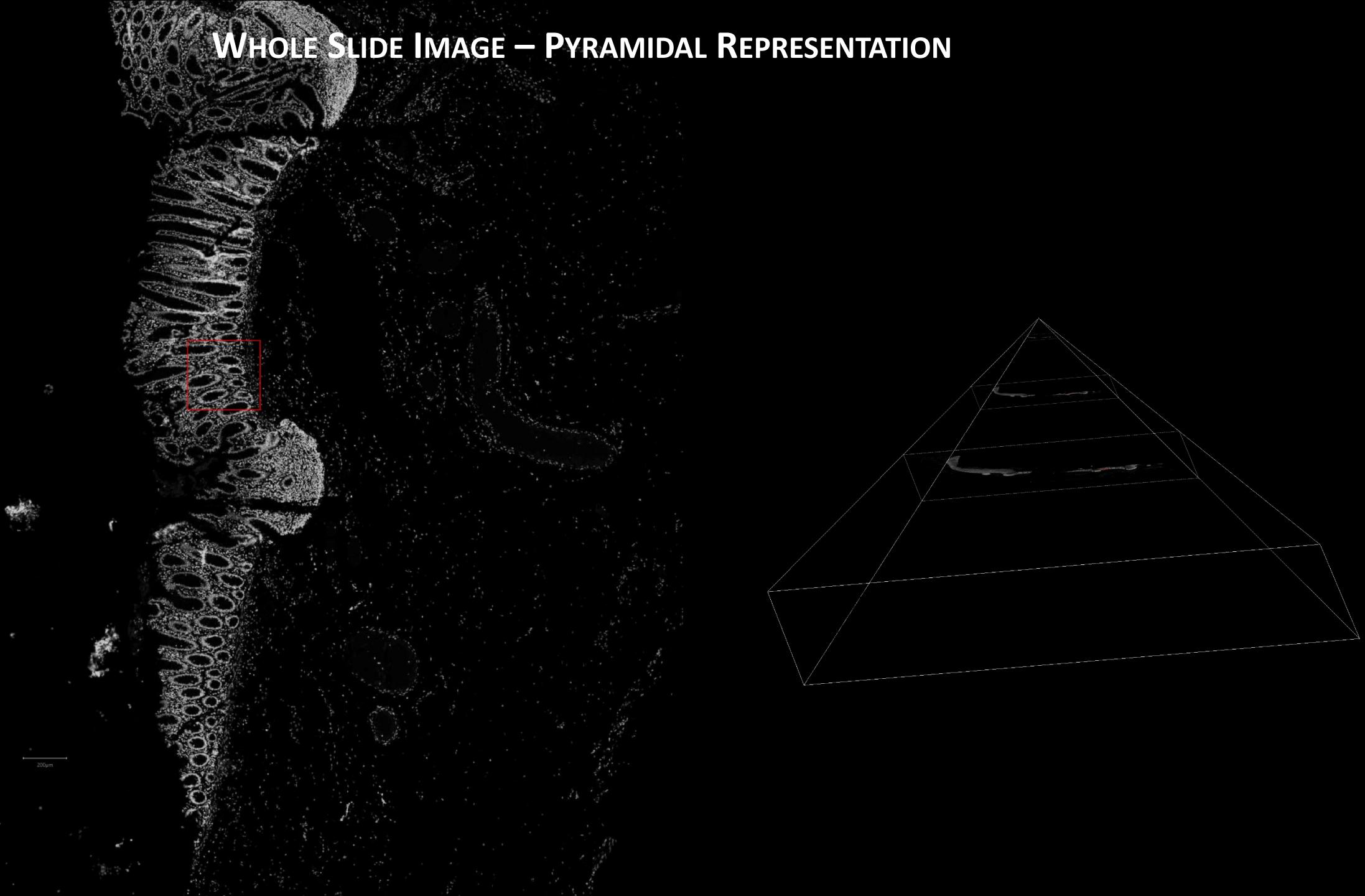


500µm

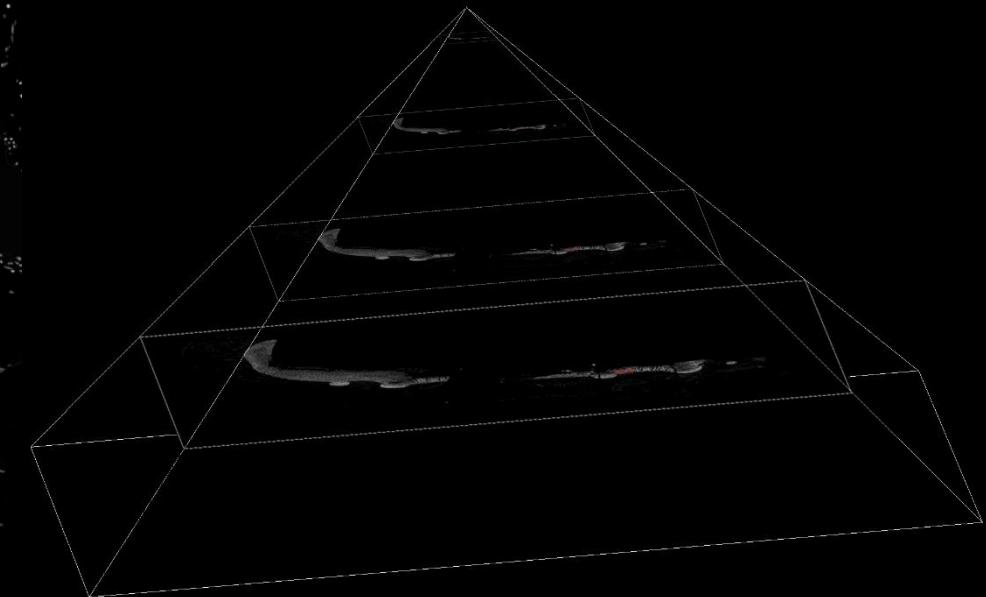
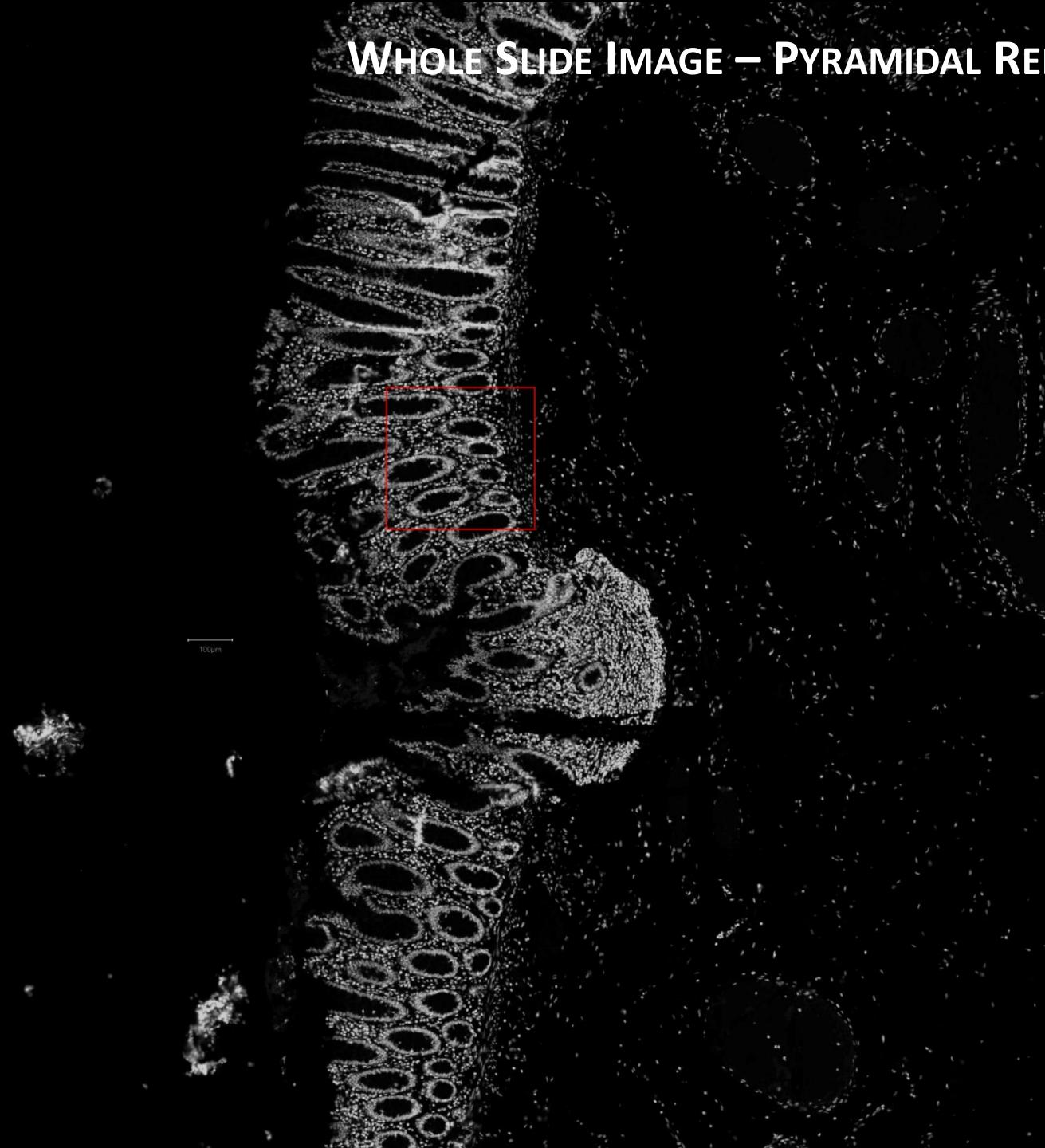
WHOLE SLIDE IMAGE – PYRAMIDAL REPRESENTATION



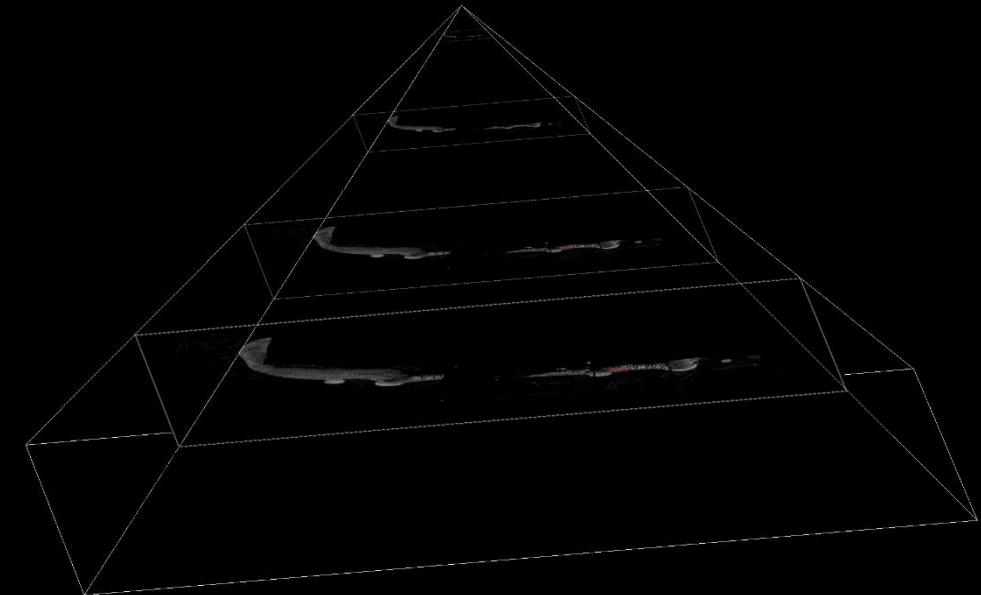
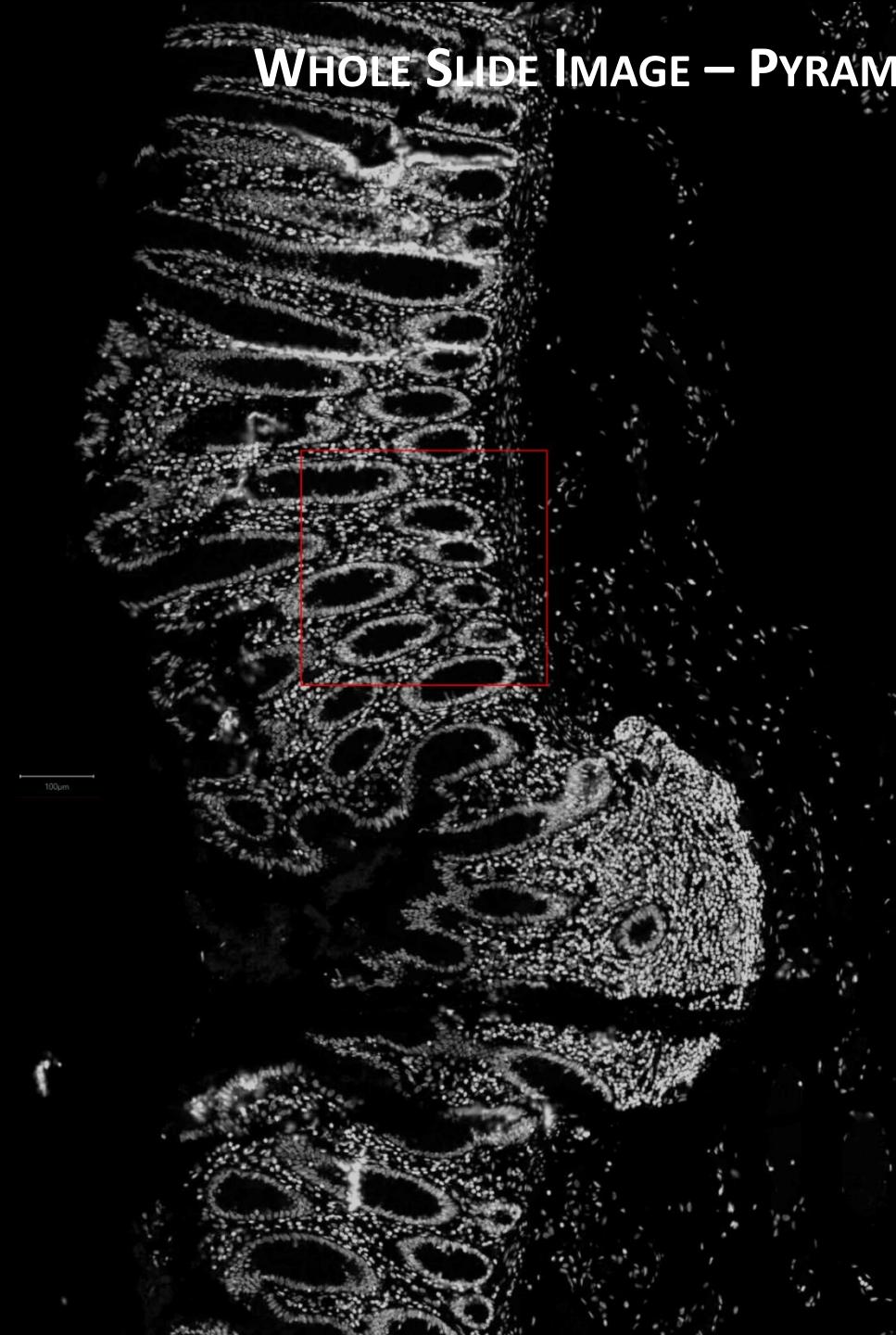
WHOLE SLIDE IMAGE – PYRAMIDAL REPRESENTATION



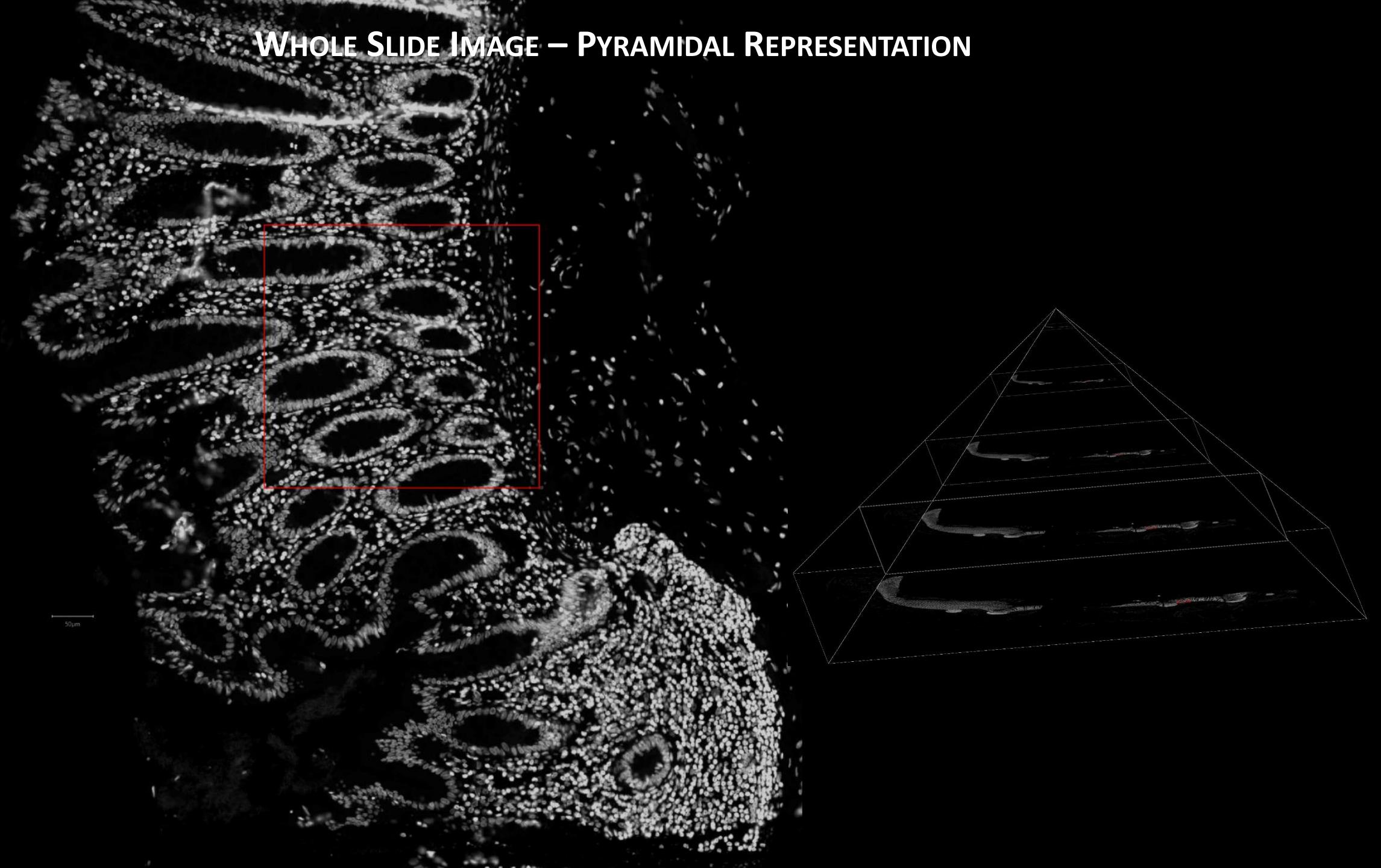
WHOLE SLIDE IMAGE – PYRAMIDAL REPRESENTATION



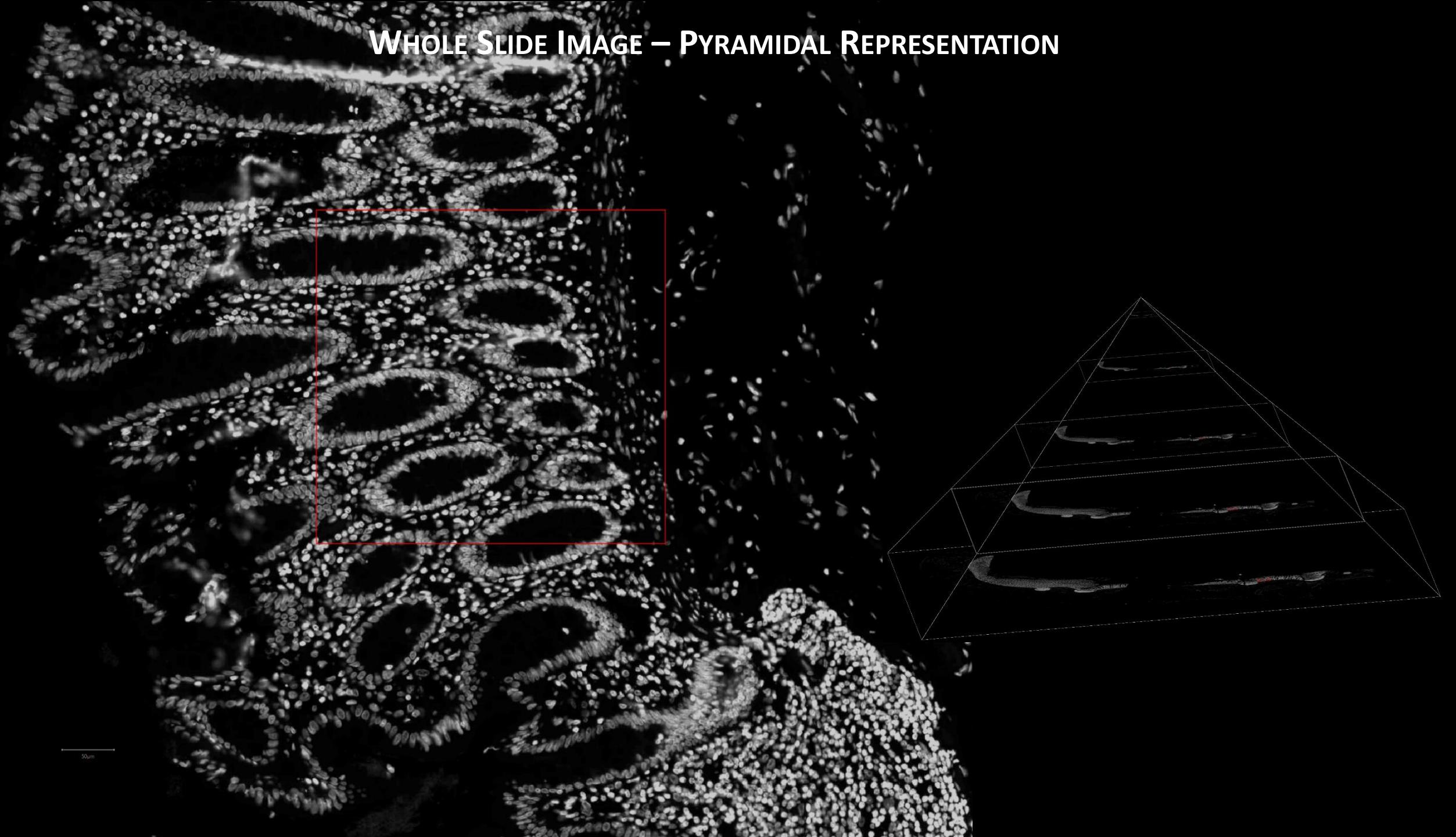
WHOLE SLIDE IMAGE – PYRAMIDAL REPRESENTATION



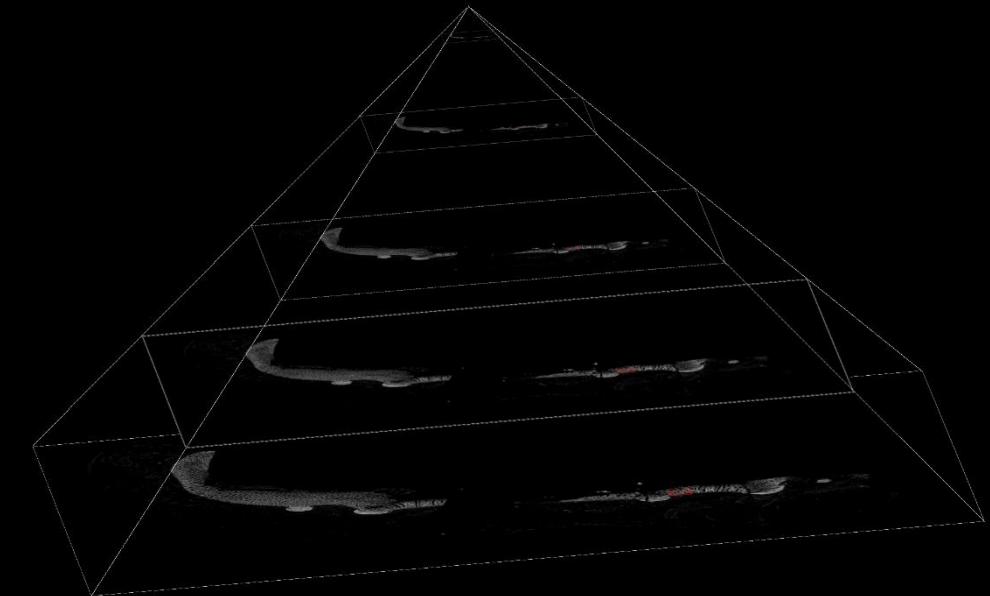
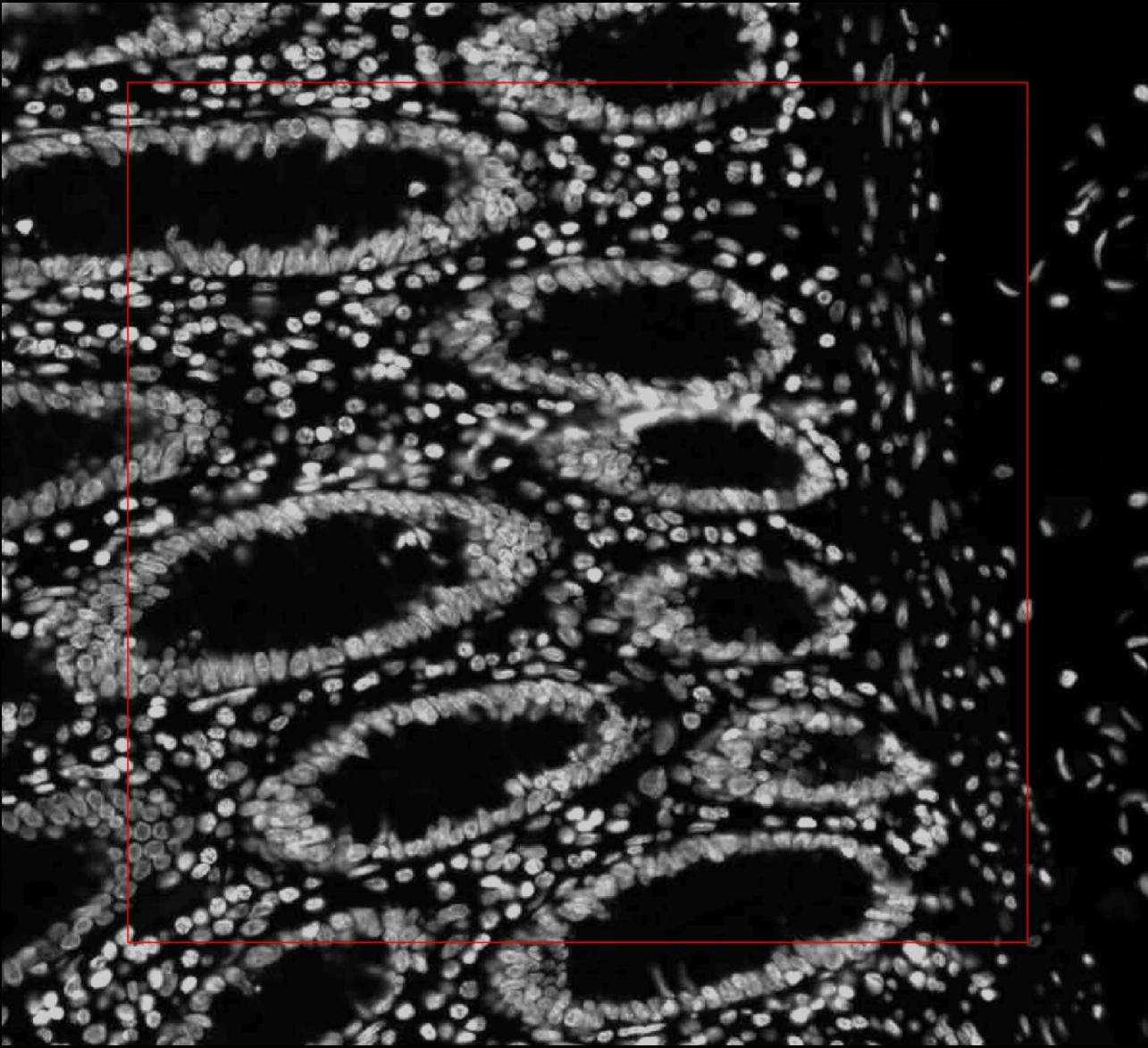
WHOLE SLIDE IMAGE – PYRAMIDAL REPRESENTATION



WHOLE SLIDE IMAGE – PYRAMIDAL REPRESENTATION



WHOLE SLIDE IMAGE – PYRAMIDAL REPRESENTATION



WHOLE-SLIDE IMAGE SIZE

- A **full H&E(S)** whole-slide image would be:
 - $9\ 259\ 259\ 259 * 1 \text{ byte} * 3 = \mathbf{27.78 \text{ GB}}$ for resolution 1
 - $9\ 259\ 259\ 259 * 1 \text{ byte} * 3 = \mathbf{6.94 \text{ GB}}$ for resolution 2
 - $9\ 259\ 259\ 259 * 1 \text{ byte} * 3 = \mathbf{1.74 \text{ GB}}$ for resolution 4
 - $9\ 259\ 259\ 259 * 1 \text{ byte} * 3 = \mathbf{434 \text{ MB}}$ for resolution 8
 - $9\ 259\ 259\ 259 * 1 \text{ byte} * 3 = \mathbf{109 \text{ MB}}$ for resolution 16
 - $9\ 259\ 259\ 259 * 1 \text{ byte} * 3 = \mathbf{27 \text{ MB}}$ for resolution 32
 - $9\ 259\ 259\ 259 * 1 \text{ byte} * 3 = \mathbf{7 \text{ MB}}$ for resolution 64
 - $9\ 259\ 259\ 259 * 1 \text{ byte} * 3 = \mathbf{1.7 \text{ MB}}$ for resolution 128
 - $9\ 259\ 259\ 259 * 1 \text{ byte} * 3 = \mathbf{424 \text{ KB}}$ for resolution 256

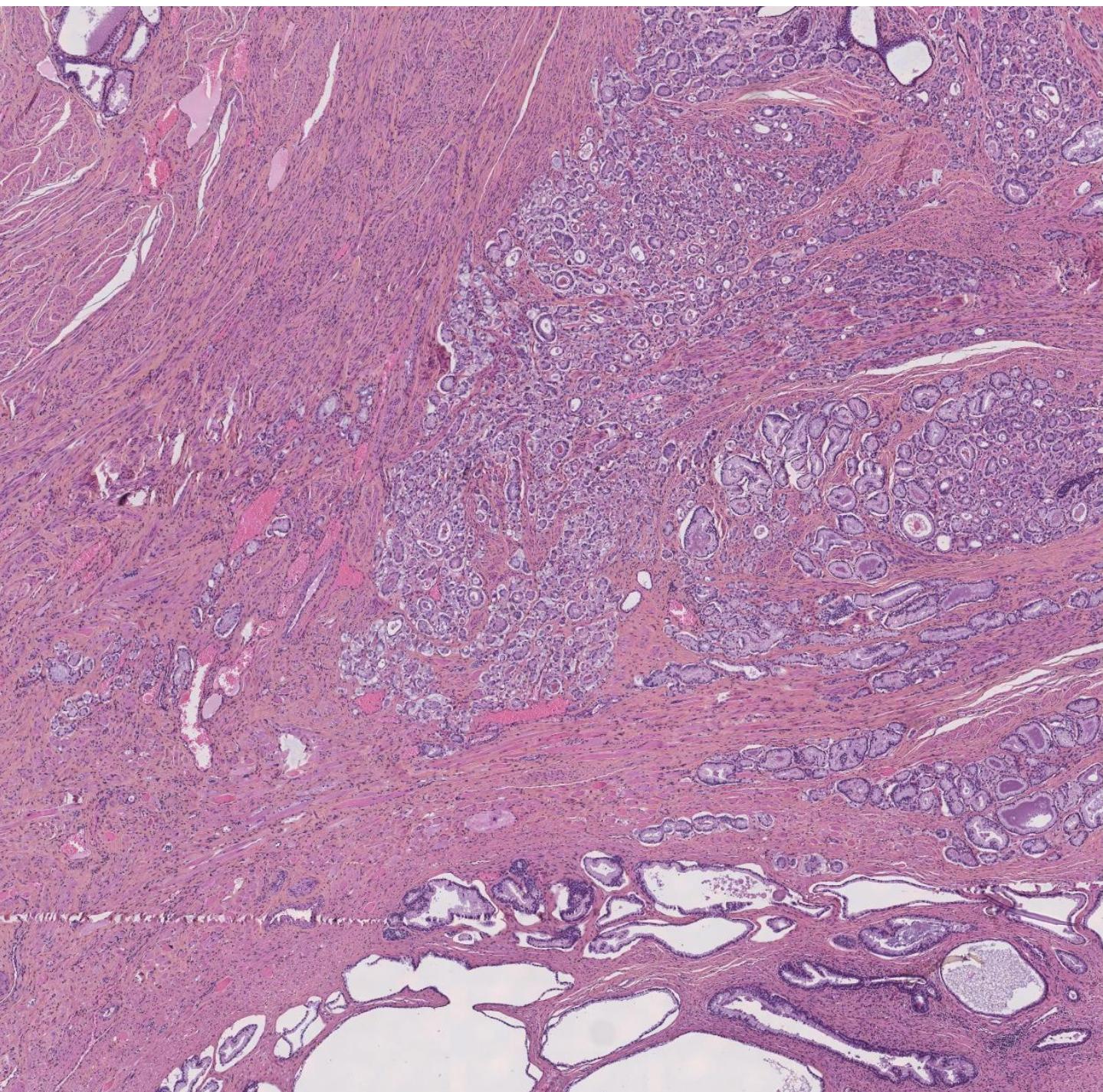
→ Total of **37 GB** uncompressed data

ANNOTATIONS

- Allow to **add information** to specific regions or entire images
- Lots of **features/measurements** can then be extracted from these regions
- **Powerful and storage-efficient** way to process images
- Can be **manually** defined or **automatically** estimated
- Can be enriched by **adding classes**

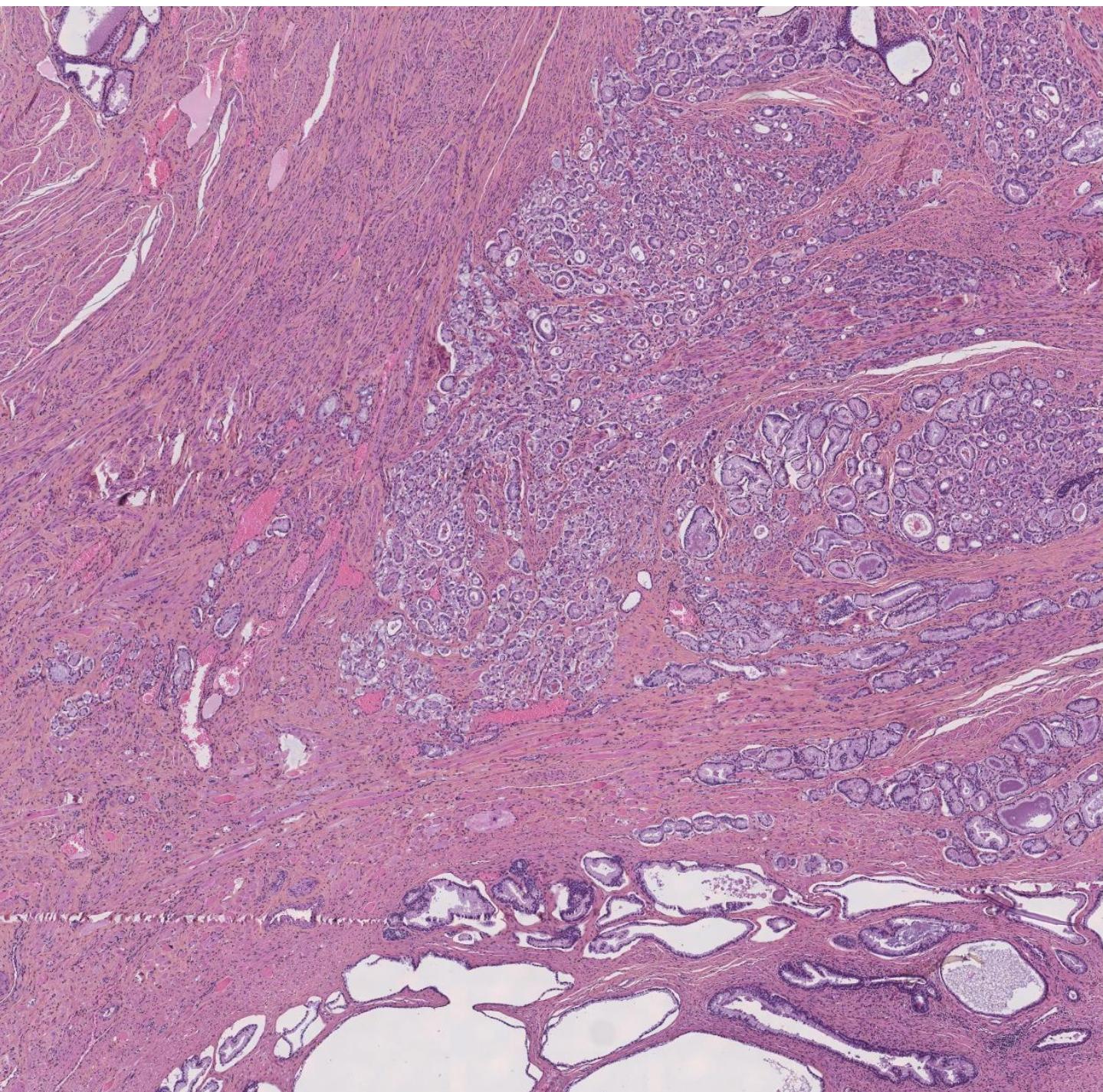
ANNOTATIONS

- Allow to **add information** to specific regions or entire images
- Lots of **features/measurements** can then be extracted from these regions
- **Powerful and storage-efficient** way to process images
- Can be **manually defined** or **automatically estimated**
- Can be enriched by **adding classes**



ANNOTATIONS

- Allow to **add information** to specific regions or entire images
- Lots of **features/measurements** can then be extracted from these regions
- **Powerful and storage-efficient** way to process images
- Can be **manually defined** or **automatically estimated**
- Can be enriched by **adding classes**
- **Open Prostate1.ome.tif**
- Create **different types** of annotations
- Play with **resolution**
- Look at the **measurements** for each **type of annotation**



STAIN ESTIMATION

- **Hematoxylin** stains **nuclei** in purple/blue
- **Eosin** stains **extracellular matrix** and **cytoplasm** in pink
- **DAB** is used to stain **antigens** in brown

STAIN ESTIMATION

- **Hematoxylin** stains **nuclei** in purple/blue
- **Eosin** stains **extracellular matrix** and **cytoplasm** in pink
- **DAB** is used to stain **antigens** in brown
- Slides are processed with a **brightfield scanner**:
→ A **digital image** with 3 color components (**RGB**) is obtained

STAIN ESTIMATION

- **Hematoxylin** stains **nuclei** in purple/blue
- **Eosin** stains **extracellular matrix** and **cytoplasm** in pink
- **DAB** is used to stain **antigens** in brown
- Slides are processed with a **brightfield scanner**:
  A **digital image** with 3 color components (**RGB**) is obtained
- **Stain estimation** consists in transforming Red-Green-Blue channels to **Hematoxylin-Eosin/DAB-Residue** channels
- It greatly facilitates the **nuclei segmentation** in the **hematoxylin** component, the **DAB** region **characterization** in the **DAB** component, ...

STAIN ESTIMATION

- **Hematoxylin** stains **nuclei** in purple/blue
- **Eosin** stains **extracellular matrix** and **cytoplasm** in pink
- **DAB** is used to stain **antigens** in brown
- Slides are processed with a **brightfield scanner**:
→ A **digital image** with 3 color components (**RGB**) is obtained
- **Stain estimation** consists in transforming Red-Green-Blue channels to **Hematoxylin-Eosin/DAB-Residue** channels
- It greatly facilitates the **nuclei segmentation** in the **hematoxylin** component, the **DAB** region **characterization** in the DAB component, ...
- **Automatic stain estimation in QuPath** is based on:

Comparative Study > *Anal Quant Cytol Histol.* 2001 Aug;23(4):291-9.

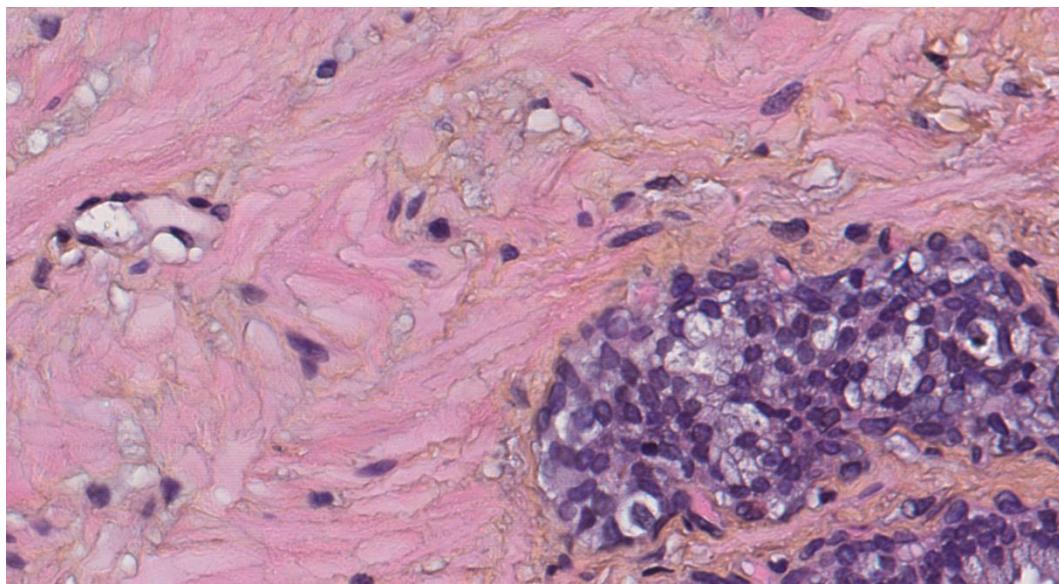
Quantification of histochemical staining by color deconvolution

A C Ruifrok ¹, D A Johnston

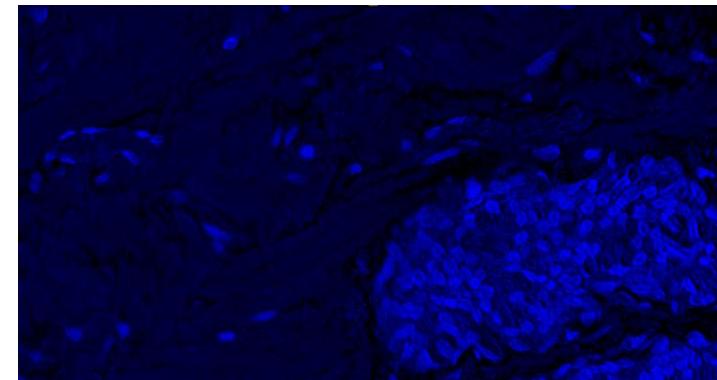
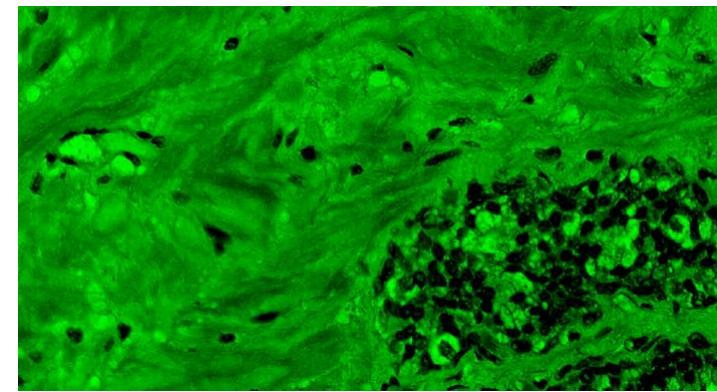
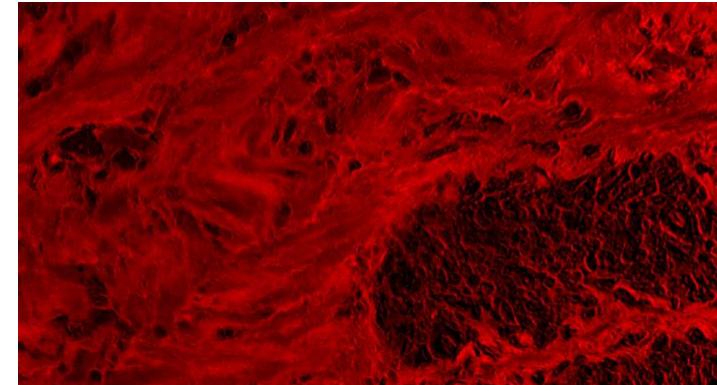
Affiliations + expand

PMID: 11531144

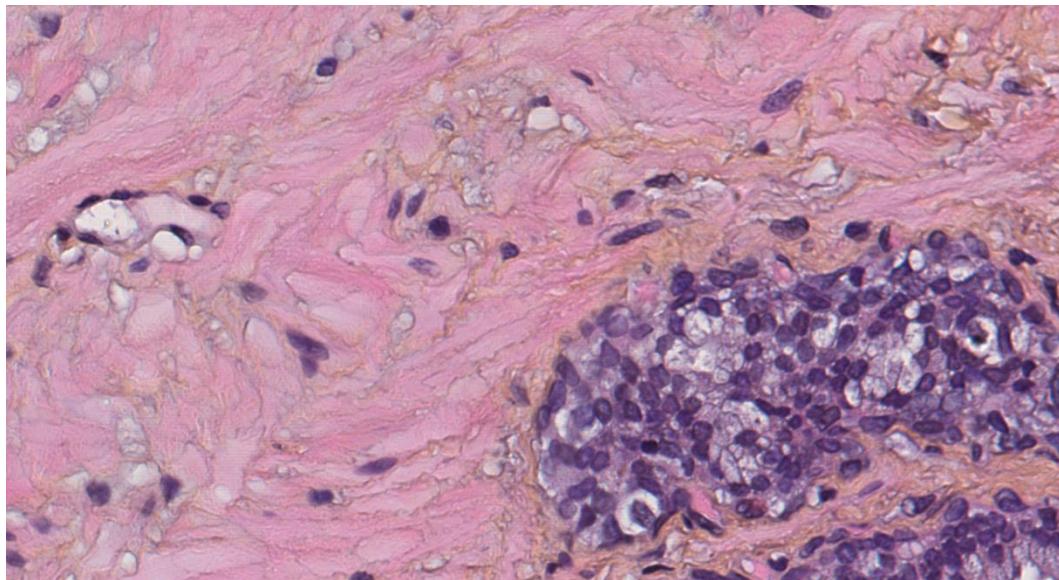
H&E STAIN ESTIMATION



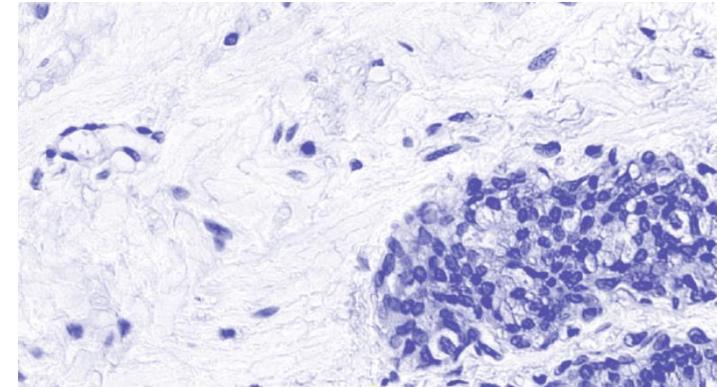
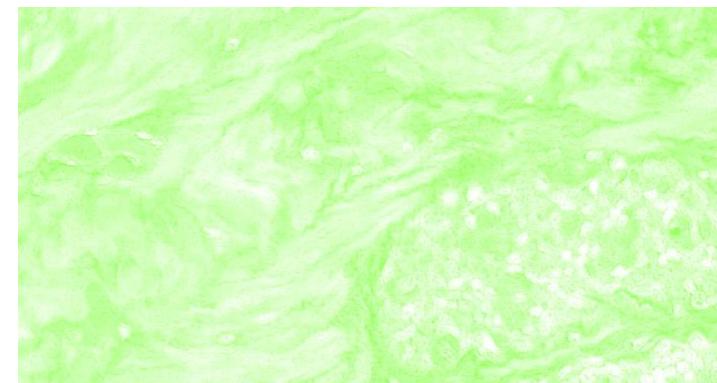
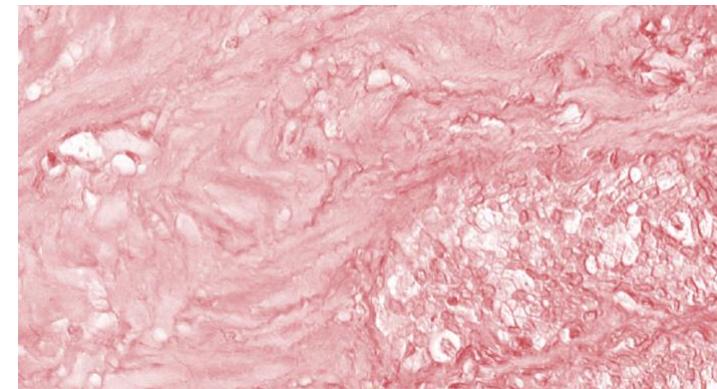
=



H&E STAIN ESTIMATION

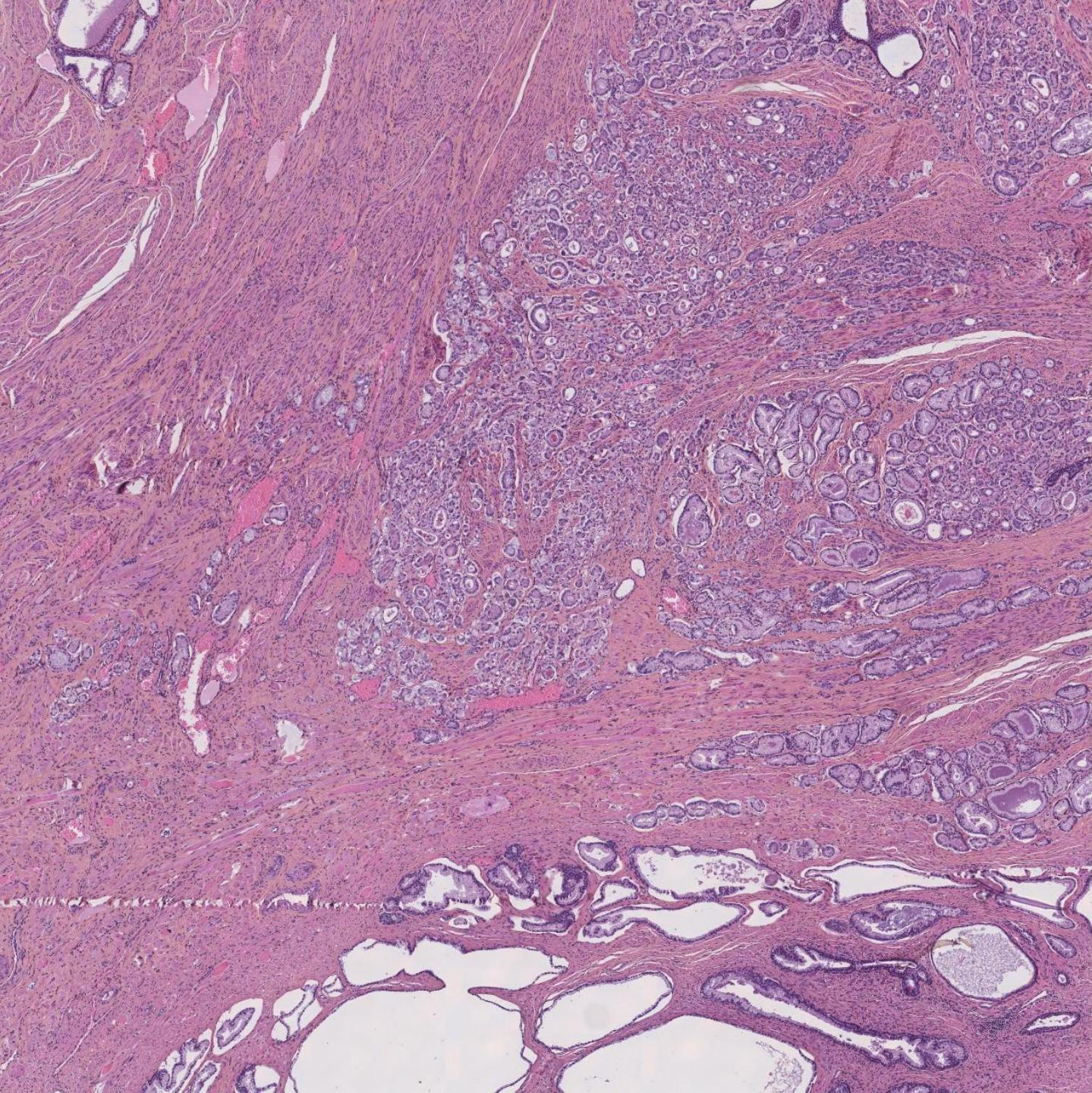


=



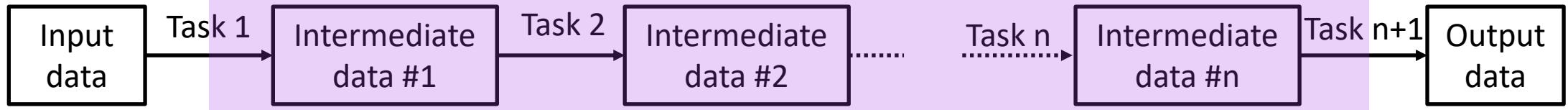
H&E STAIN ESTIMATION

- **Open Prostate_1.ome.tif**
- Create a small rectangle annotation and **estimate stain vectors**
- **Manually define Hematoxylin and Eosin components**
- **Visualize the differences**



EXPLICIT PROGRAMMING

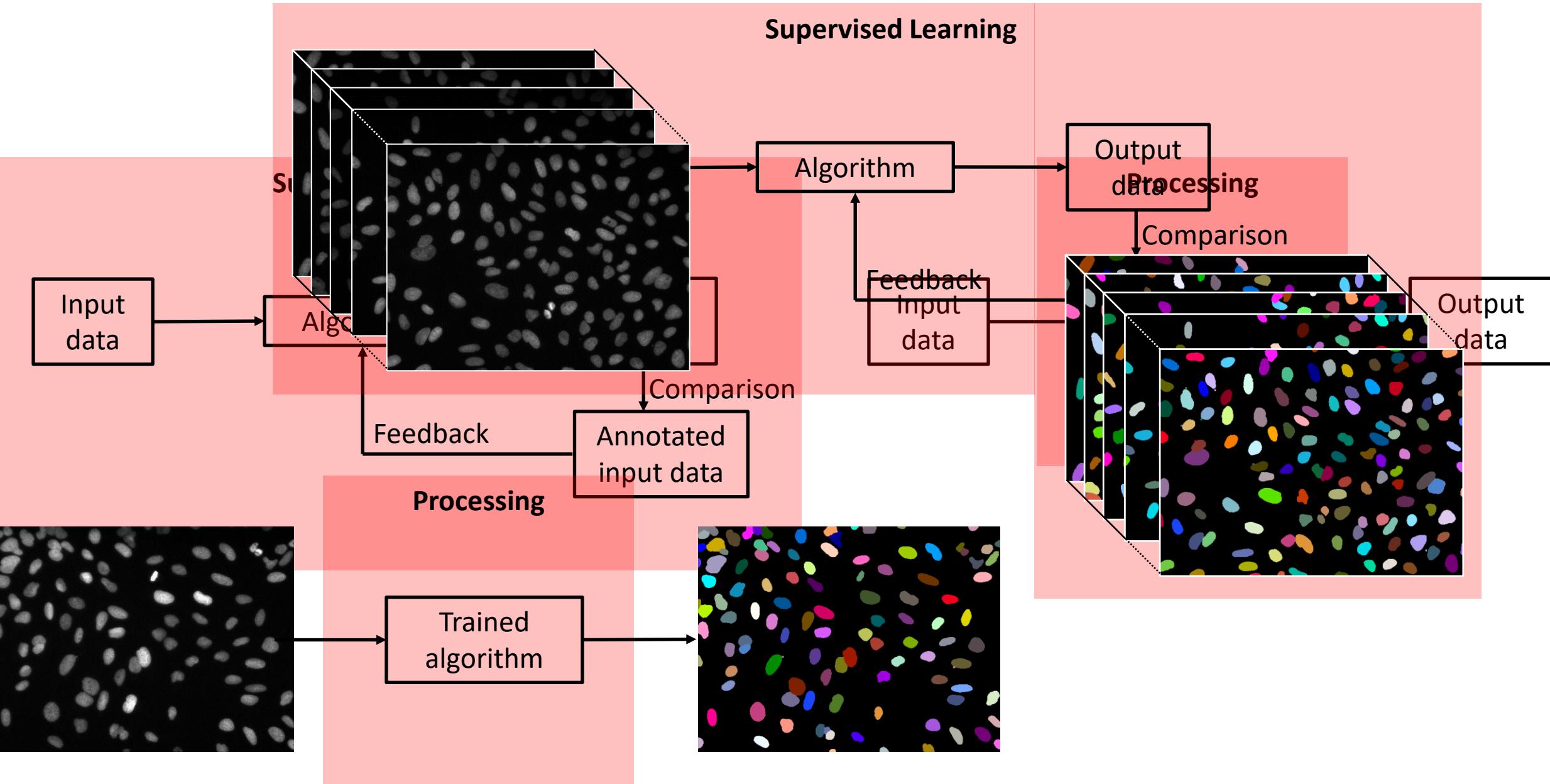
Image processing workflow



Input image



SUPERVISED MACHINE LEARNING

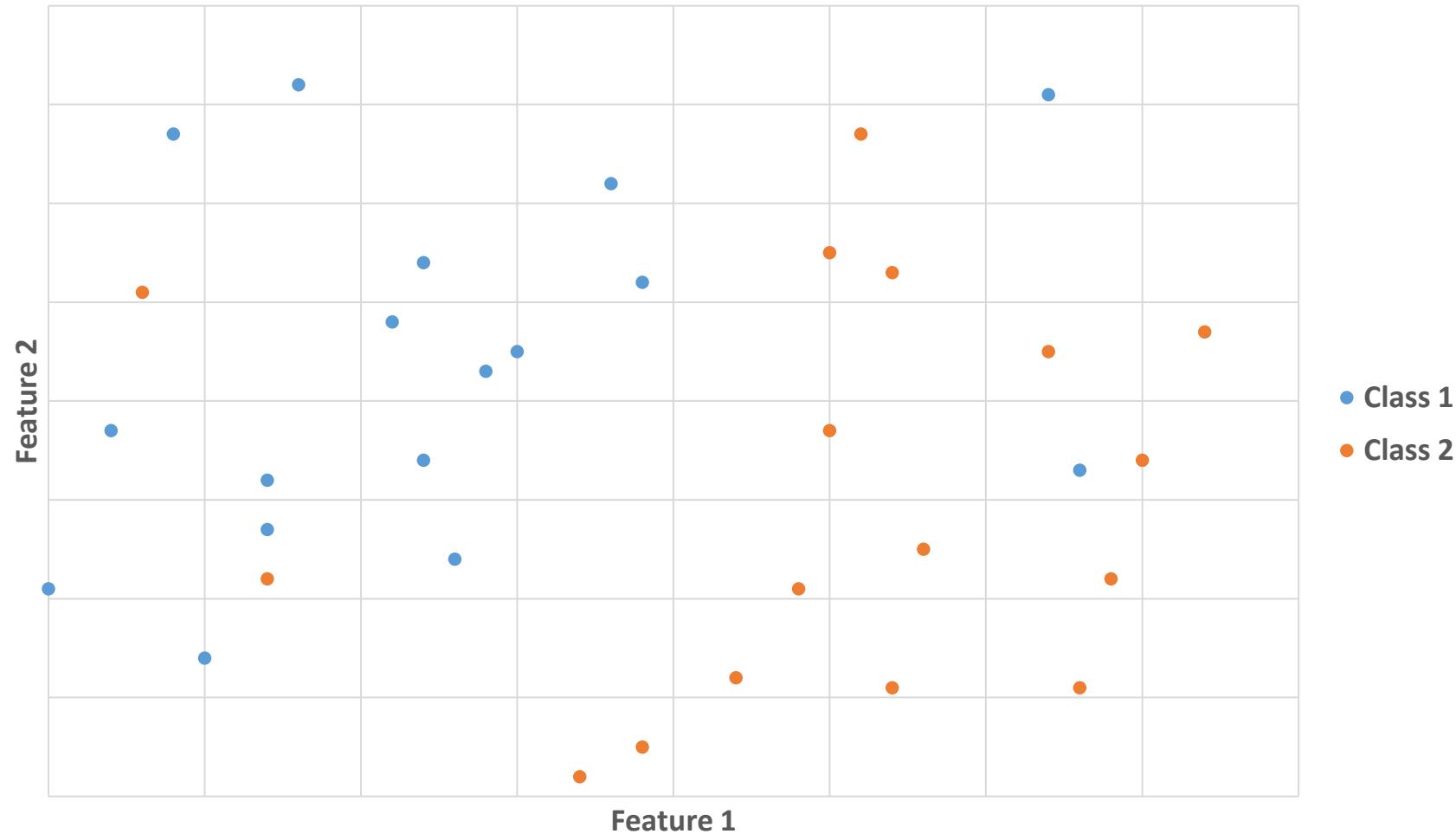


SHALLOW MACHINE LEARNING FOR PIXEL CLASSIFICATION

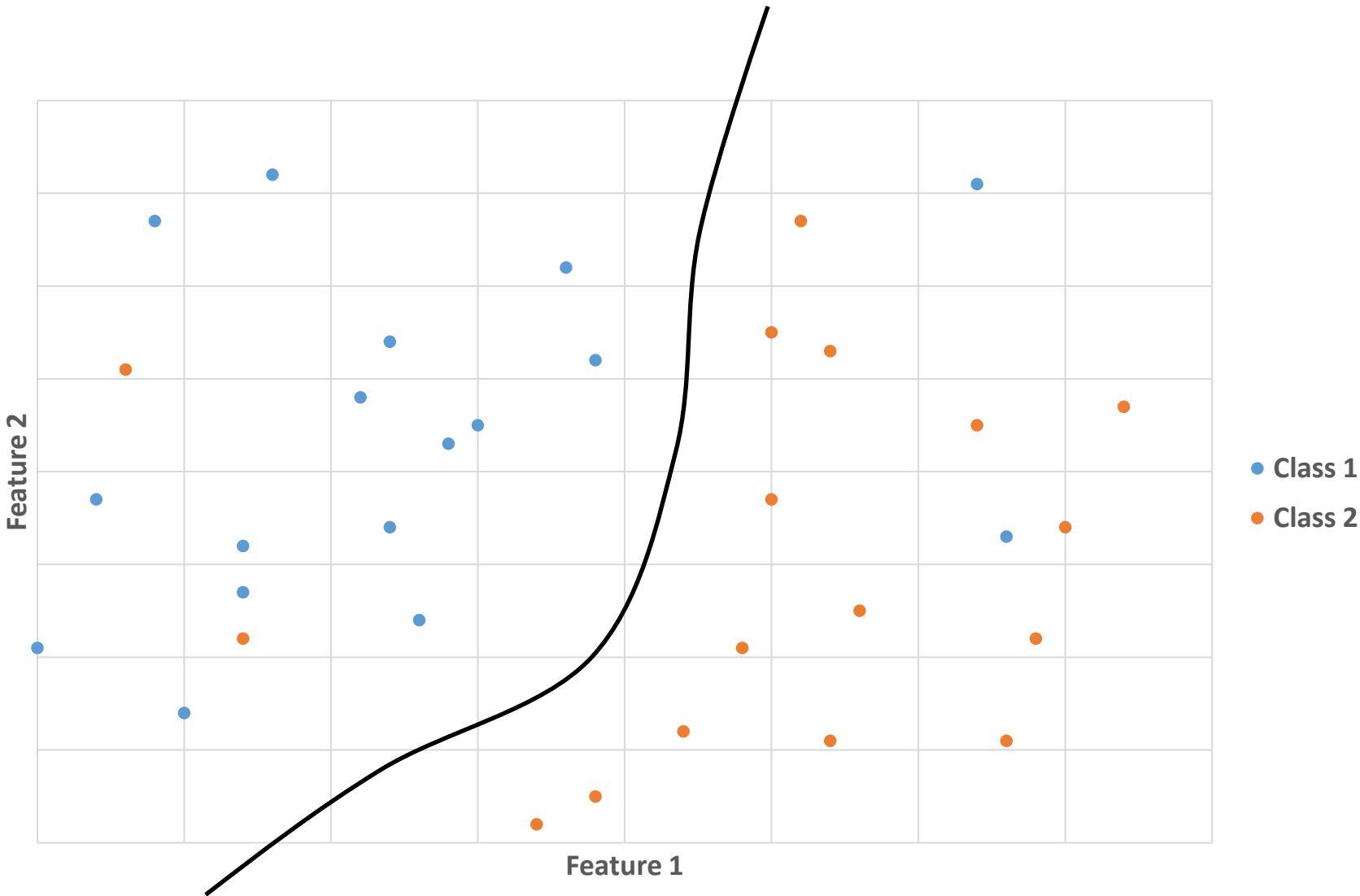
Supervised classification:

- Examples of classes are **manually** defined by the user
- A **classifier** is **trained** by using **defined features** with these examples
- Data is then **automatically classified** by using the trained classifier

SHALLOW MACHINE LEARNING FOR PIXEL CLASSIFICATION

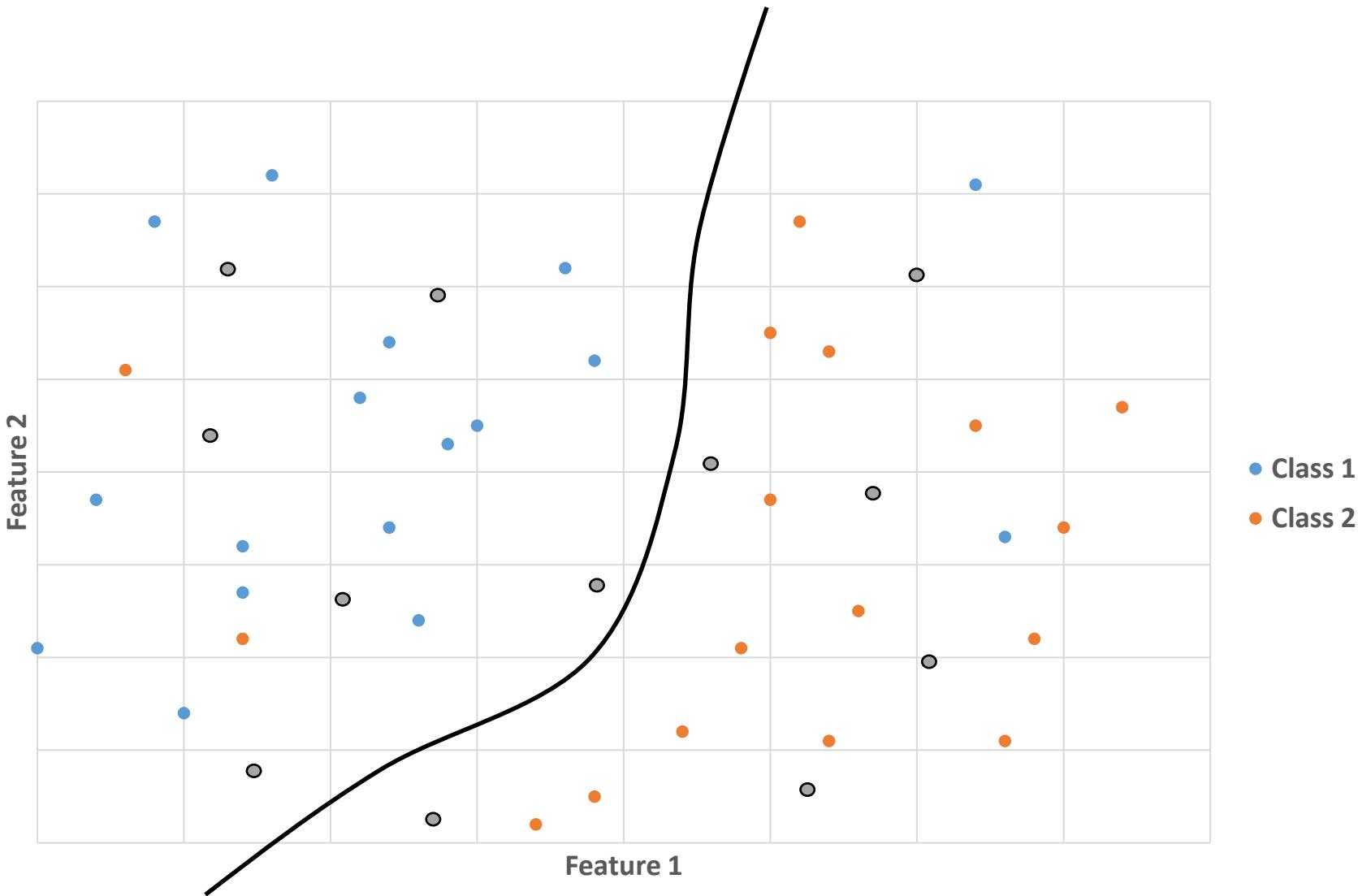


SHALLOW MACHINE LEARNING FOR PIXEL CLASSIFICATION



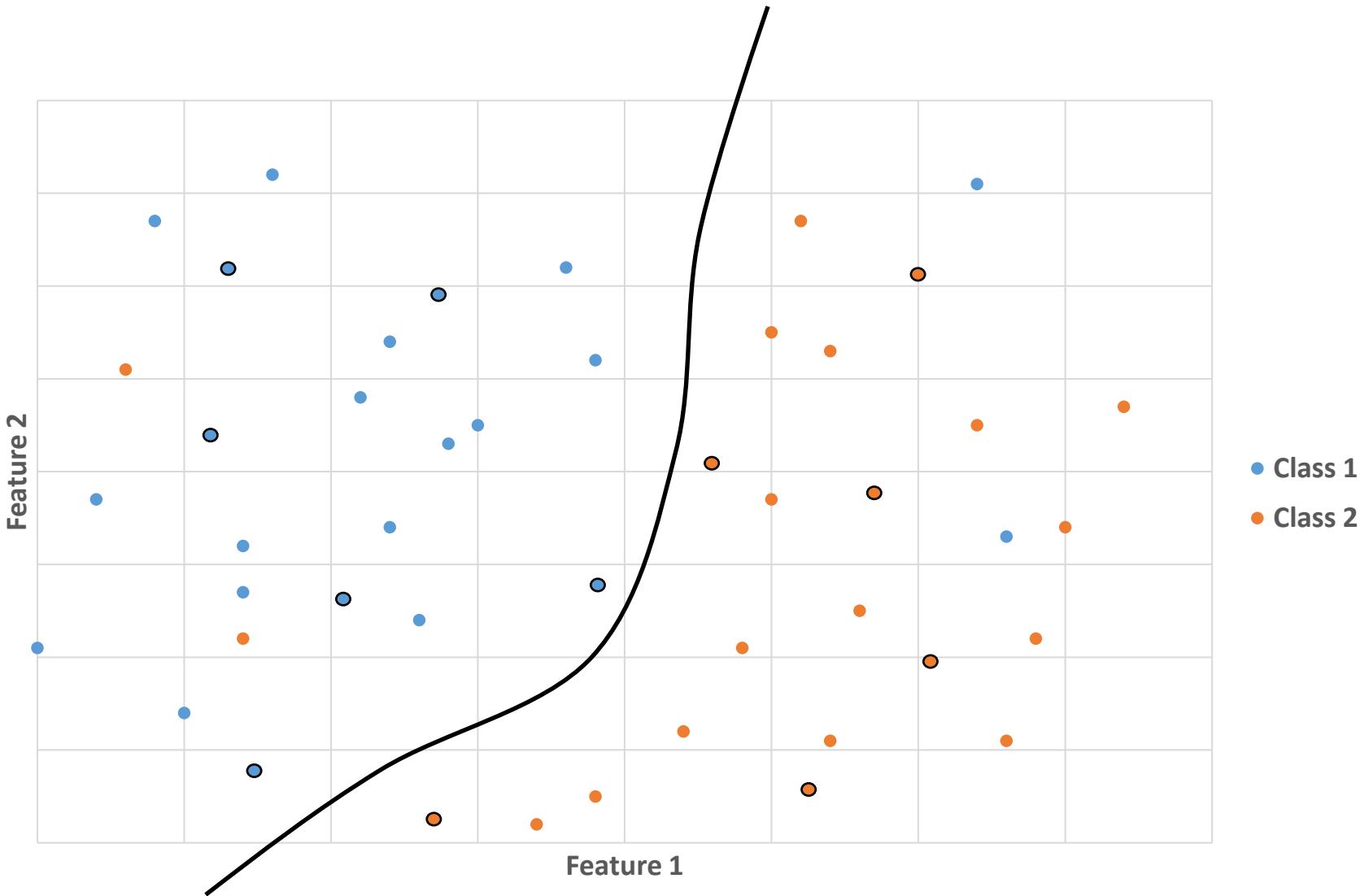
Training a classifier consists in estimating the "best" separation between classes

SHALLOW MACHINE LEARNING FOR PIXEL CLASSIFICATION



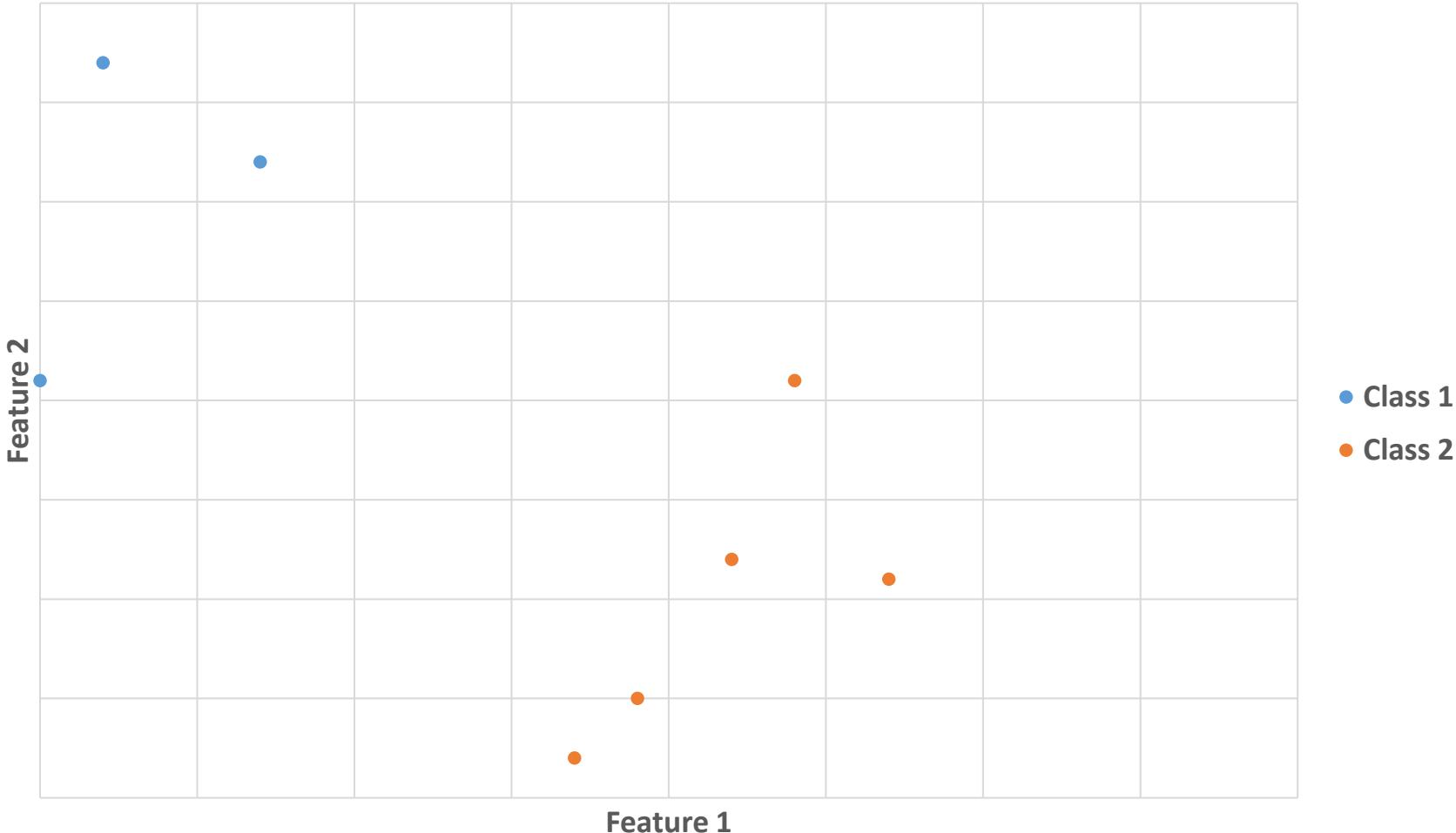
The **estimated class** for new data will be given by the **trained classifier**

SHALLOW MACHINE LEARNING FOR PIXEL CLASSIFICATION



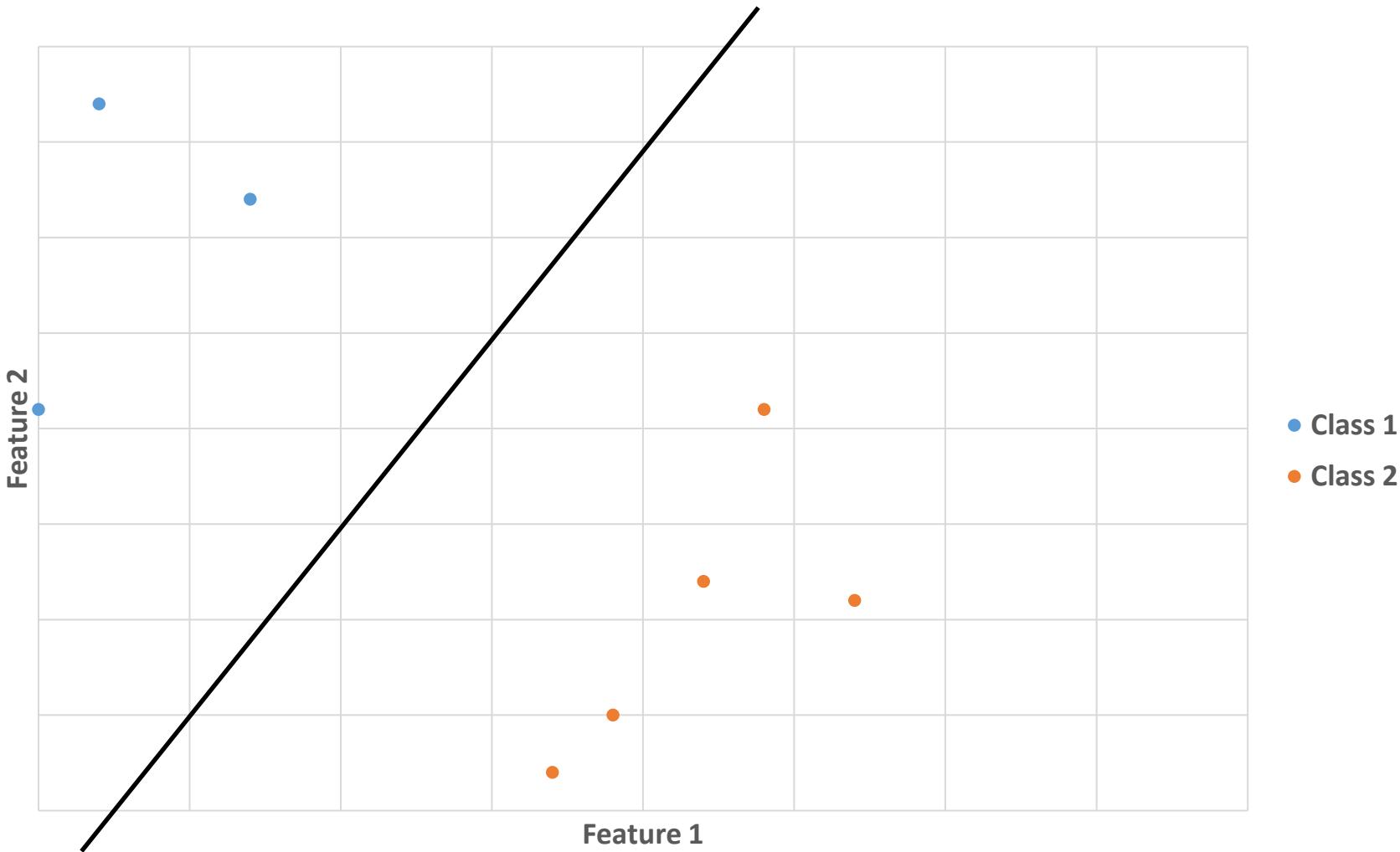
The **estimated class** for new data will be given by the **trained classifier**

SHALLOW MACHINE LEARNING FOR PIXEL CLASSIFICATION



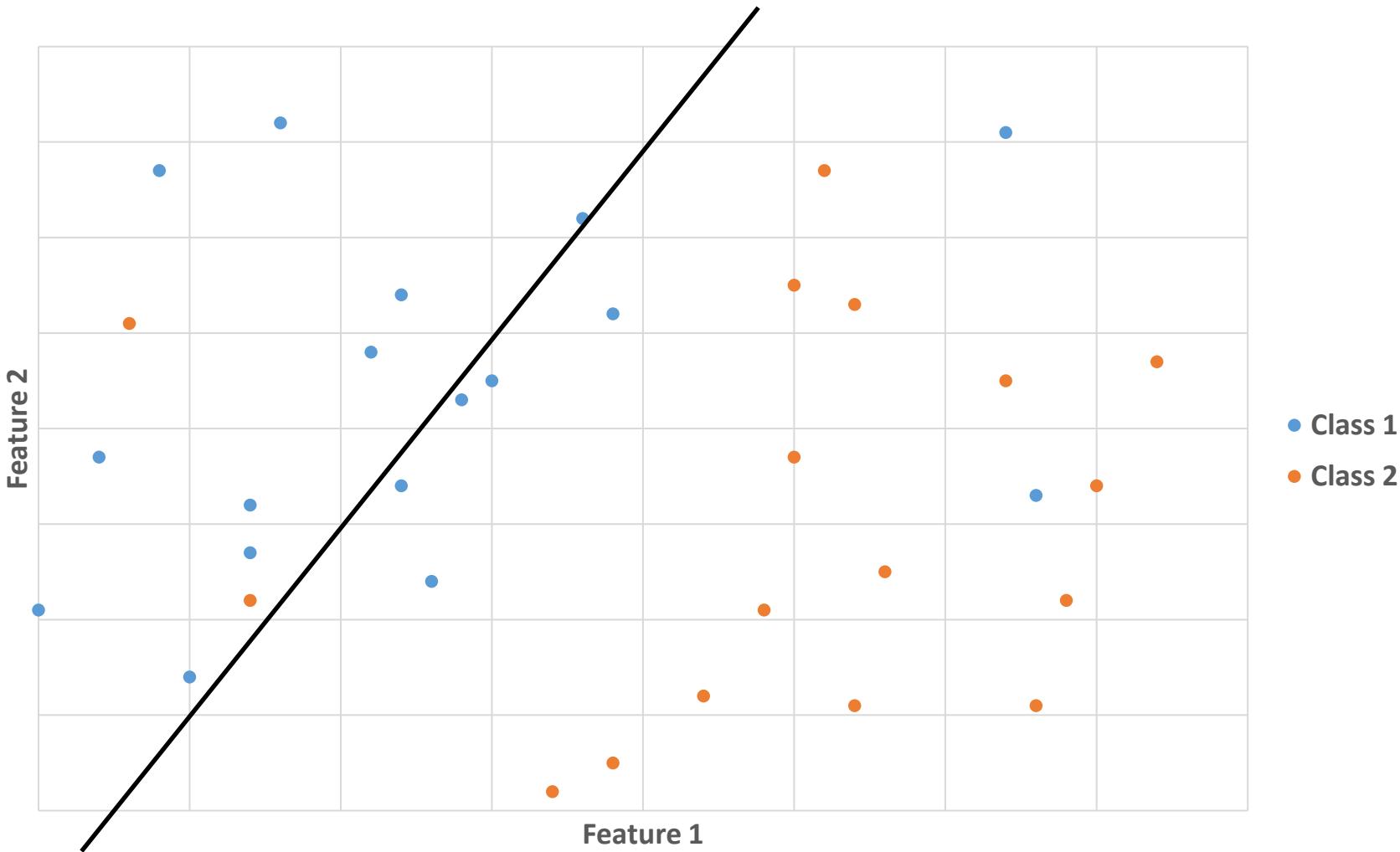
Training will be easier if **enough** and **representative** data is used

SHALLOW MACHINE LEARNING FOR PIXEL CLASSIFICATION



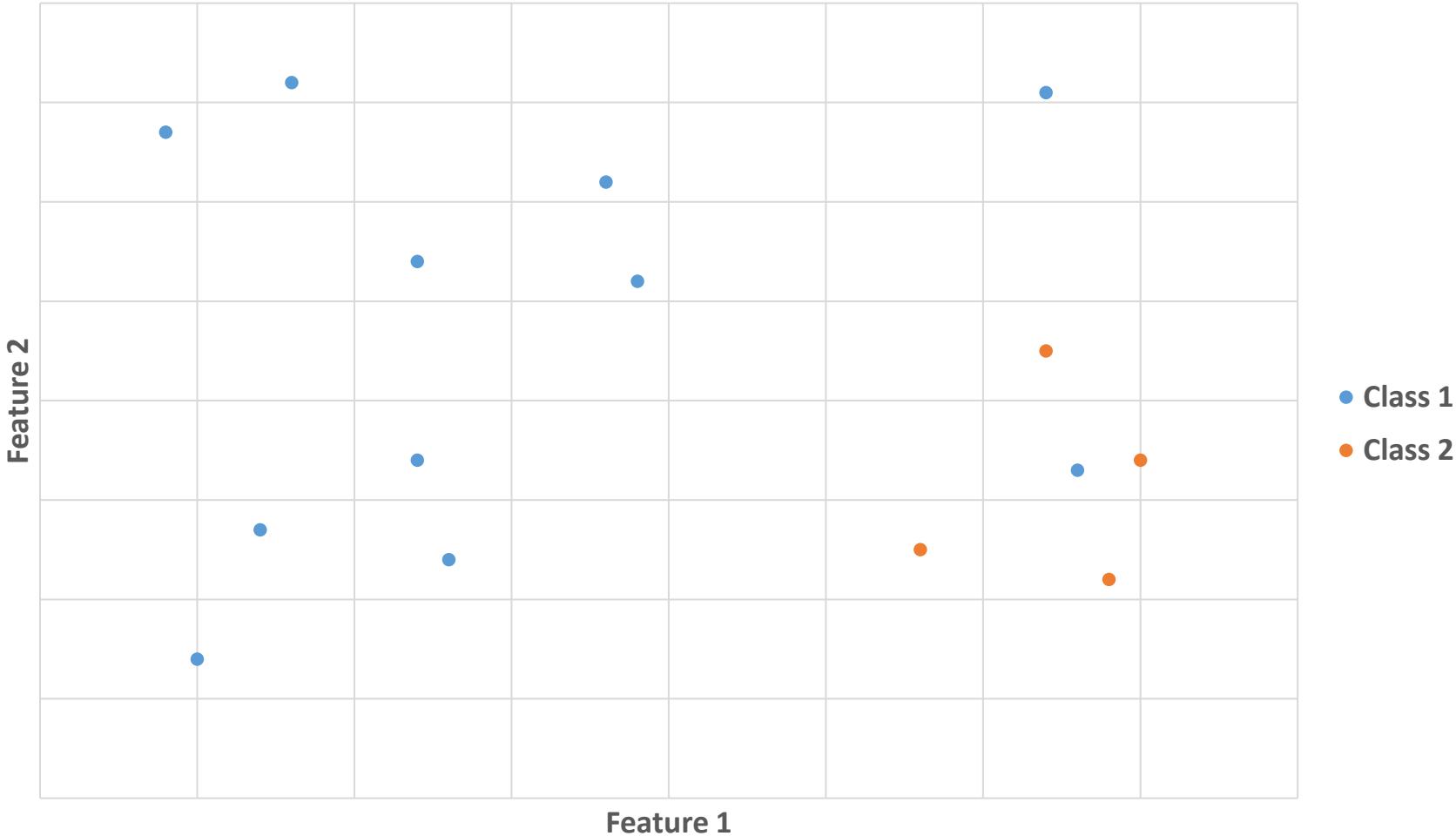
Training will be easier if **enough** and **representative** data is used

SHALLOW MACHINE LEARNING FOR PIXEL CLASSIFICATION



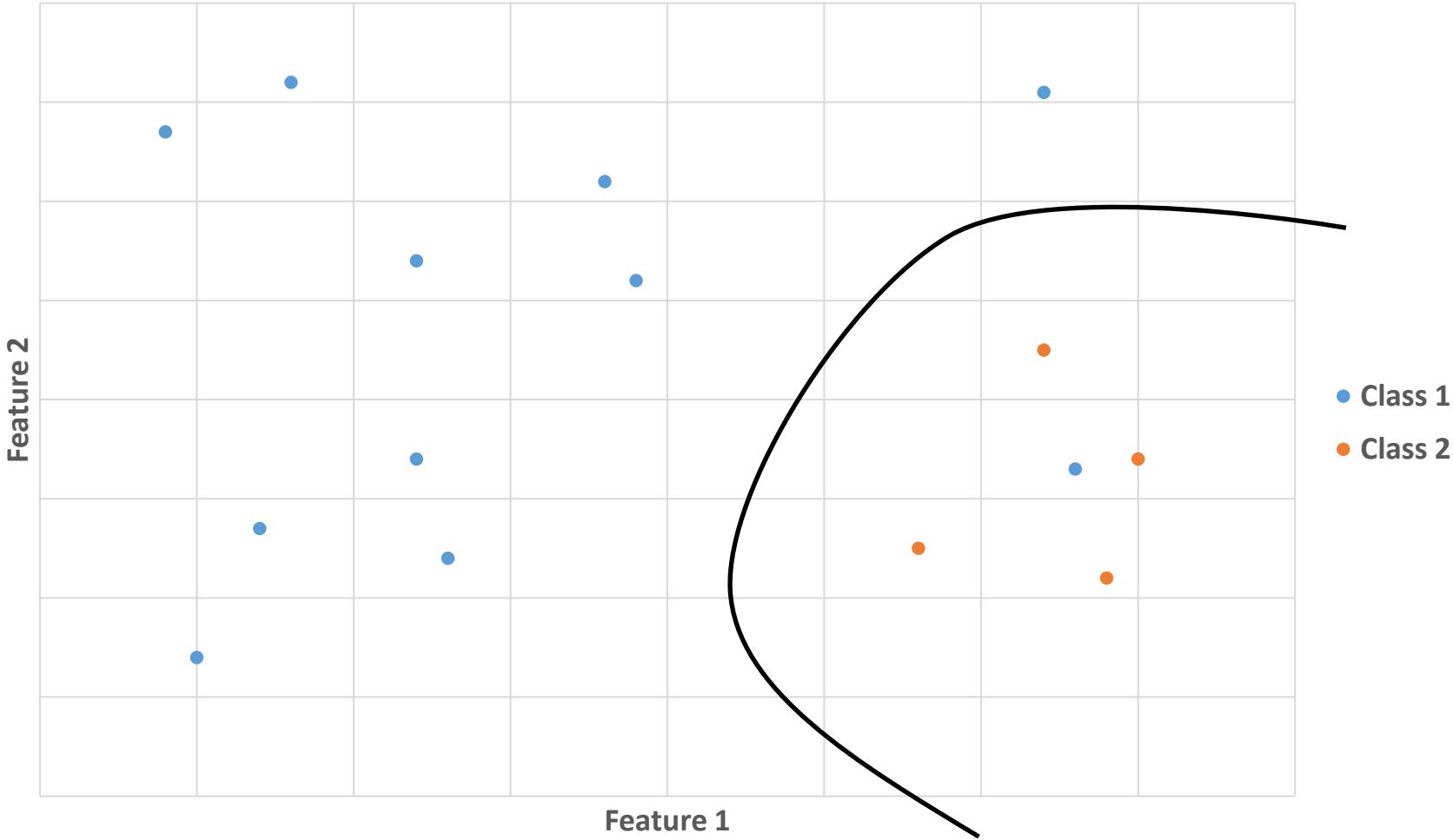
Training will be easier if **enough** and **representative** data is used

SHALLOW MACHINE LEARNING FOR PIXEL CLASSIFICATION



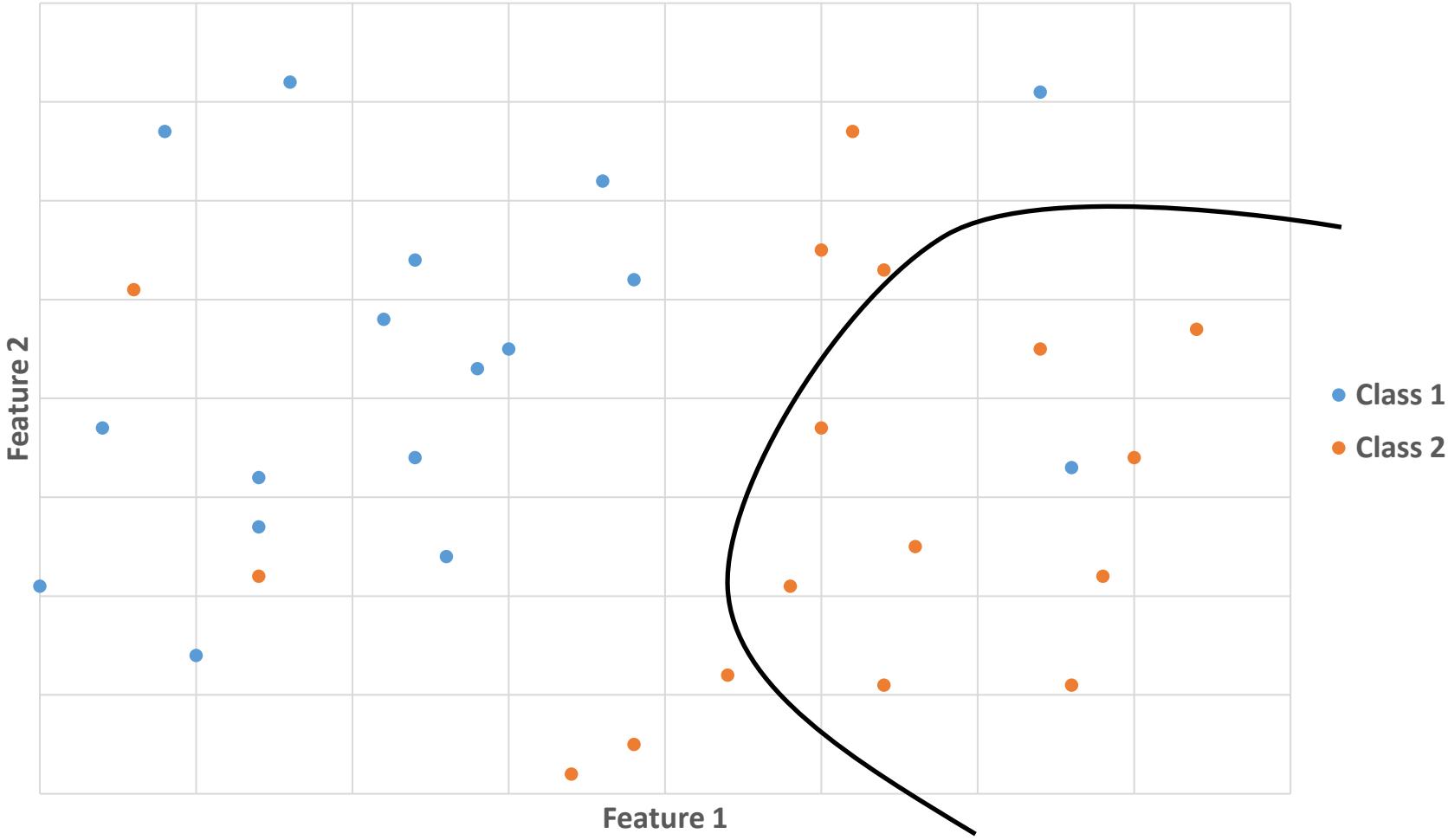
Class imbalance makes the training **more difficult**

SHALLOW MACHINE LEARNING FOR PIXEL CLASSIFICATION



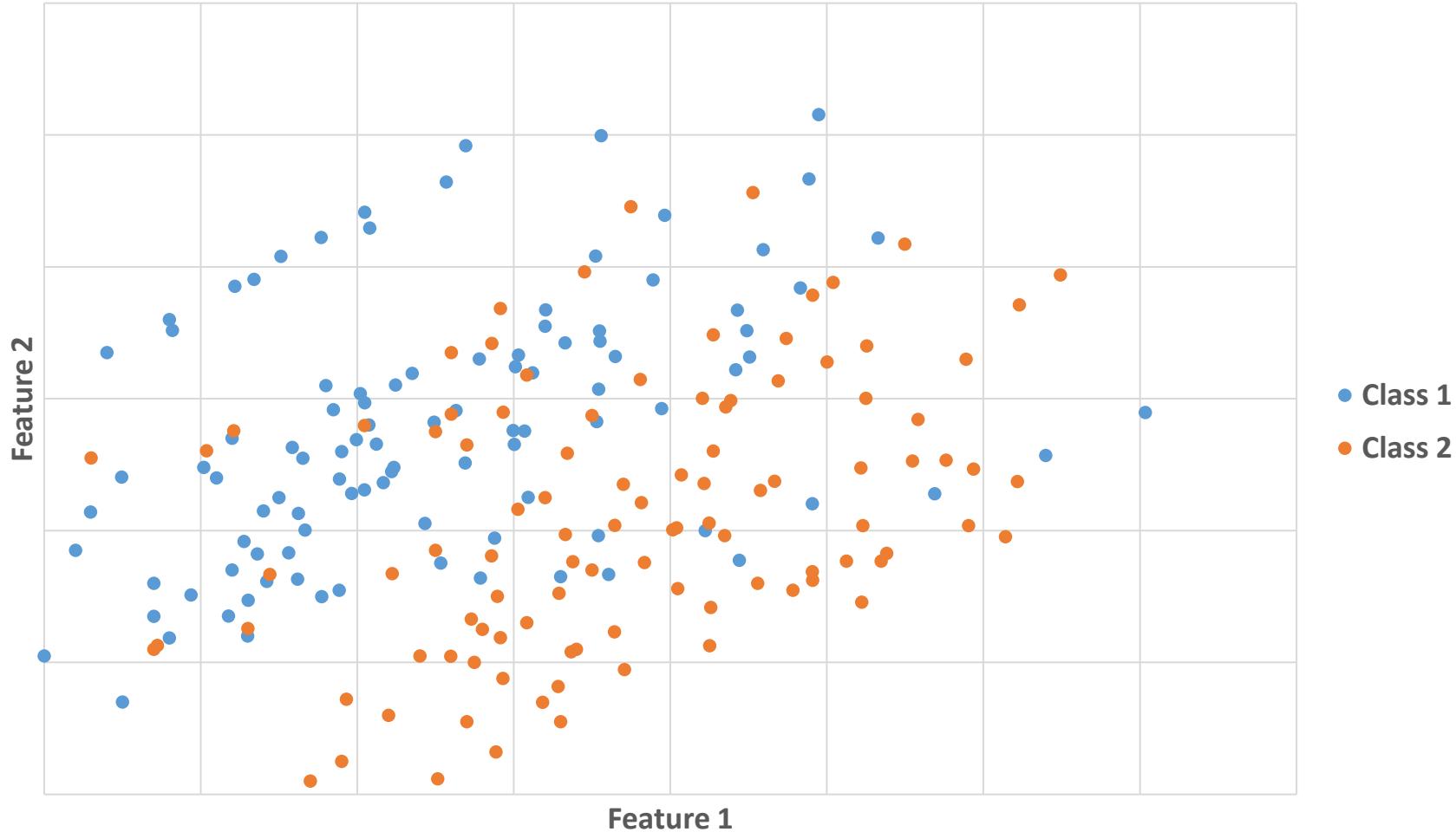
Class imbalance makes the training **more difficult**

SHALLOW MACHINE LEARNING FOR PIXEL CLASSIFICATION



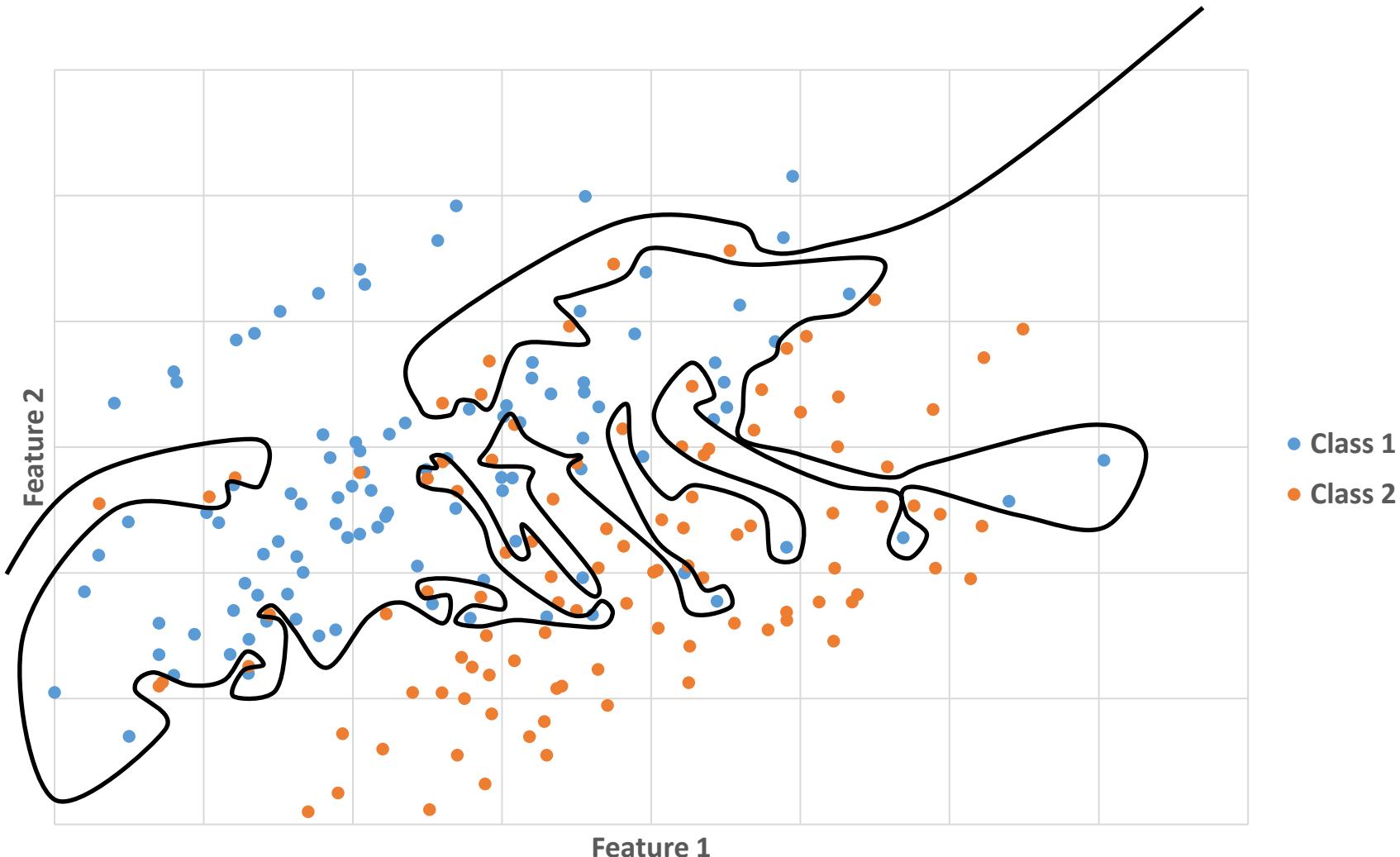
Class imbalance makes the training **more difficult**

SHALLOW MACHINE LEARNING FOR PIXEL CLASSIFICATION



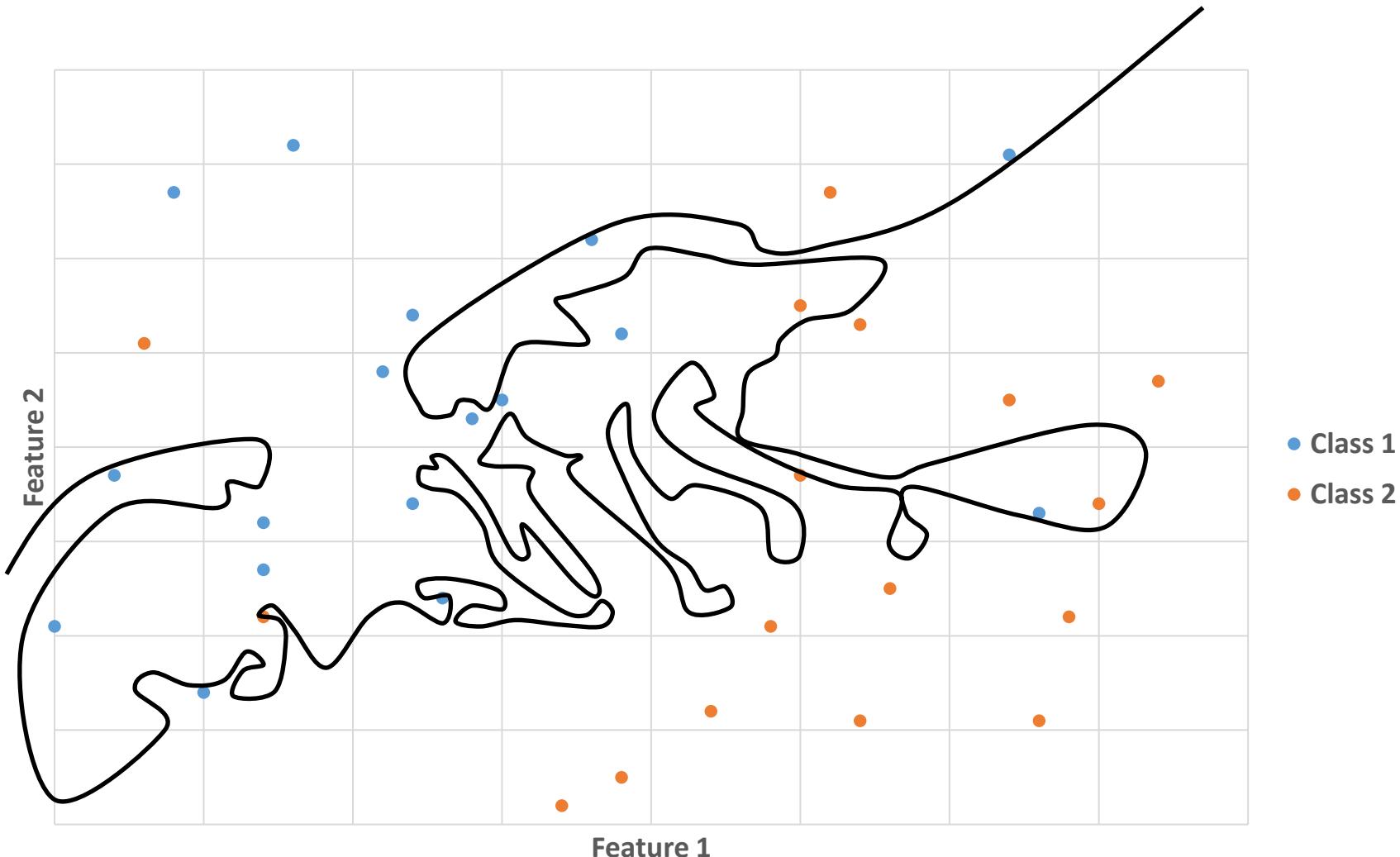
Too many annotations can lead to **over-fitting**: works great on training data but poorly on new data

SHALLOW MACHINE LEARNING FOR PIXEL CLASSIFICATION



Too many annotations can lead to **over-fitting**: works great on training data but poorly on new data

SHALLOW MACHINE LEARNING FOR PIXEL CLASSIFICATION



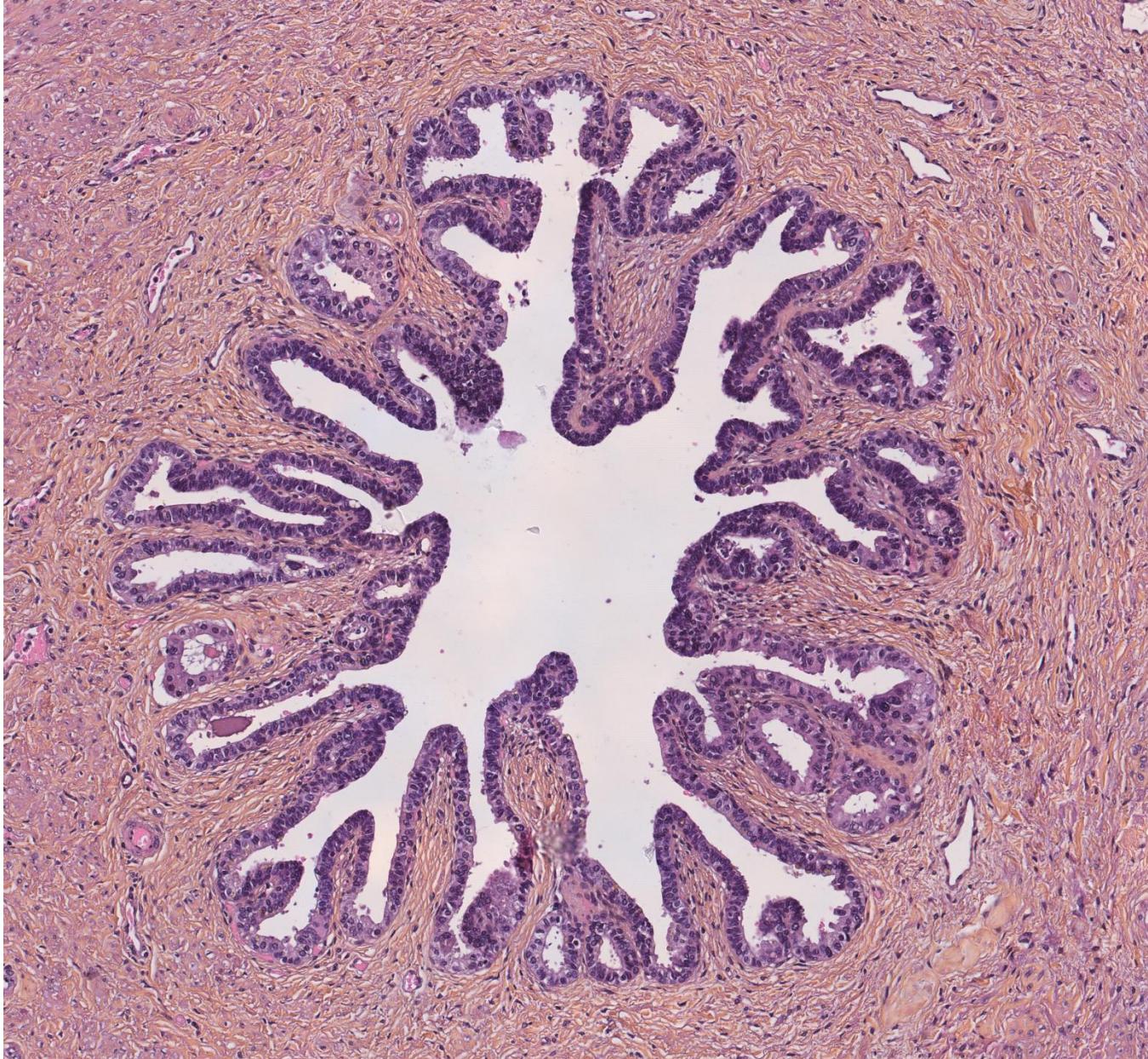
Too many annotations can lead to **over-fitting**: works great on training data but poorly on new data

SHALLOW MACHINE LEARNING FOR PIXEL CLASSIFICATION

- Select **image features appropriate** to the classification problem
- Manually annotate regions/objects that are **representative** of what is seen in images
- Use **V** tool for annotations to **avoid over-representation**
- Define roughly the **same amount** of annotations for **each class**
- **Do not** manually annotate an **entire region of slide** to avoid over-fitting

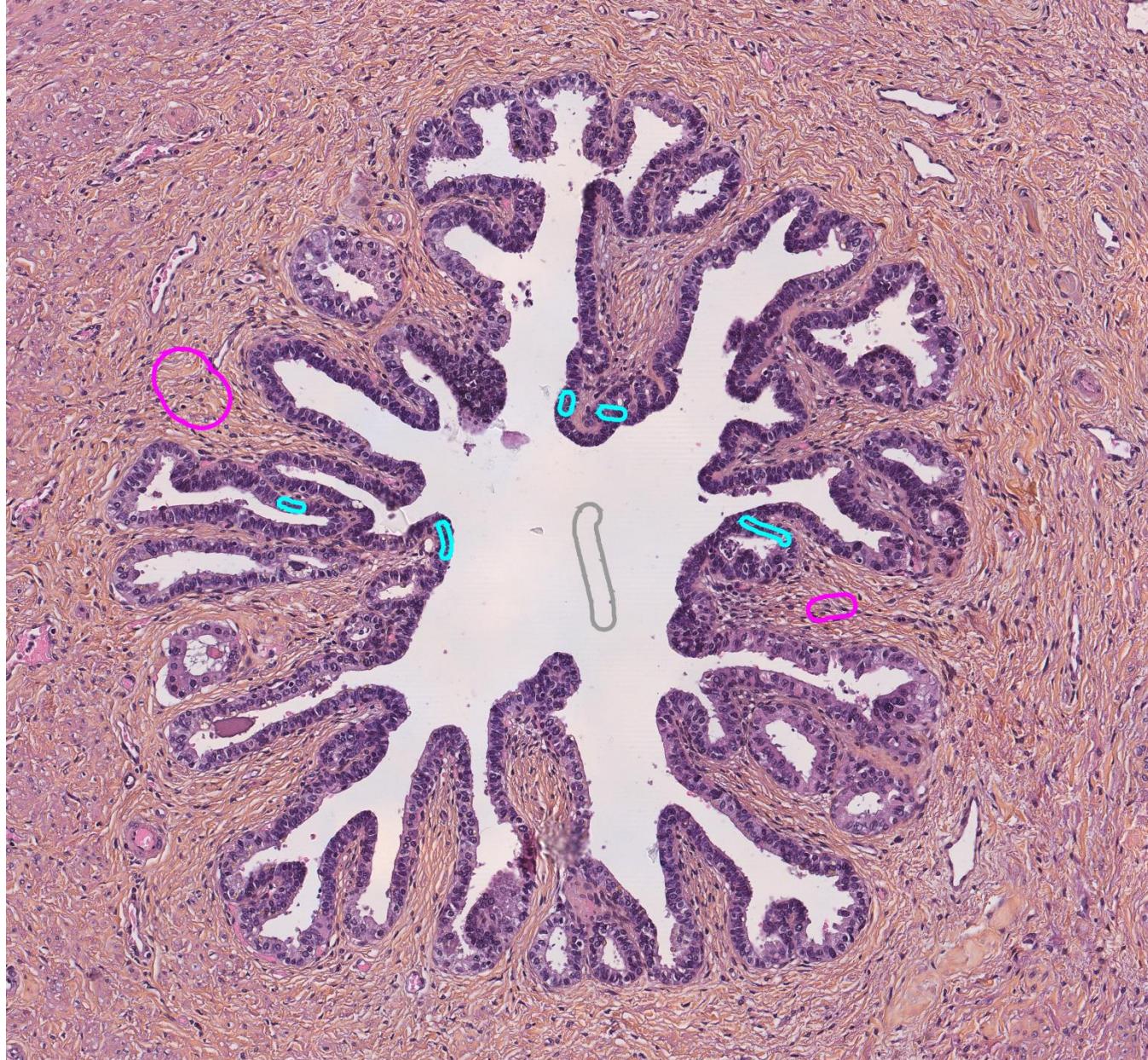
PIXEL CLASSIFICATION

Find **regions** corresponding to **epithelium**, **stroma** and **background**



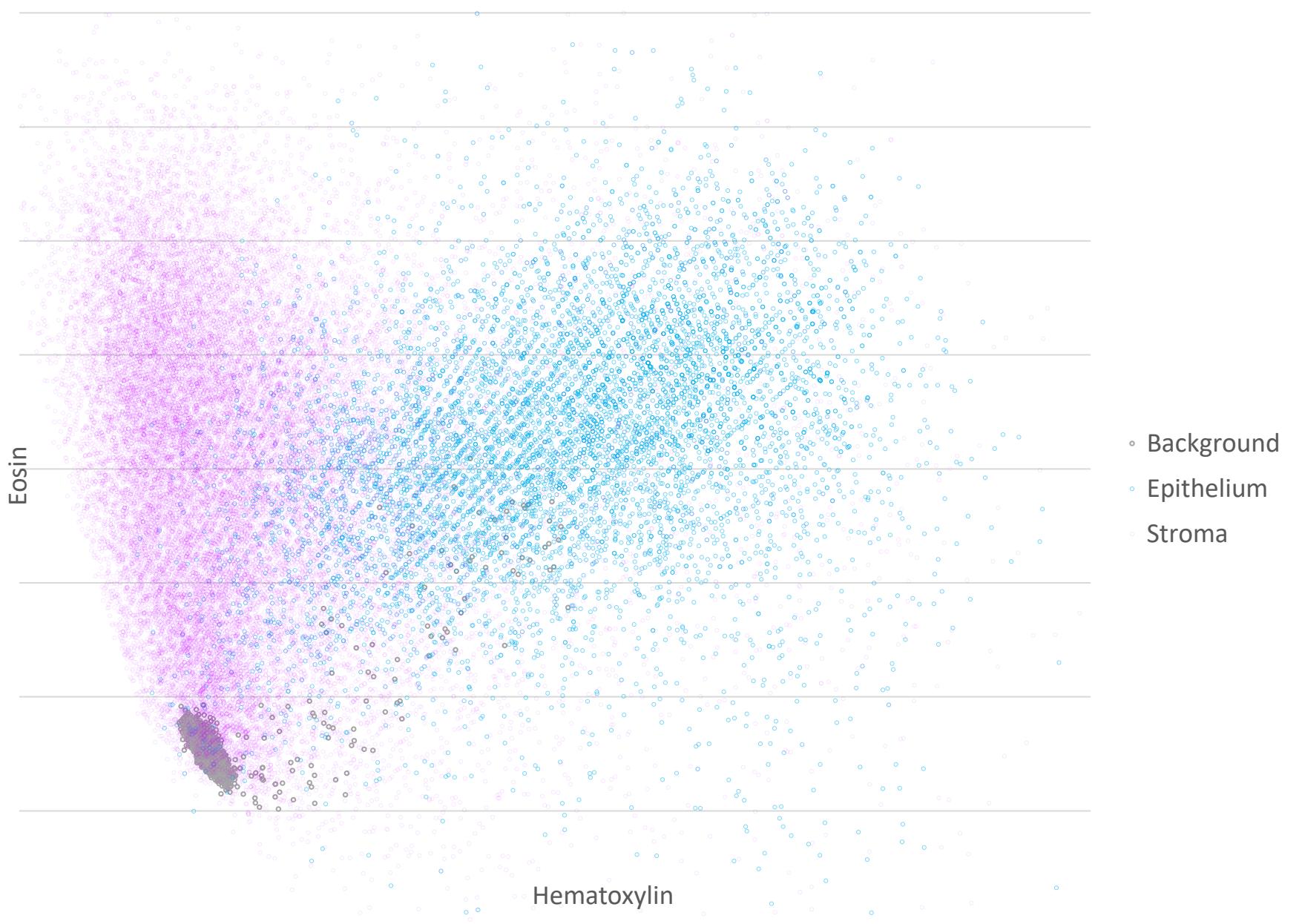
PIXEL CLASSIFICATION

Find **regions** corresponding to **epithelium**, **stroma** and **background**

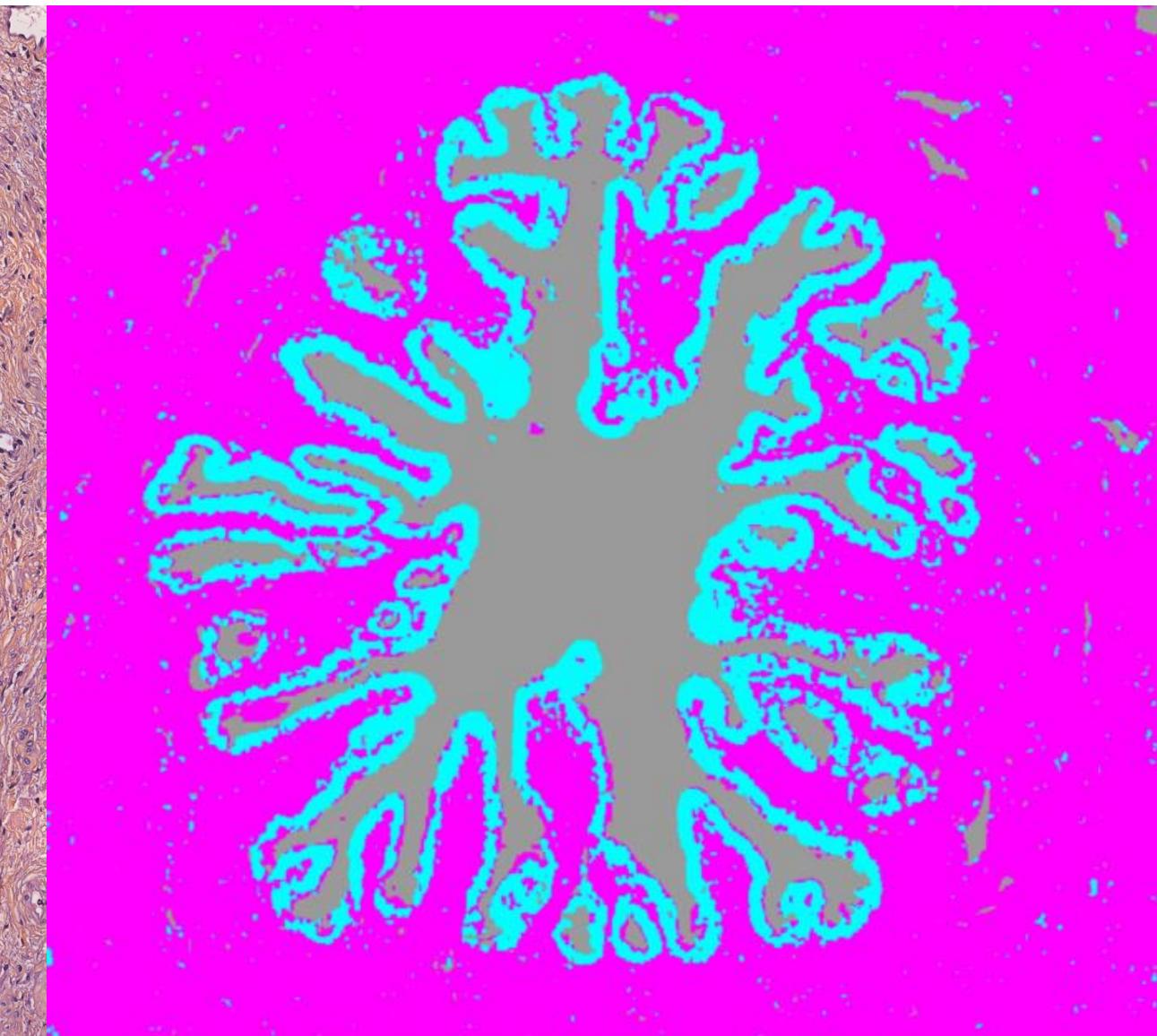
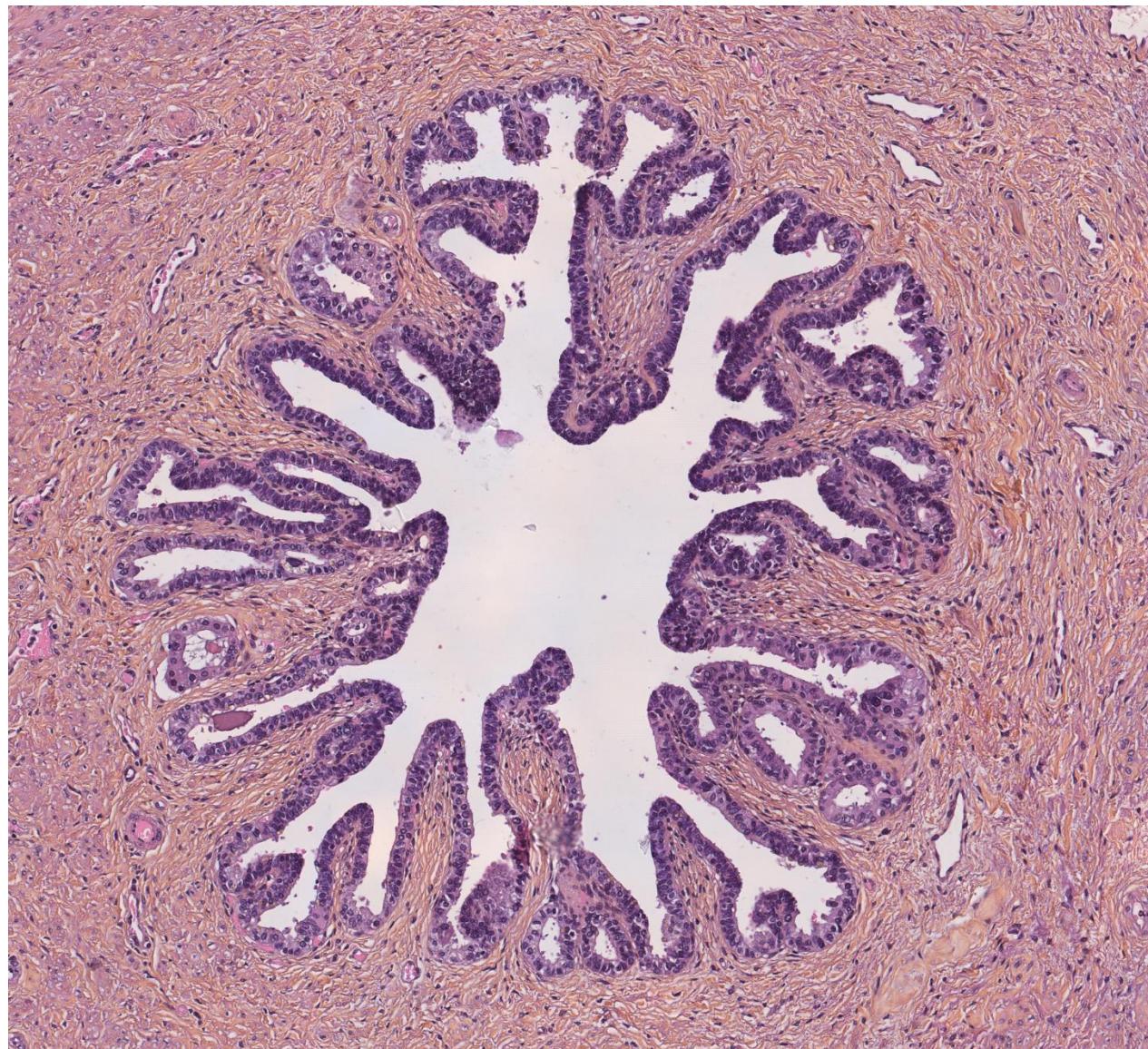


PIXEL CLASSIFICATION

Find **regions** corresponding to **epithelium**, **stroma** and **background**

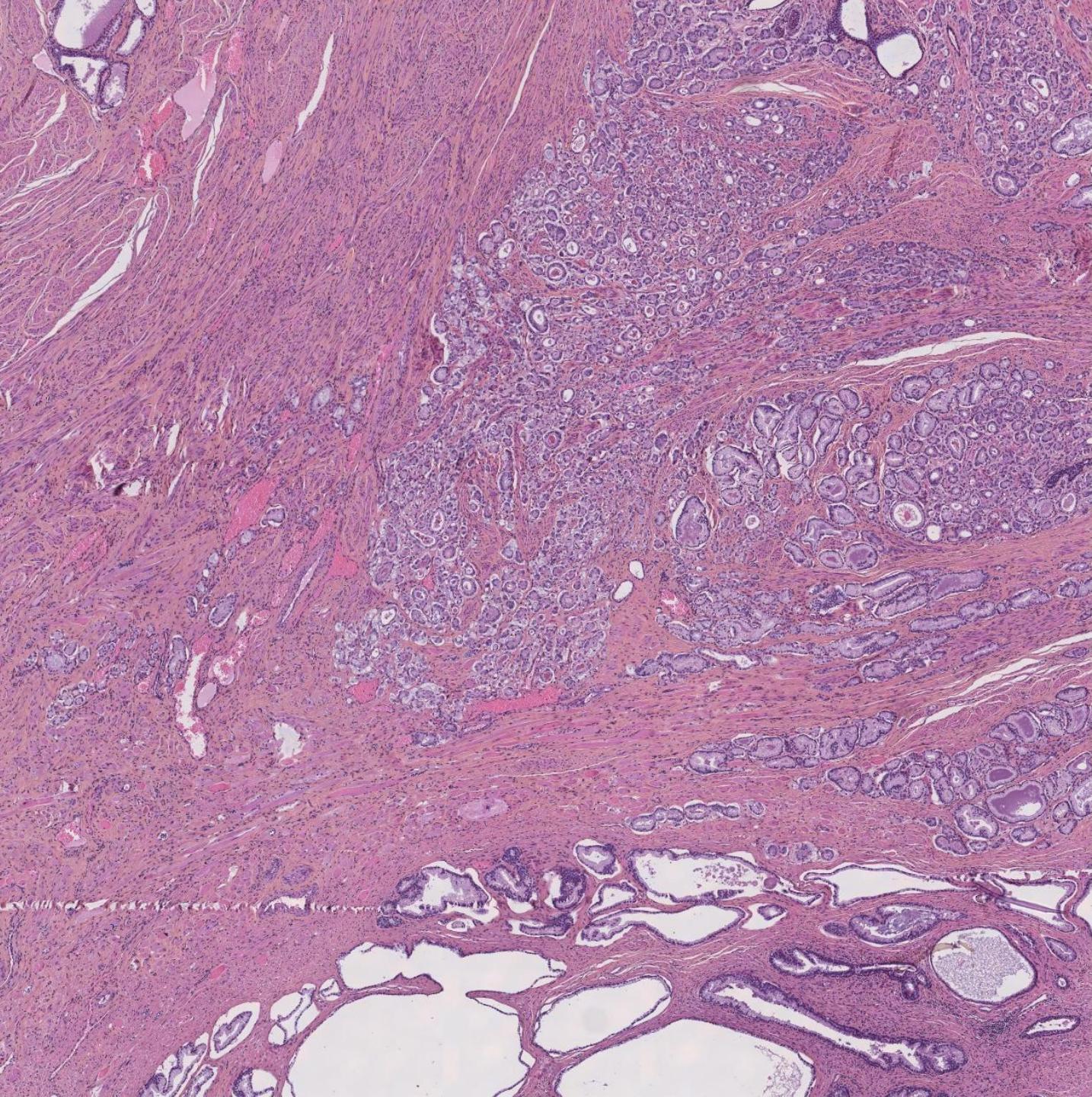


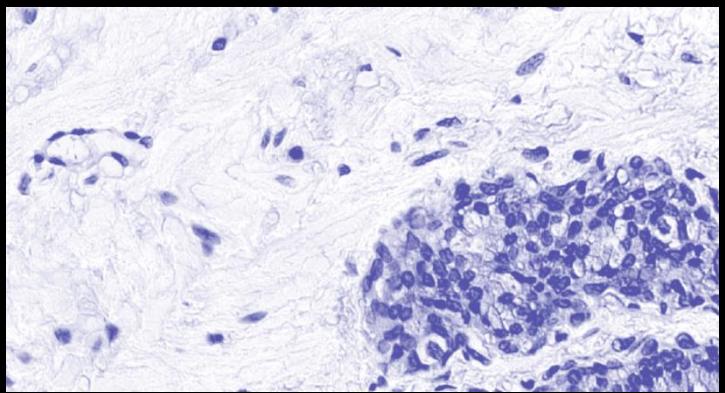
PIXEL CLASSIFICATION



PIXEL CLASSIFICATION

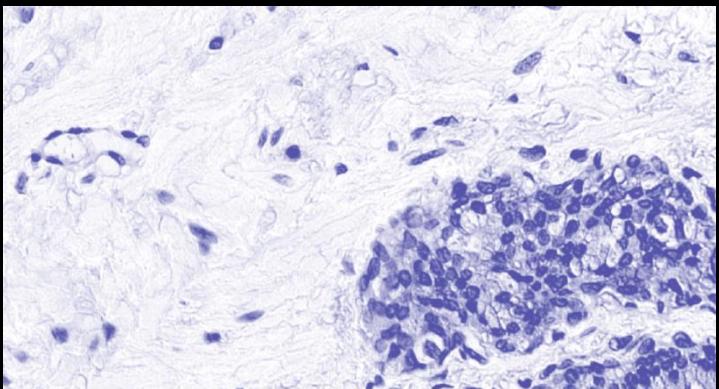
- Open prostate_1.ome.tif, prostate_2.ome.tif, prostate_3.ome.tif, prostate_4.ome.tif, prostate_5.ome.tif and prostate_6.ome.tif
- Create **annotations** in each image that recapitulate the **diversity** of the tissue
- Create **regions annotations**
- Open "Pixel classifier"
- Annotate pixels belonging to **background**, **epithelium** and **stroma**
- Save classifier and apply it to each image with a script (workflow tab)
- Get proportions of tissues



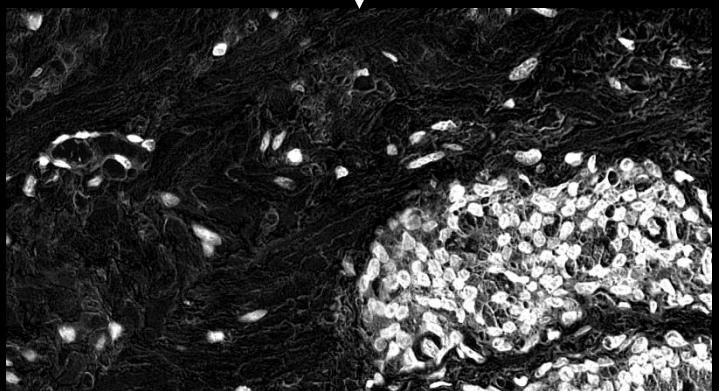


CELL DETECTION

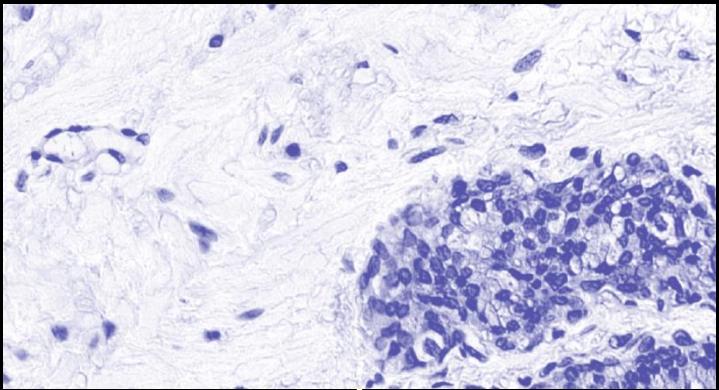
CELL DETECTION



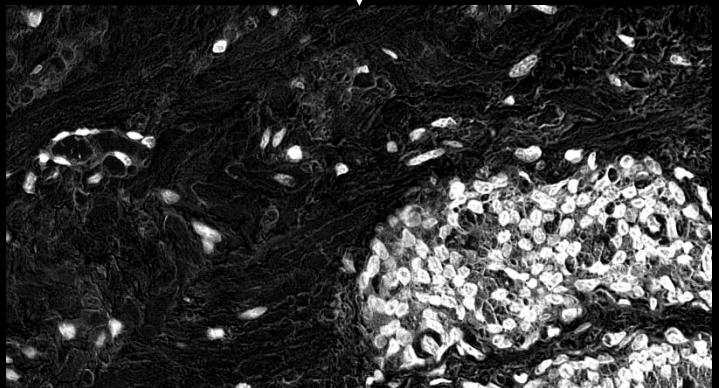
Gray levels



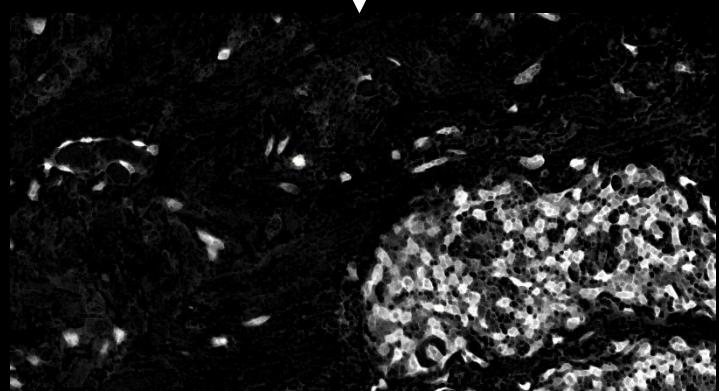
CELL DETECTION



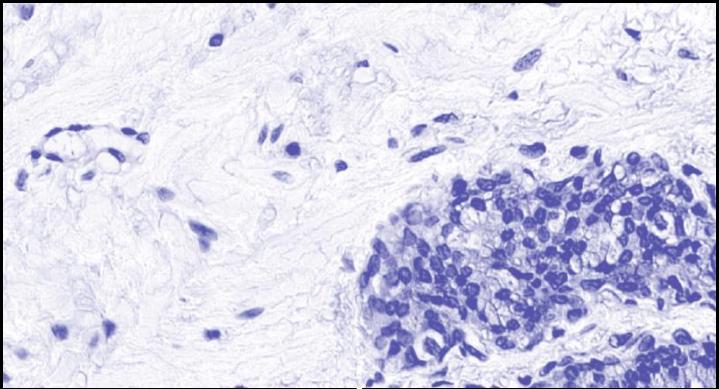
Gray levels



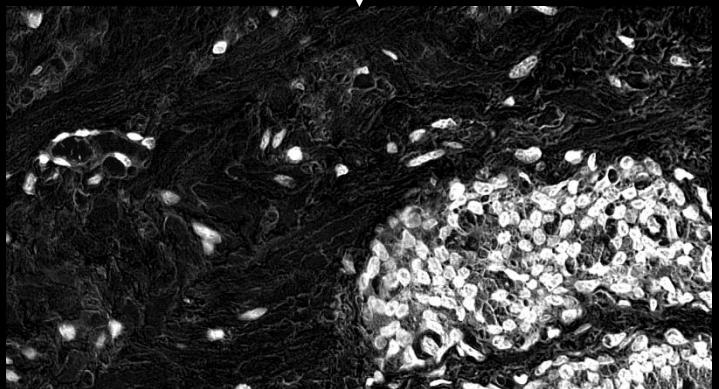
Minimum filtering



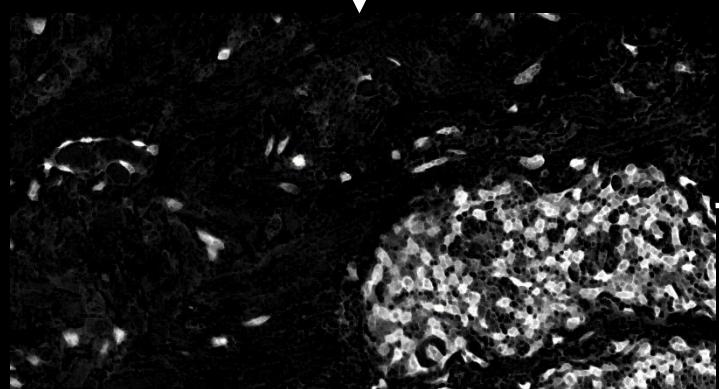
CELL DETECTION



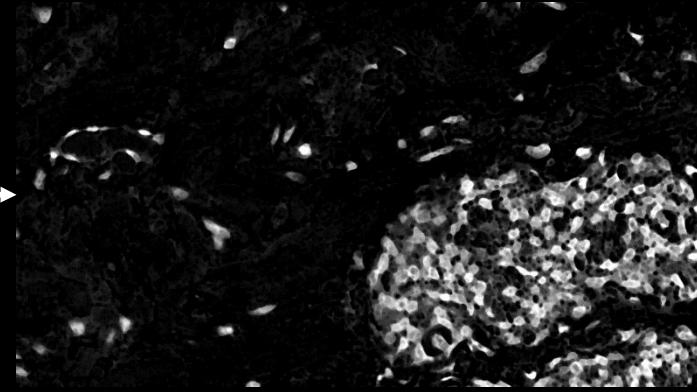
Gray levels

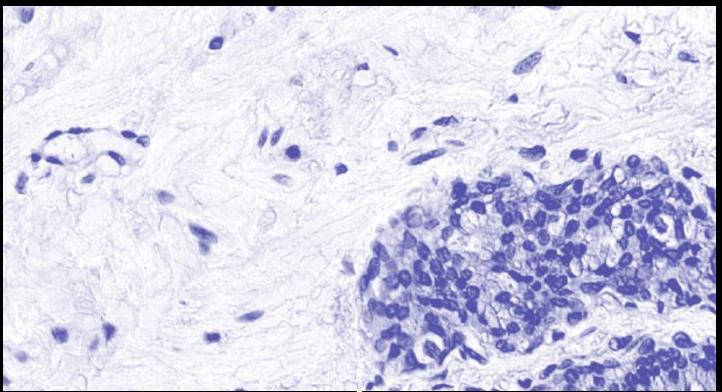


Minimum filtering

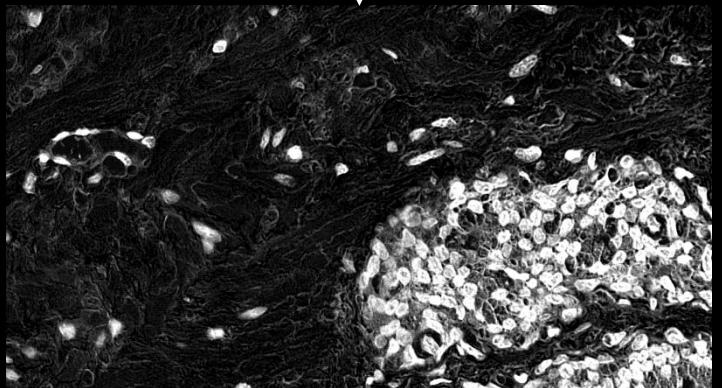


Gaussian blur

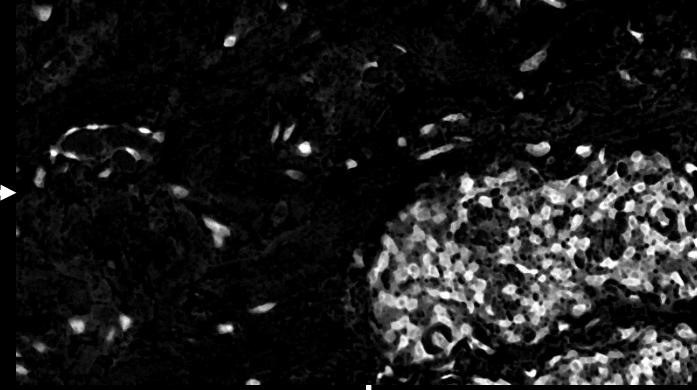
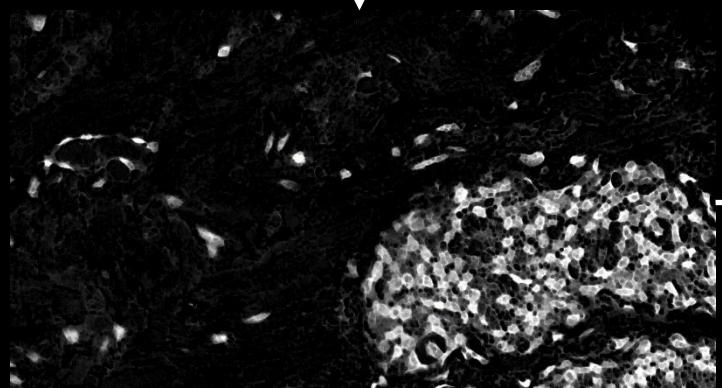




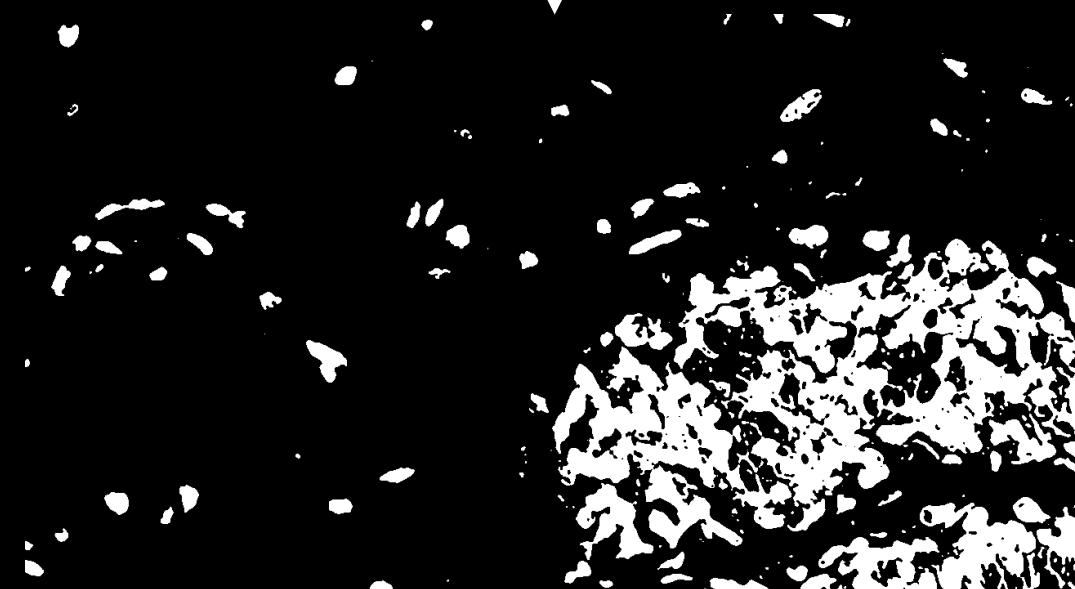
CELL DETECTION



Gaussian blur

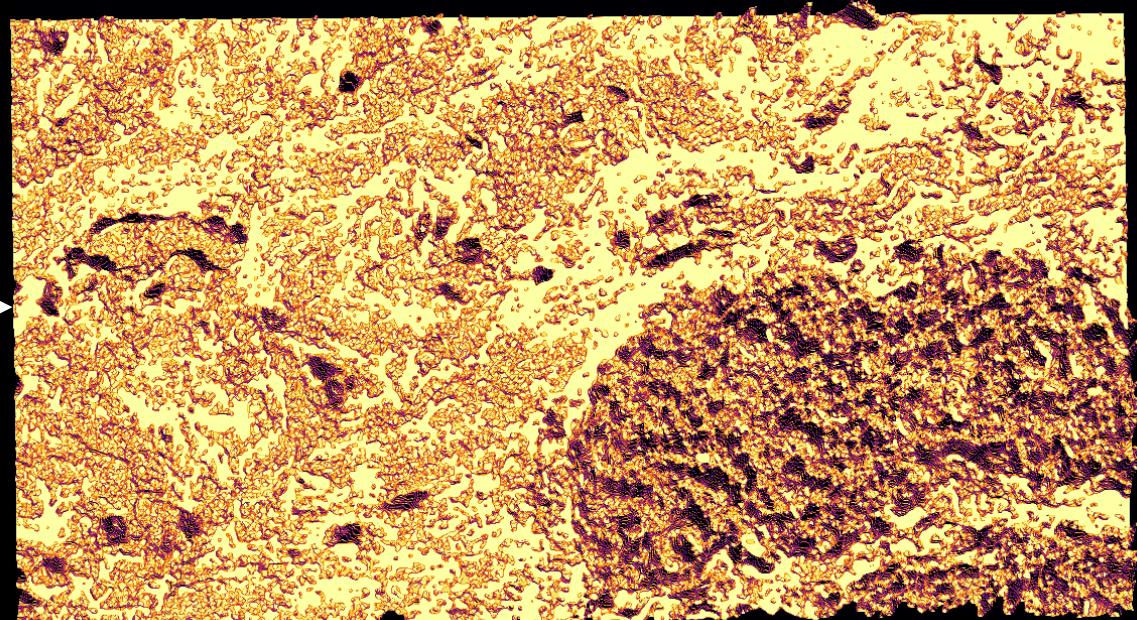
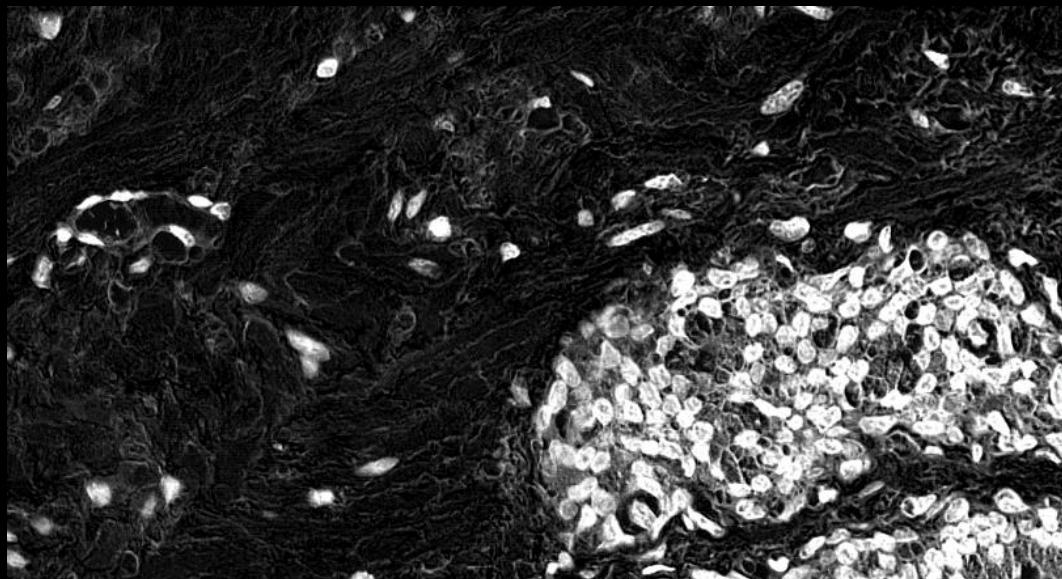


Thresholding



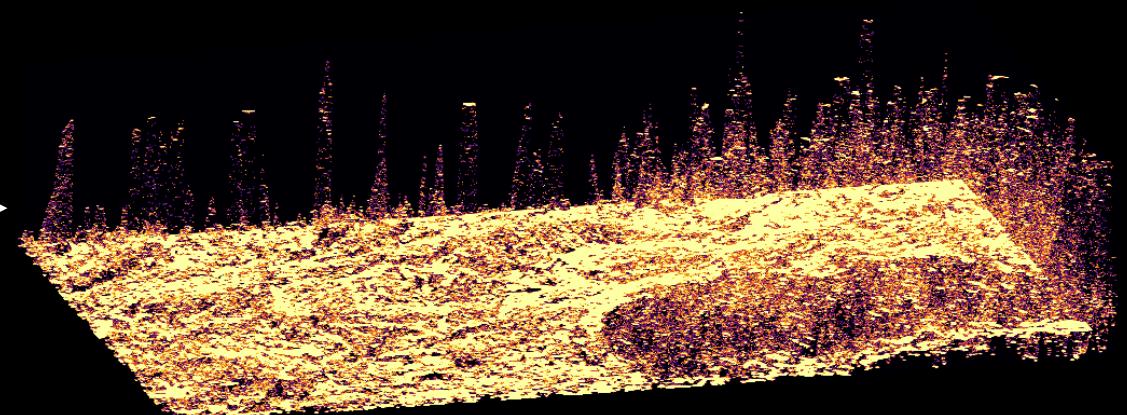
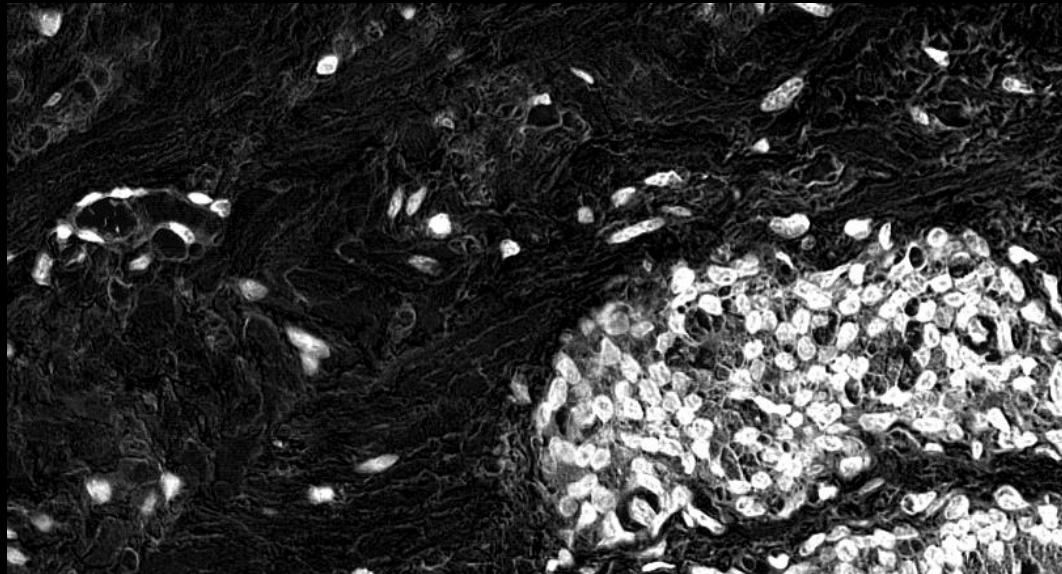
WATERSHED

Transform image so that intensity becomes 3rd dimension



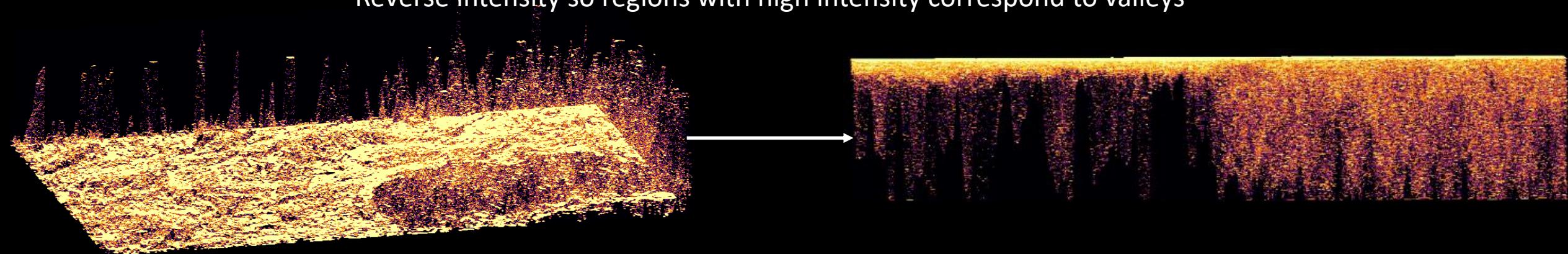
WATERSHED

Transform image so that intensity becomes 3rd dimension

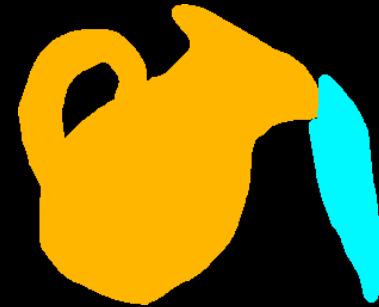


WATERSHED

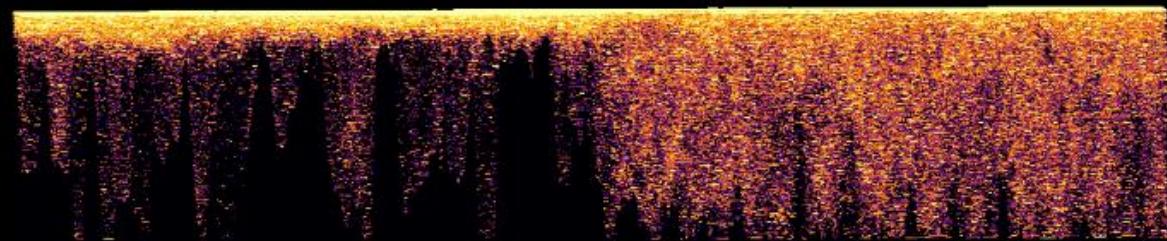
Reverse intensity so regions with high intensity correspond to valleys



WATERSHED

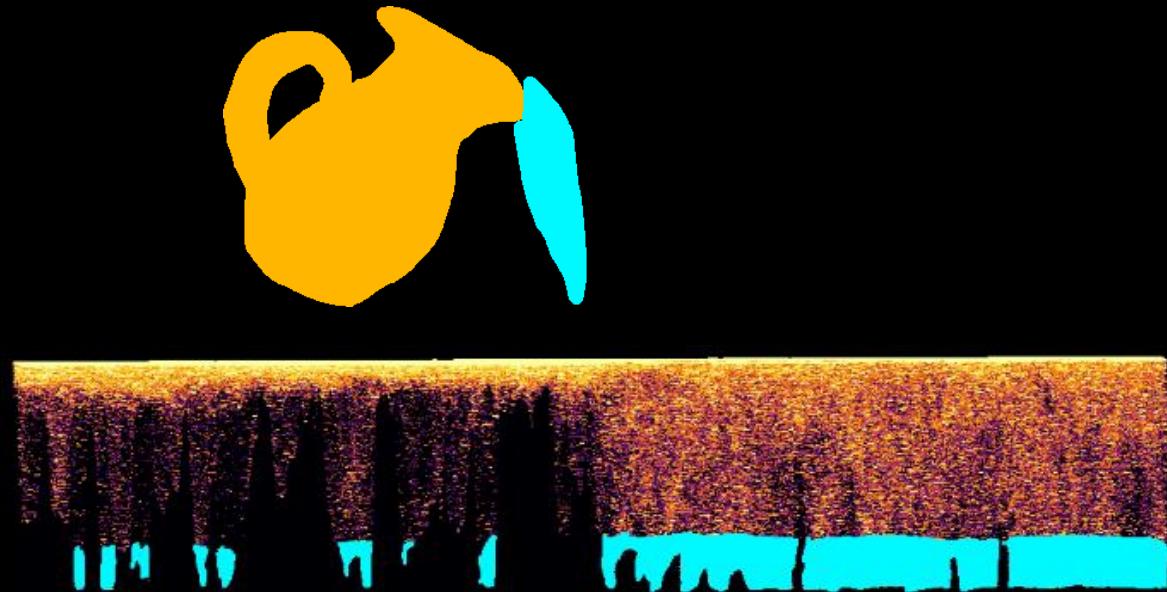


Pour water into valleys



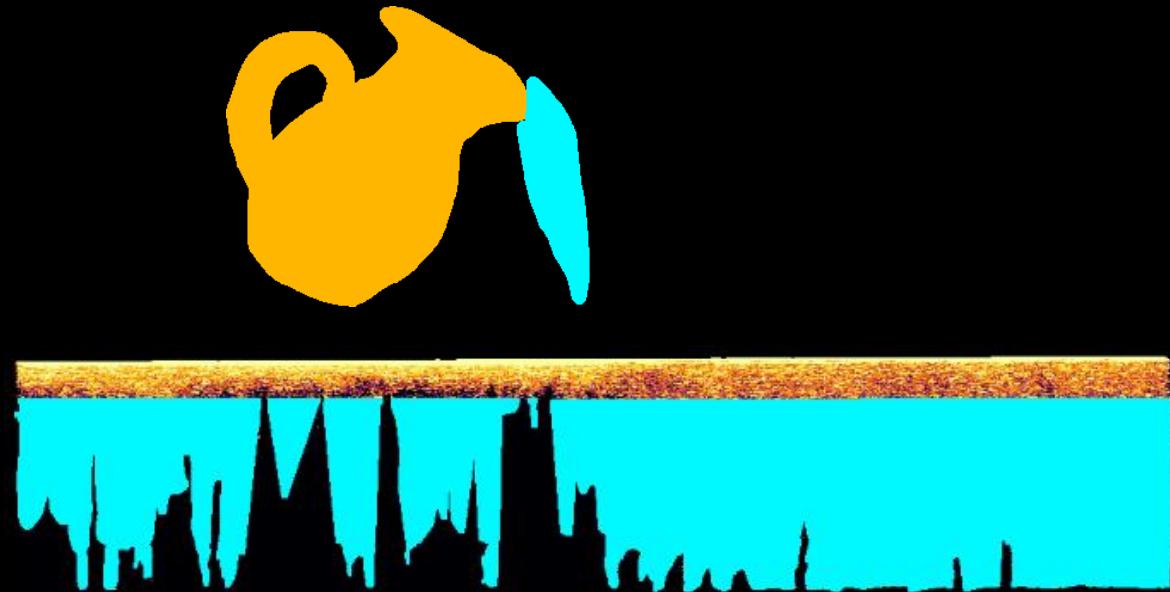
WATERSHED

Pour water into valleys



WATERSHED

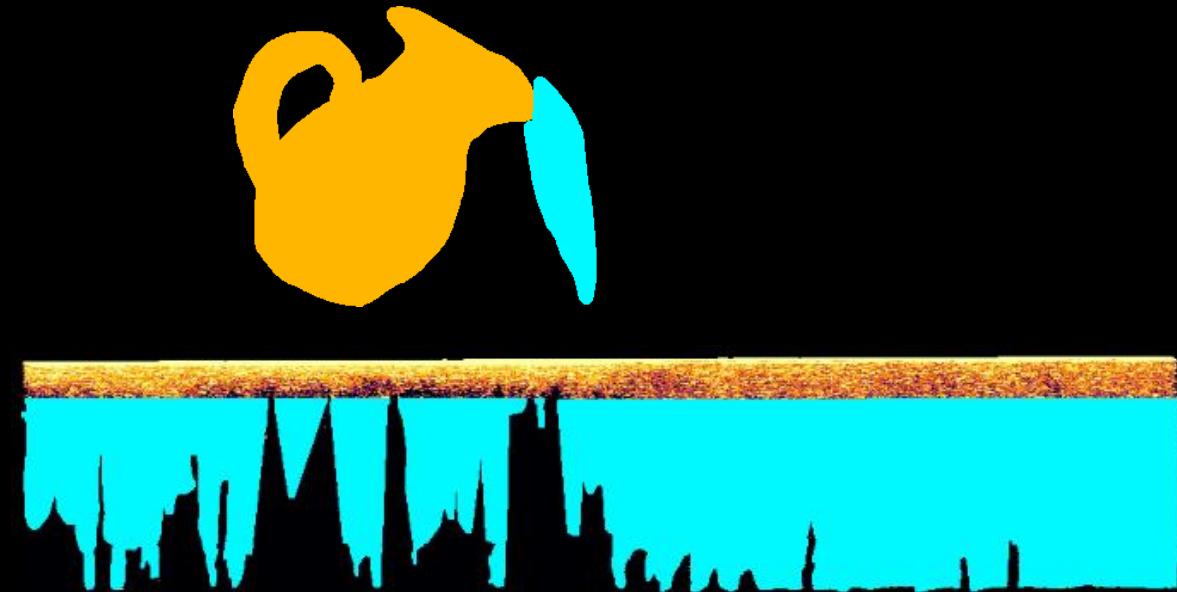
Pour water into valleys



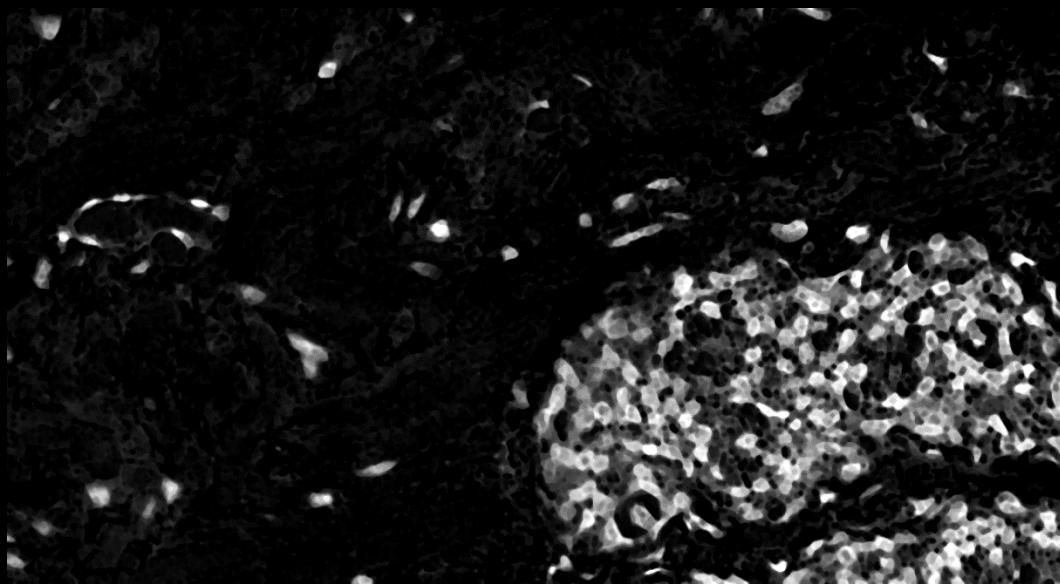
WATERSHED

Pour water into valleys

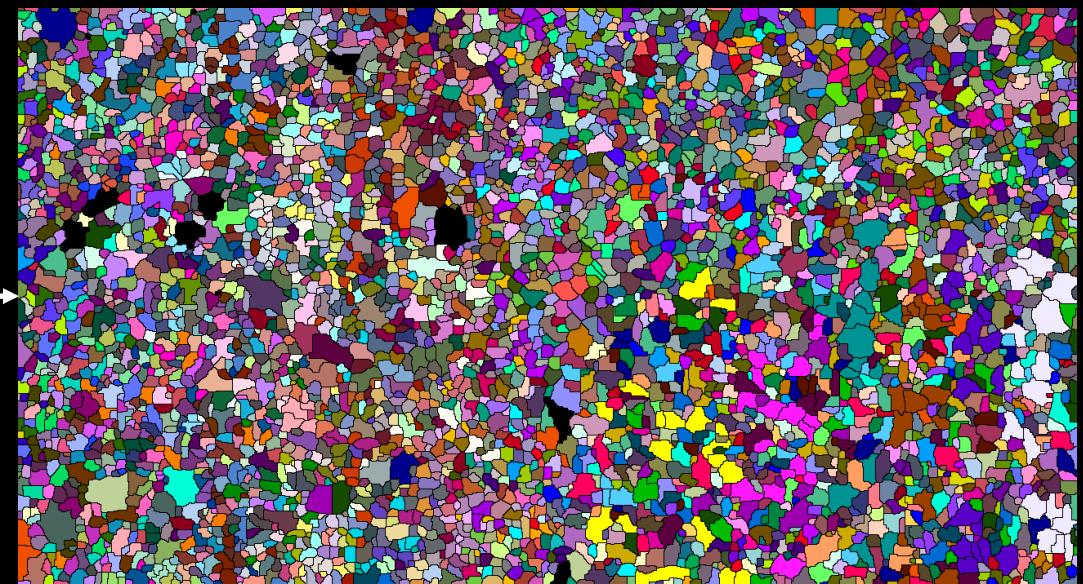
Threshold



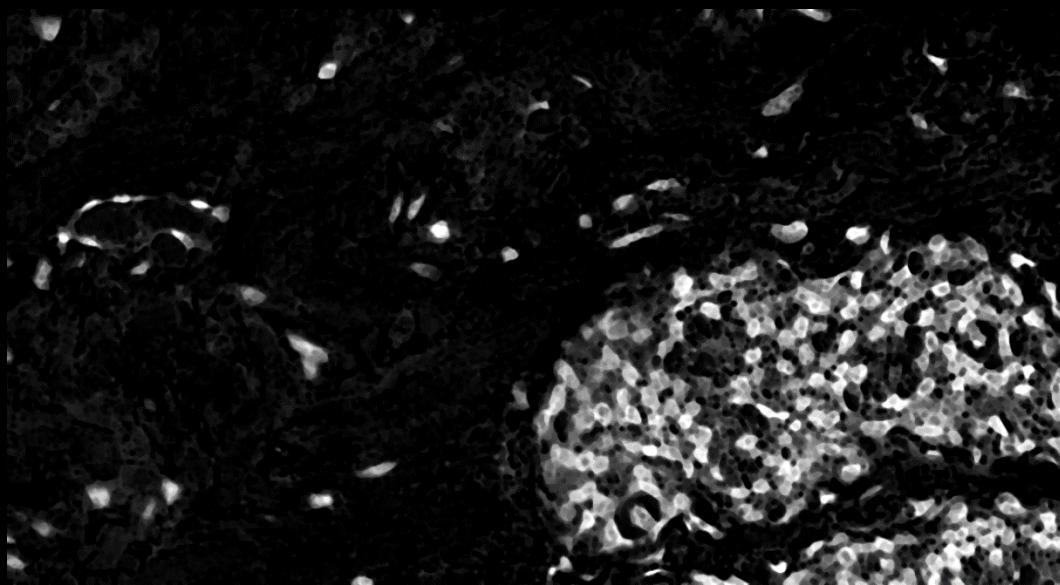
WATERSHED



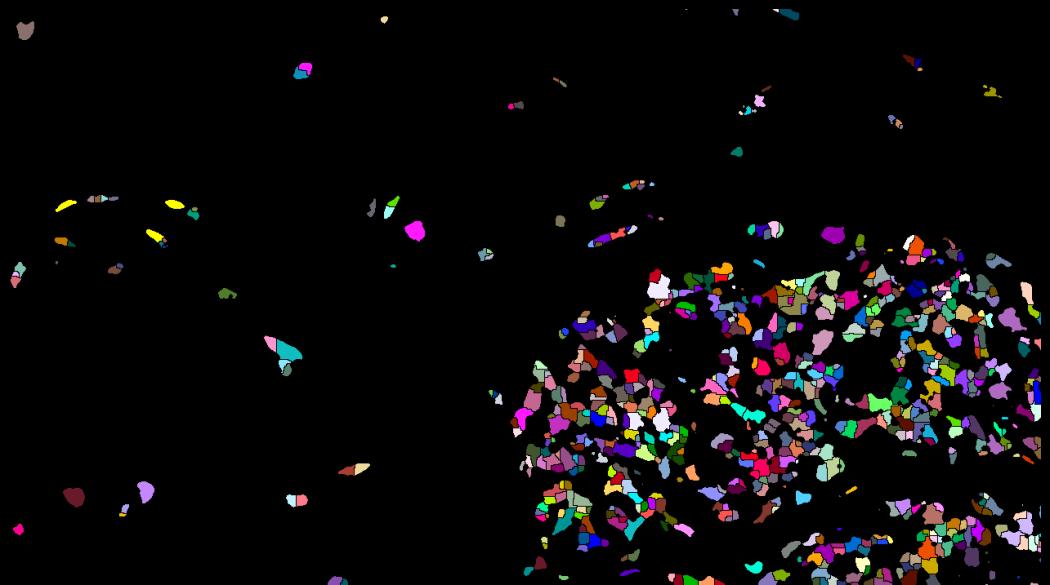
Watershed



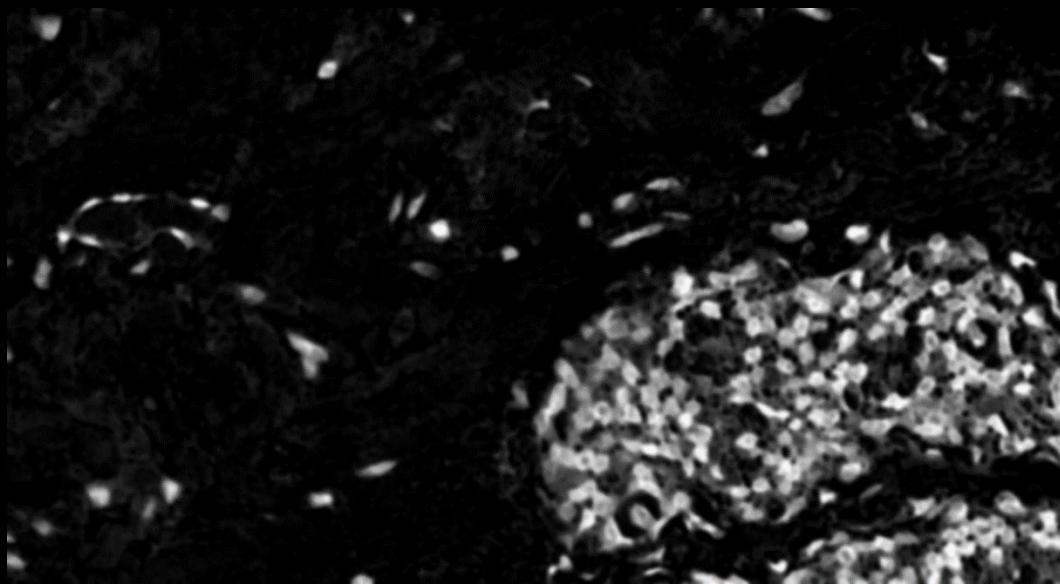
WATERSHED



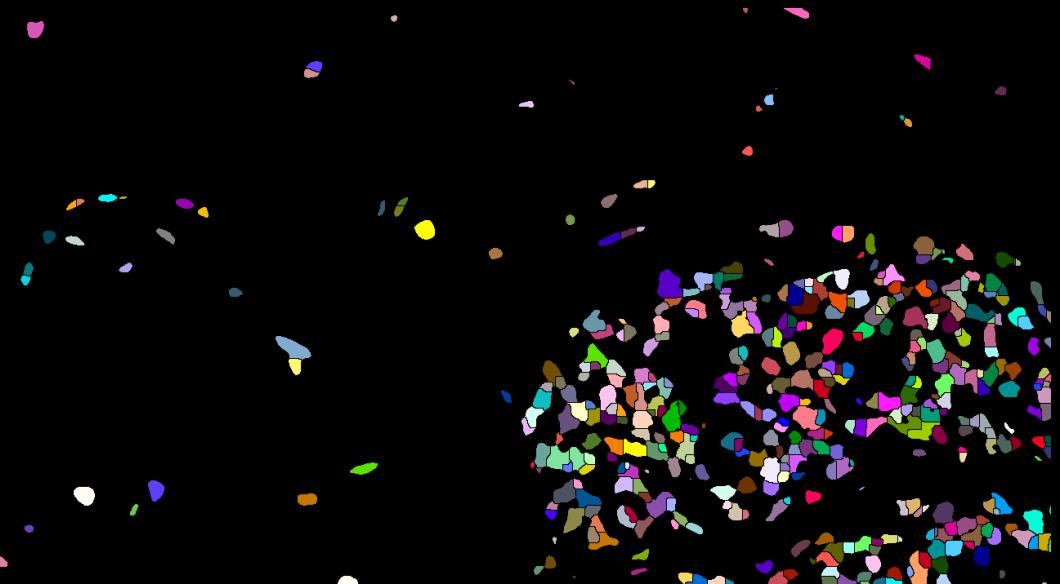
Watershed



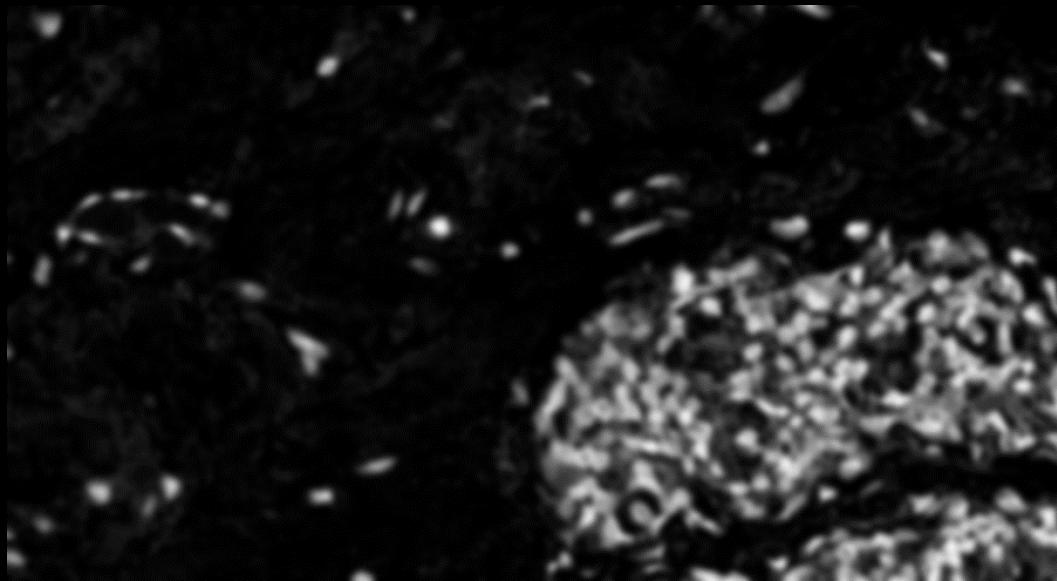
WATERSHED



Watershed



WATERSHED



Watershed



CELL DETECTION TOOL IN QuPATH

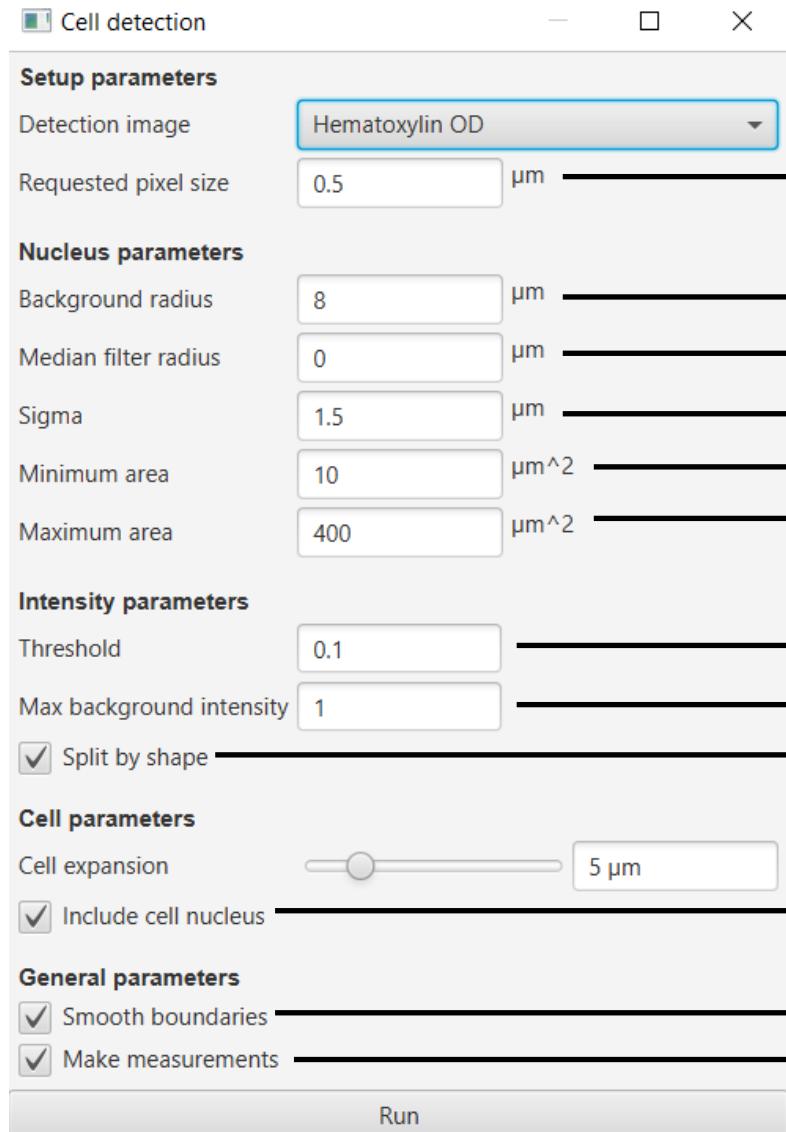


Image **component** used for cell detection

Image **resolution** used for cell detection

Radius used for **minimum filtering**

Radius for **median filtering**

Kernel size used for **Gaussian blur** before watershed

Nuclei with **area inferior** to this value are **filterd out**

Nuclei with **area superior** to this value are **filterd out**

Threshold used for **watershed**

Threshold used for **minimum filtering**

Separate nuclei based on shape (**binary watershed**)

Size used for cell expansion to define **cytoplasm area**

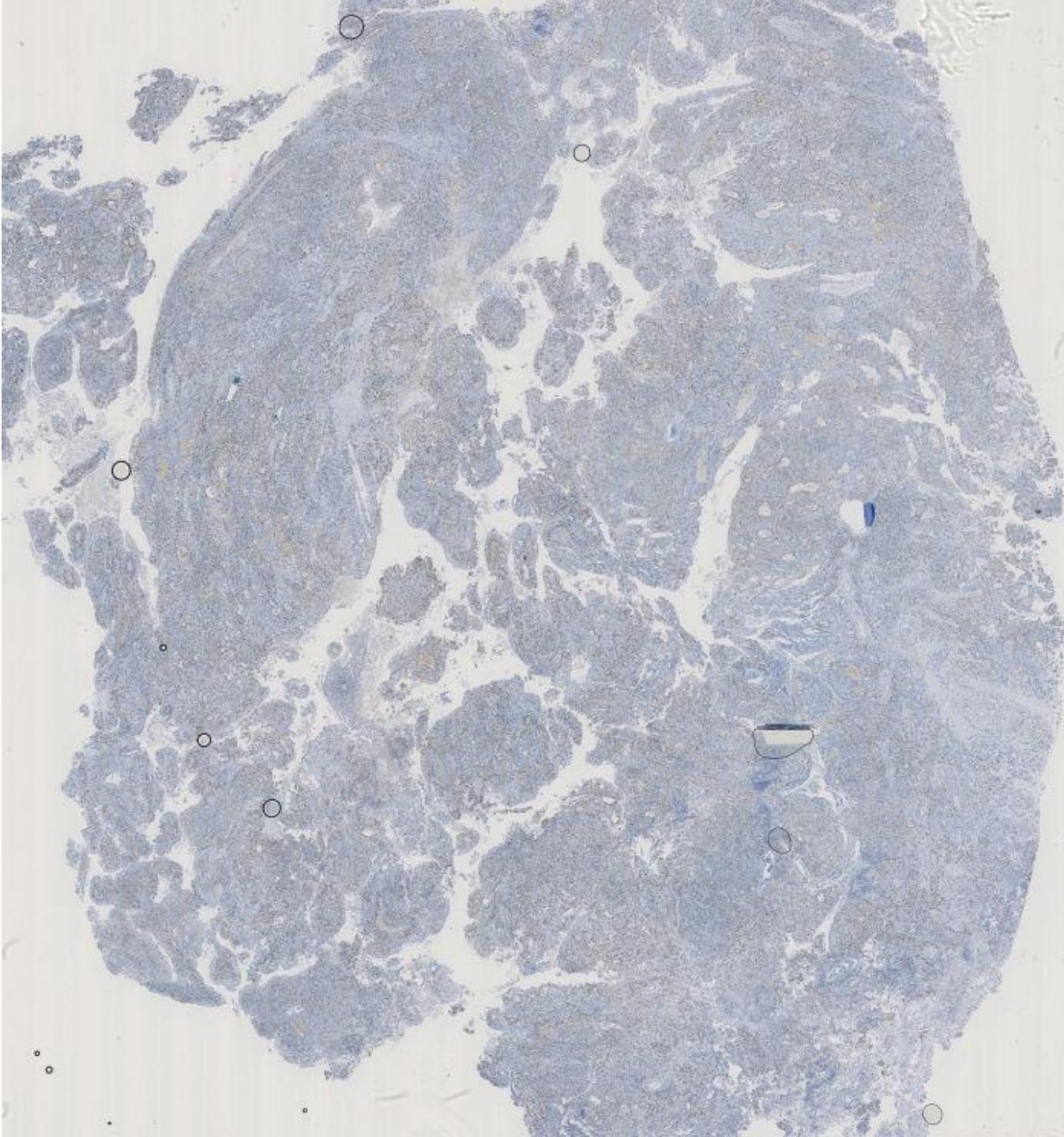
Define **nuclei** and **cytoplasm** areas or only cell areas

Define **smooth cell boundaries**

Get **measurements** associated with nuclei

NUCLEI SEGMENTATION

- Open KI67_lung.ndpi
- Open **Positive cell detection**
- Create an **annotation**
- In the annotation, **detect positive** and **negative cells** by defining **one threshold**
- **Apply "Positive cell detection" on the whole slide**
- **Get number of nuclei and proportion of positive cells**



Cell Detection with Star-convex Polygons

Uwe Schmidt^{1,*}, Martin Weigert^{1,*}, Coleman Broaddus¹, and Gene Myers^{1,2}

¹ Max Planck Institute of Molecular Cell Biology and Genetics, Dresden, Germany
Center for Systems Biology Dresden, Germany
² Faculty of Computer Science, Technical University Dresden, Germany

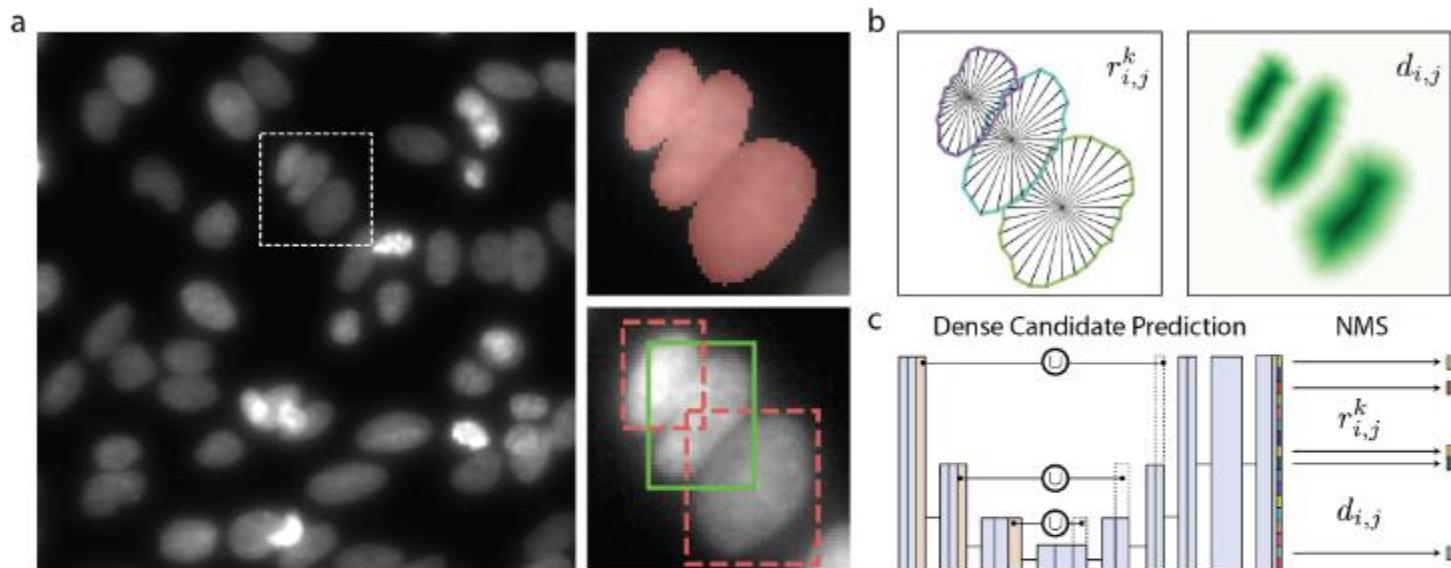


Fig. 1: (a) Potential segmentation errors for images with crowded nuclei: Merging of touching cells (upper right) or suppression of valid cell instances due to large overlap of bounding box localization (lower right). (b) The proposed STARDIST method predicts object probabilities $d_{i,j}$ and star-convex polygons parameterized by the radial distances $r_{i,j}^k$. (c) We densely predict $r_{i,j}^k$ and $d_{i,j}$ using a simple U-Net architecture [15] and then select the final instances via non-maximum suppression (NMS).

Cell Detection with Star-convex Polygons

Uwe Schmidt^{1,*}, Martin Weigert^{1,*}, Coleman Broaddus¹, and Gene Myers^{1,2}

¹ Max Planck Institute of Molecular Cell Biology and Genetics, Dresden, Germany
Center for Systems Biology Dresden, Germany
² Faculty of Computer Science, Technical University Dresden, Germany

For the workshop, a Stardist model was trained with data coming from 3 articles:

- **Whole-cell segmentation of tissue images with human-level performance using large-scale data annotation and deep learning.** *Nature Biotechnology* (2022).
- **A deep learning segmentation strategy that minimizes the amount of manually annotated images.** *F1000 Research* (2022).
- **Deep learning tools and modeling to estimate the temporal expression of cell cycle proteins from 2D still images.** *PLOS Computational Biology* (2022).

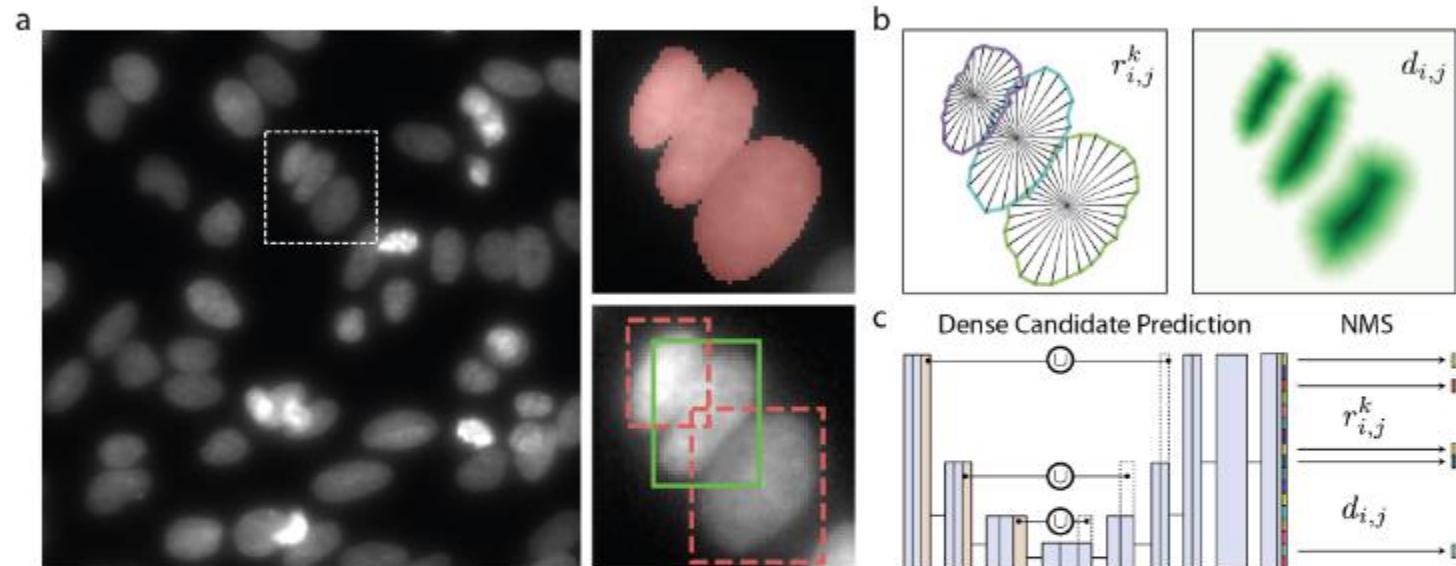


Fig. 1: (a) Potential segmentation errors for images with crowded nuclei: Merging of touching cells (upper right) or suppression of valid cell instances due to large overlap of bounding box localization (lower right). (b) The proposed STARDIST method predicts object probabilities $d_{i,j}$ and star-convex polygons parameterized by the radial distances $r_{i,j}^k$. (c) We densely predict $r_{i,j}^k$ and $d_{i,j}$ using a simple U-Net architecture [15] and then select the final instances via non-maximum suppression (NMS).

SEGMENTATION WITH STARDIST

**Download the latest
Stardist extension for
QuPath and drag it into
QuPath**

The screenshot shows a GitHub repository page for 'qupath/qupath-extension-stardist'. The repository is public and has 3 watches and 5 forks. The 'Code' tab is selected. Below the tabs are 'Issues' (2), 'Pull requests', 'Actions', 'Security', and 'Insights'. A search bar at the top right says 'Find a release'. The main content area shows the 'v0.3.0' release, which was published on Sep 02, 2021, by petebankhead. It includes a download link for 'qupath-extension-stardist-0.3.0.jar' (24.7 KB) and source code in zip and tar.gz formats. There are also three other assets listed under 'Assets'.

Search or jump to... / Pull requests Issues Marketplace Explore

qupath / qupath-extension-stardist Public Watch 3 Fork 5

Code Issues 2 Pull requests Actions Security Insights

Releases Tags Find a release

Sep 02, 2021 petebankhead v0.3.0 dd1f354 Compare

v0.3.0 Latest

This version of the *QuPath Stardist extension* is designed to work with QuPath v0.3.0 (and possibly later versions).

Assets 3

- qupath-extension-stardist-0.3.0.jar 24.7 KB
- Source code (zip)
- Source code (tar.gz)

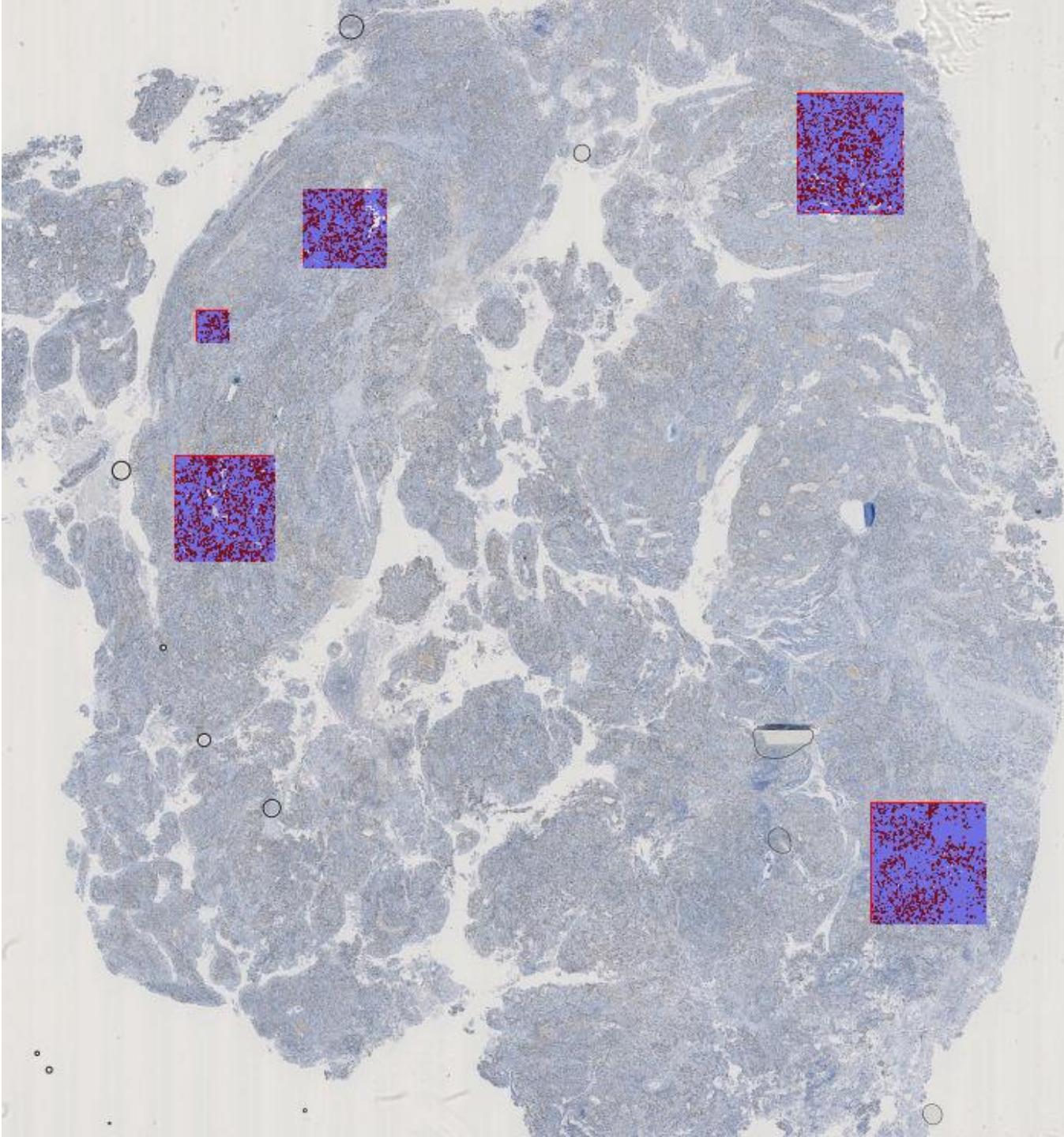
SEGMENTATION WITH STARDIST

Open **Scrip editor**, then open
nucleus_detection_hematoxylin_da
b.groovy

```
1 import qupath.ext.stardist.StarDist2D
2 import qupath.lib.images.servers.ColorTransforms
3 import qupath.imagej.gui.ImageJMacroRunner
4
5 min_nuclei_area = 15
6
7 // Specify the model directory (you will need to change this!)
8 def pathModel = "C:/Work/QuPath/scripts/StardistModels/TissueNet_all.pb"
9
10 def stardist_segmentation = StarDist2D.builder(pathModel)
11     .threshold(0.5)                      // Prediction threshold
12     .normalizePercentiles(1, 99.8)        // Percentile normalization
13     .pixelSize(0.5)                     // Resolution for detection
14     .channels(
15         ColorTransforms.createColorDeconvolvedChannel(getCurrentImageData().getColorDeconvolutionStains(), 1),
16         ColorTransforms.createColorDeconvolvedChannel(getCurrentImageData().getColorDeconvolutionStains(), 2)
17     )
18     .cellExpansion(5.0)                  // Approximate cells based upon nucleus expansion
19     .cellConstrainScale(1.5)            // Constrain cell expansion using nucleus size
20     .measureShape()                   // Add shape measurements
21     .measureIntensity()              // Add cell measurements (in all compartments)
22     .build()
23
24
25 def imageData = getCurrentImageData()
26 def hierarchy = imageData.getHierarchy()
27 def annotations = hierarchy.getAnnotationObjects()
28
29 // Run detection for the selected objects
30 stardist_segmentation.detectObjects(imageData, annotations)
31
32 //def toDelete = getDetectionObjects().findAll {measurement(it, 'Circularity') < 0.9}
33 def toDelete = getDetectionObjects().findAll {measurement(it, 'Area µm^2') < min_nuclei_area}
34 removeObjects(toDelete, true)
35
36
37 println 'Done!'
```

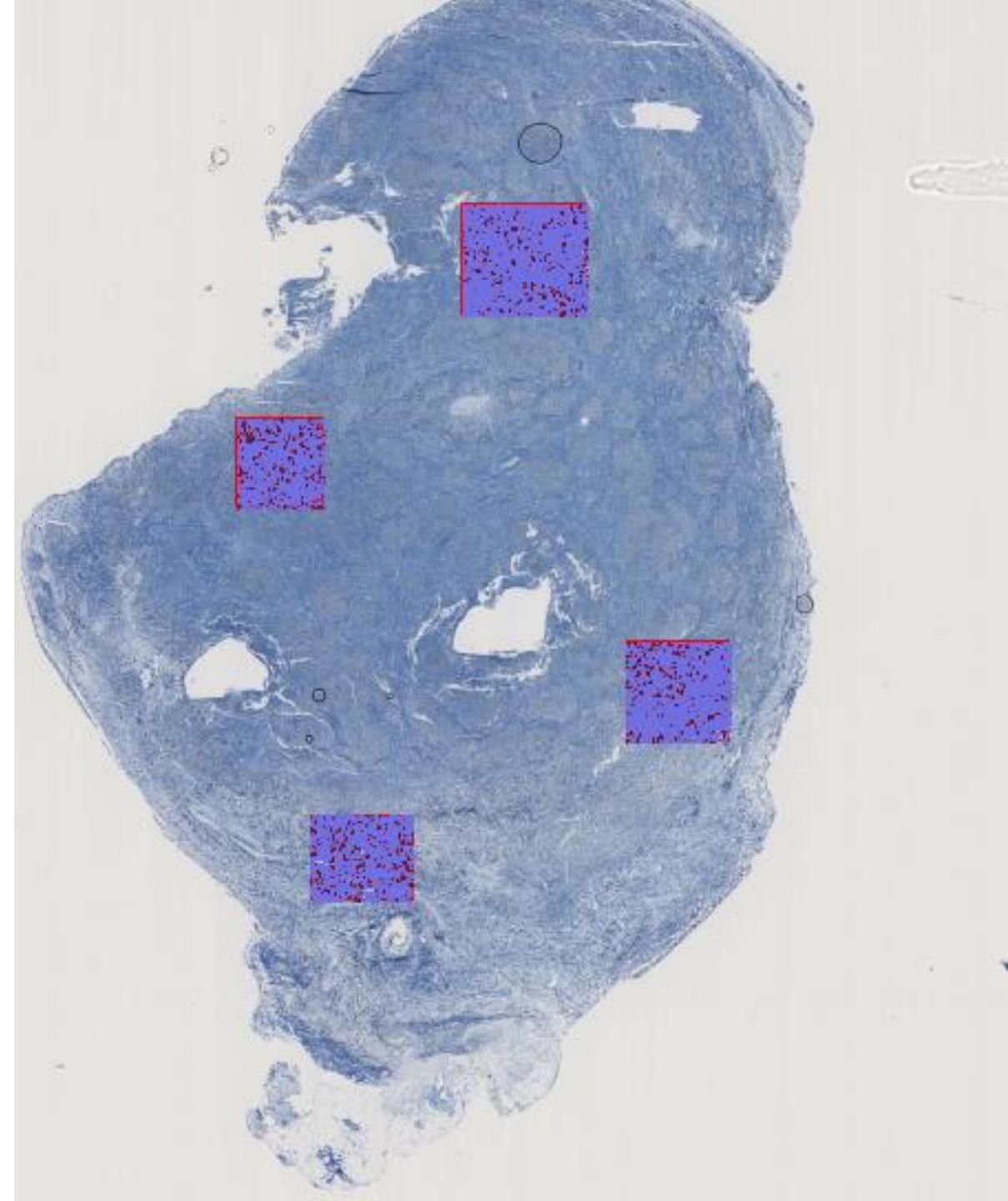
SEGMENTATION WITH STARDIST

- Open KI67_lung.ndpi
- Define a **small rectangle annotation** to test the **Stardist parameters**
- Open **Create single measurement classifier** and define threshold to identify DAB+ cells
- Run **stardist** on a small number of annotations and identify DAB+ cells
- Get the **proportion of DAB+ cells**
- **Compare** with the results obtained with the **watershed-based approach**

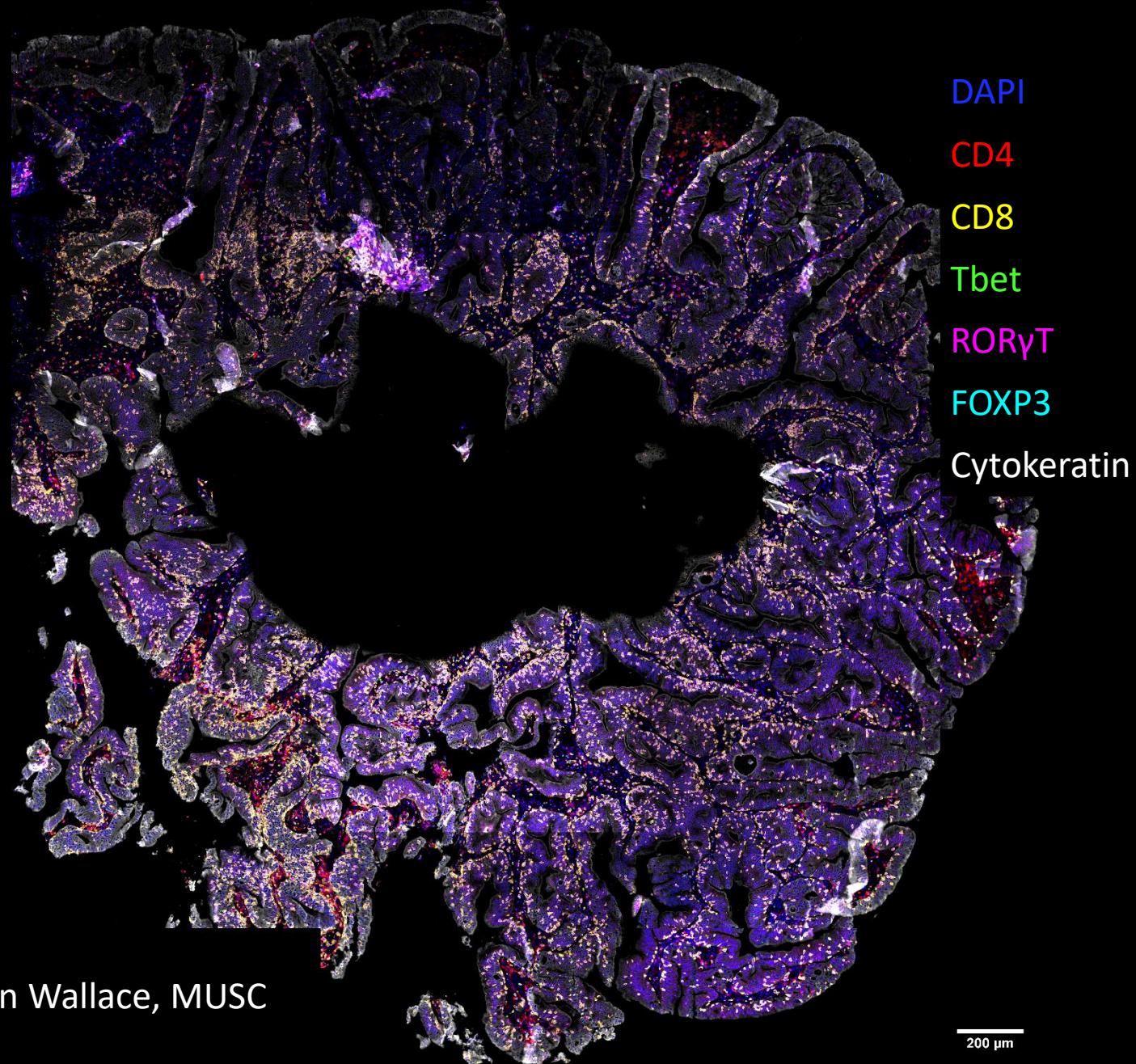


SEGMENTATION WITH STARDIST

- Open KI67_lymphoma.ndpi
- Define **3 or 4 ROIs**
- Modify script to do **both segmentation and thresholding** (workflow tab)
- Is it a **good way to quantify** this image ?



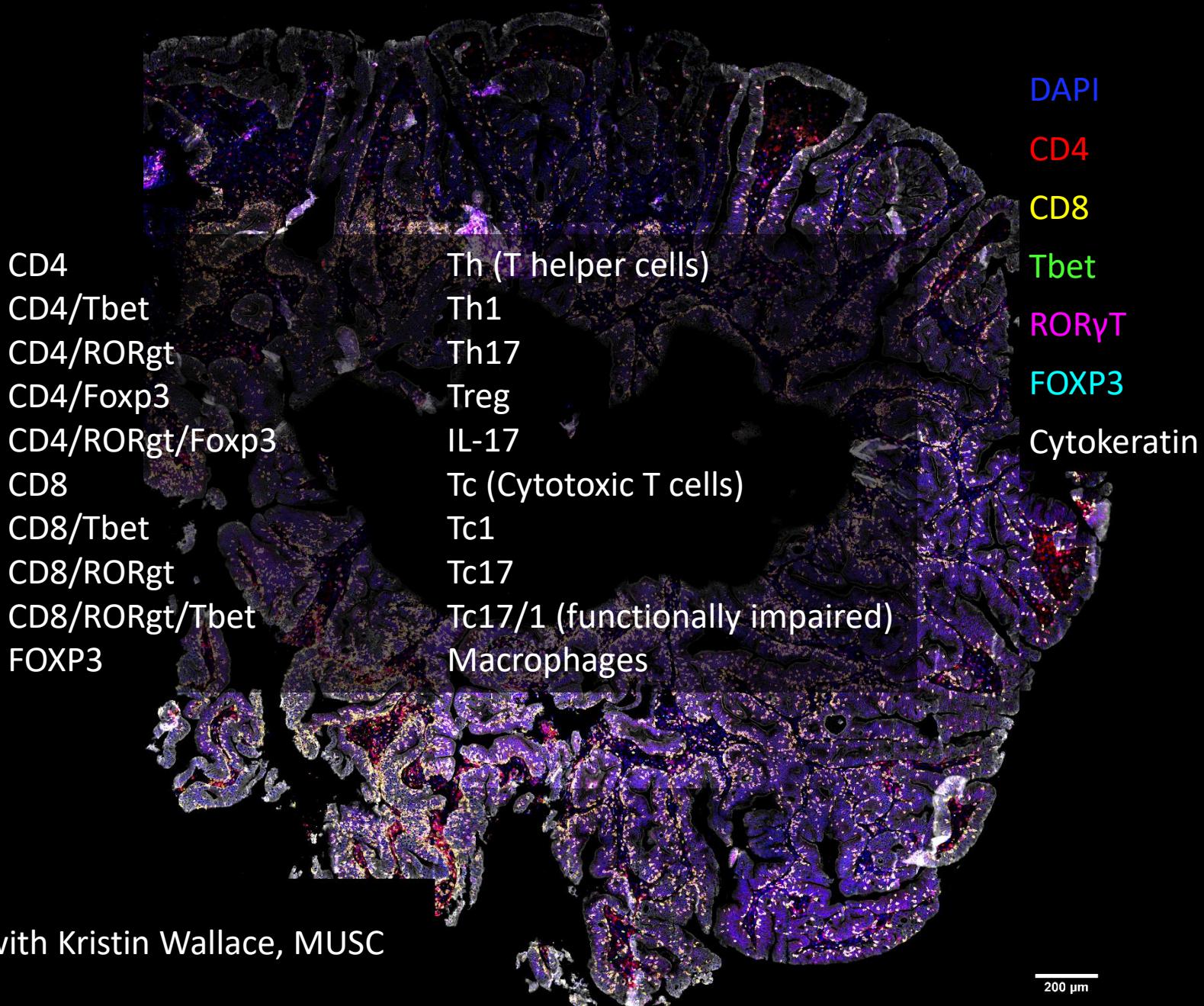
MULTI/HYPER-PLEXED IMAGES



Polyp study with Kristin Wallace, MUSC

200 μ m

MULTI/HYPER-PLEXED IMAGES



SHALLOW MACHINE LEARNING FOR OBJECT CLASSIFICATION

As for pixel classification:

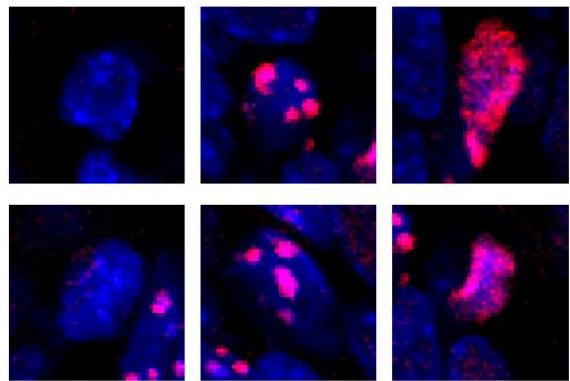
- **Examples** of classes are **manually** defined by the user
- A **classifier** is **trained** with these examples
- Data is then **automatically classified** by using the trained classifier

But this time, features are **measurements associated to detections** (most often cells or nuclei) such as:

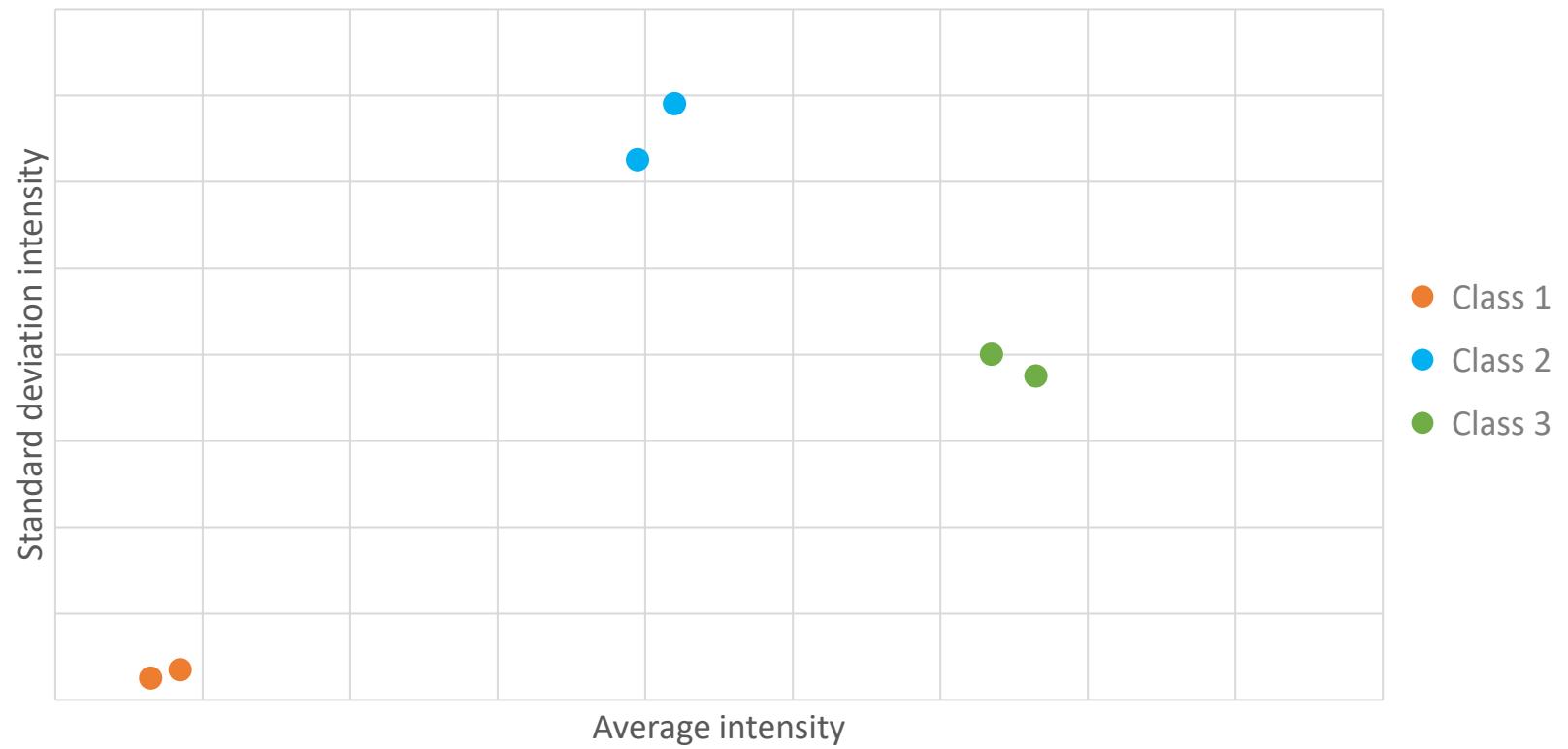
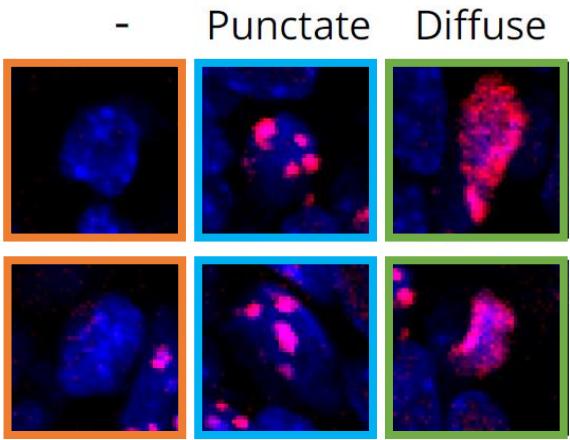
- **Average** intensity
- **Median** intensity
- **Standard deviation** of intensity
- **Minimum/Maximum** intensity
- **Object area**
- **Object Circularity**
- ...

SHALLOW MACHINE LEARNING FOR OBJECT CLASSIFICATION

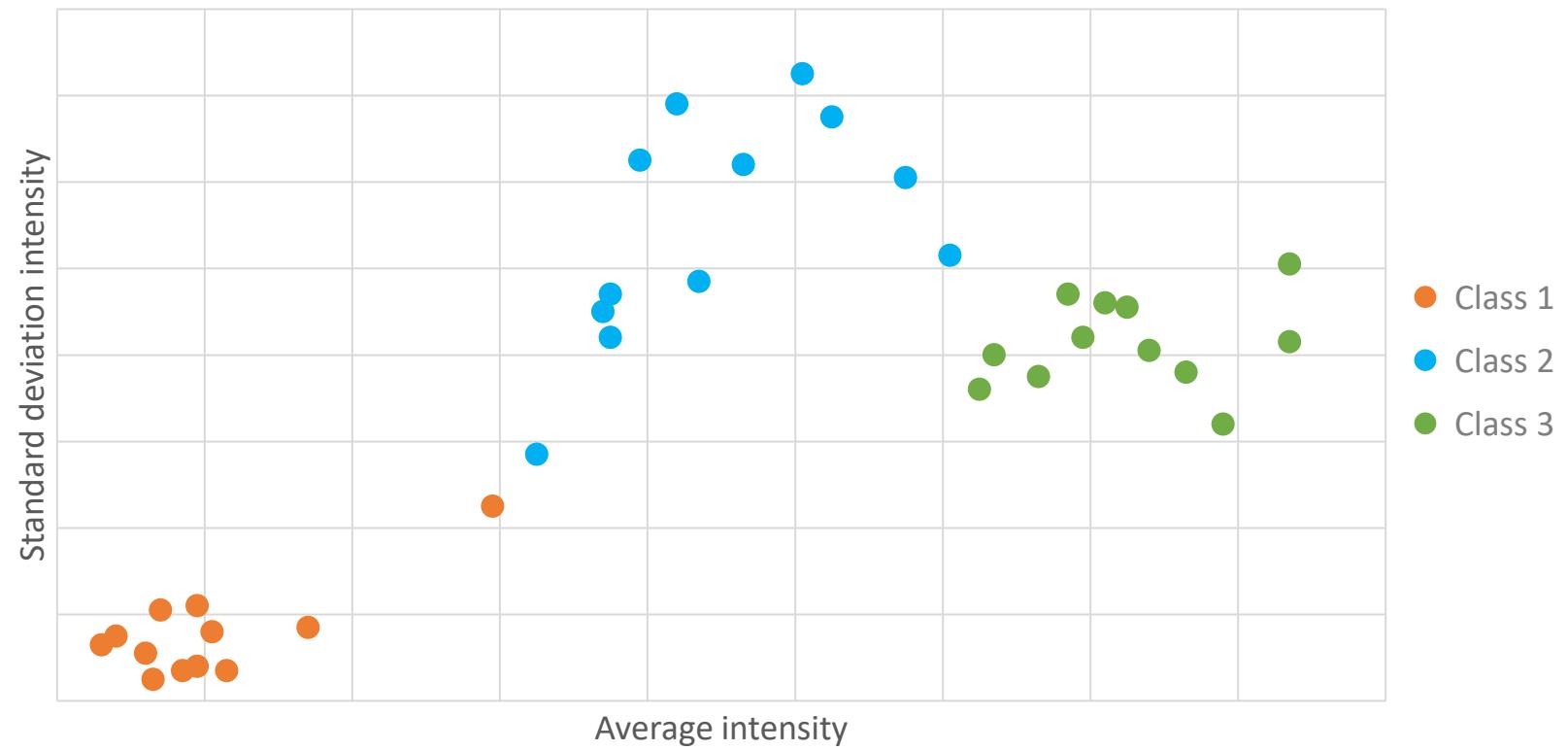
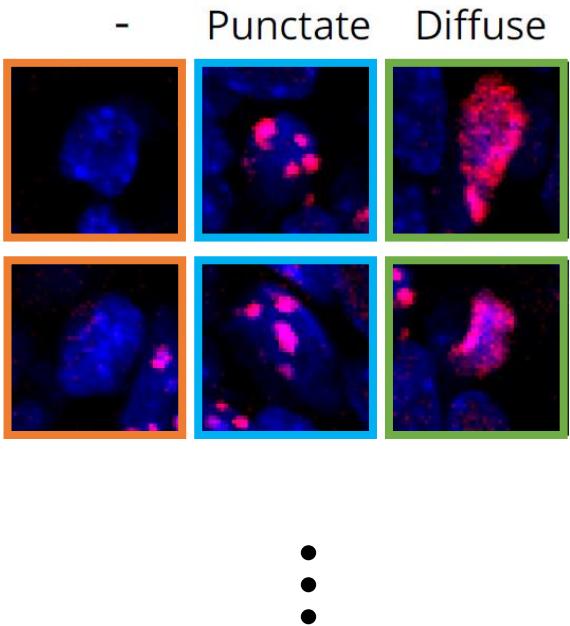
- Punctate Diffuse



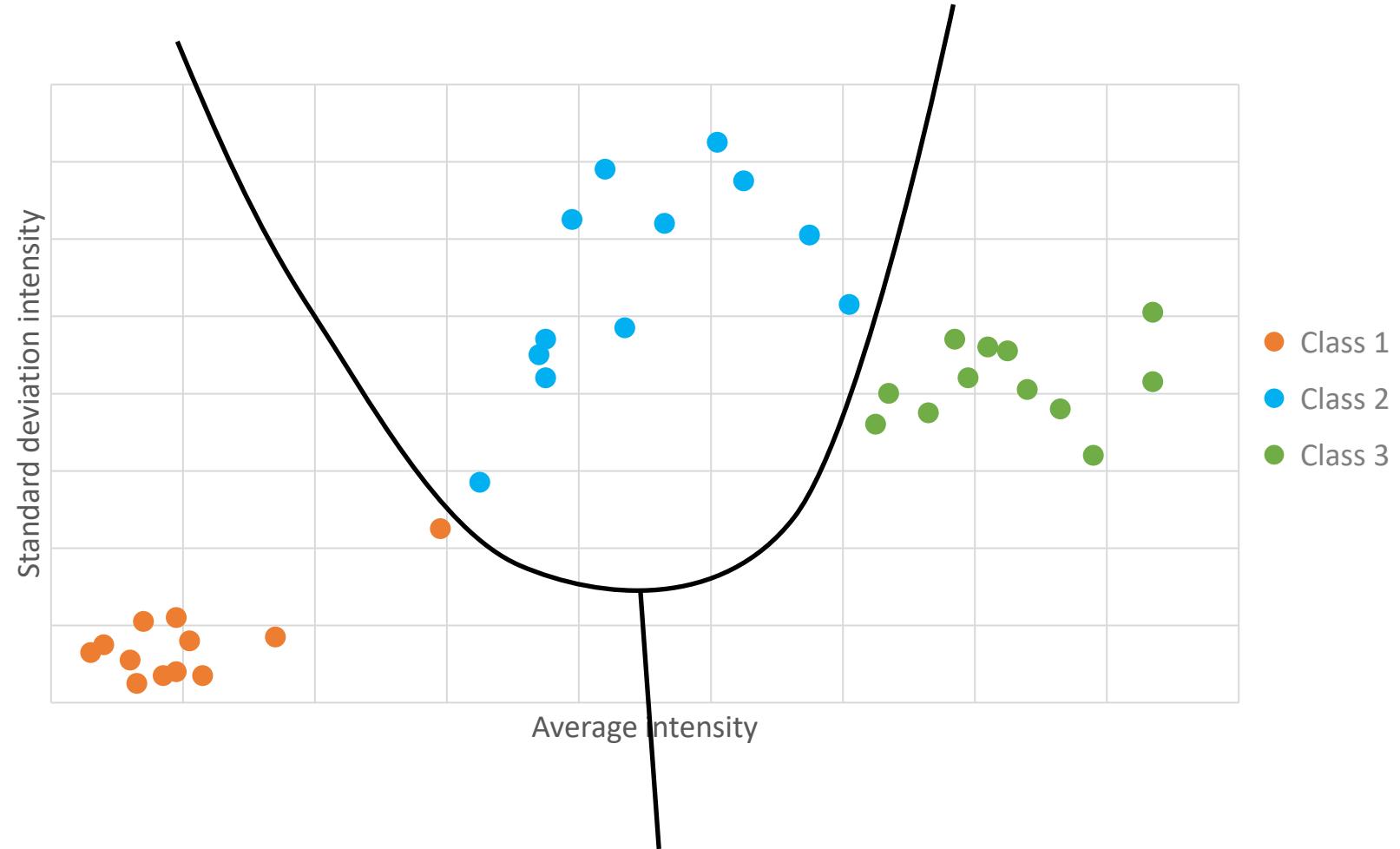
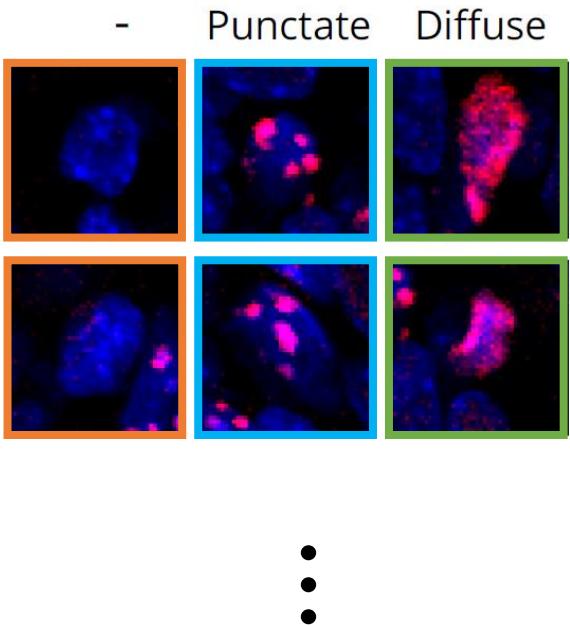
SHALLOW MACHINE LEARNING FOR OBJECT CLASSIFICATION



SHALLOW MACHINE LEARNING FOR OBJECT CLASSIFICATION

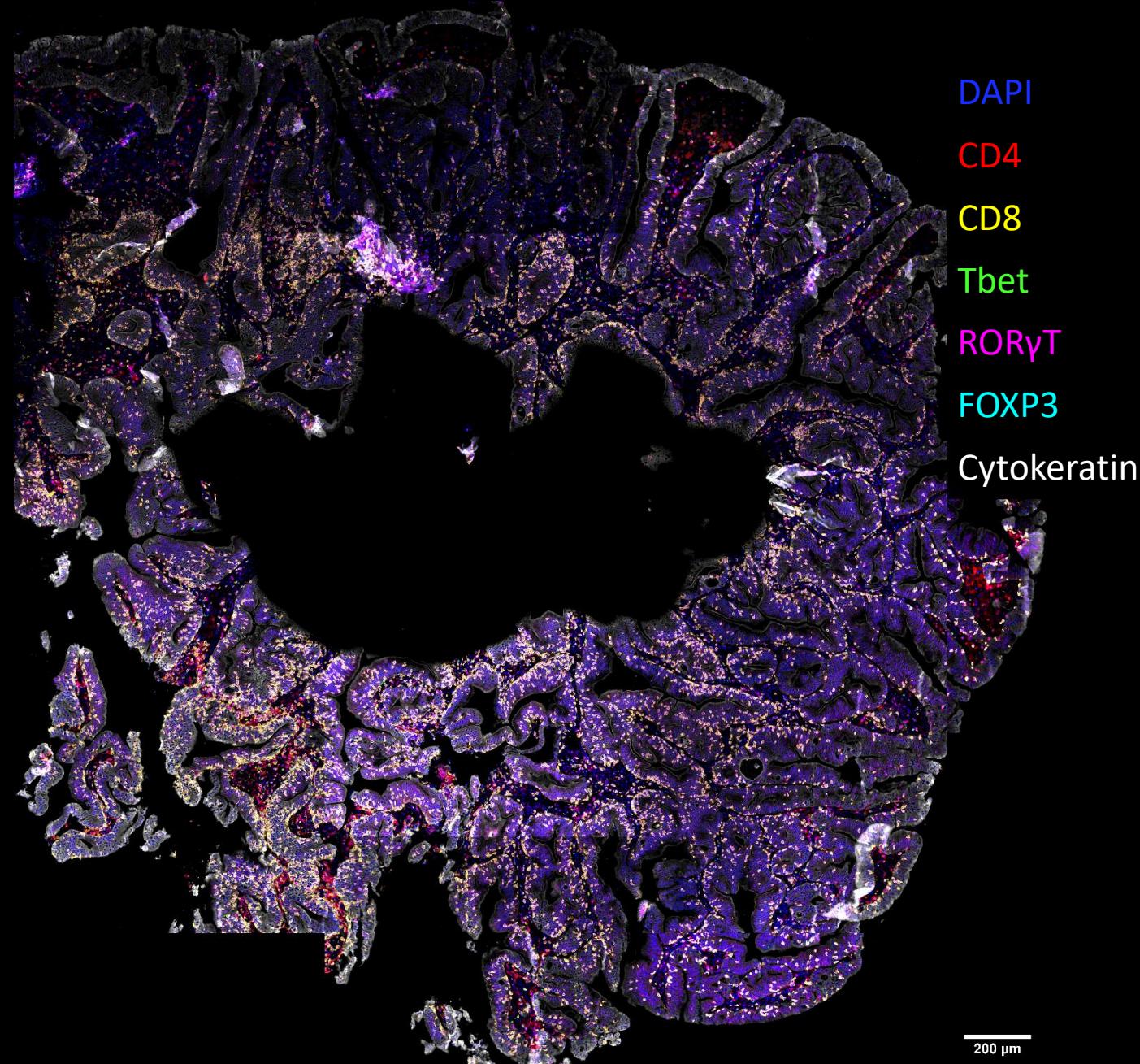


SHALLOW MACHINE LEARNING FOR OBJECT CLASSIFICATION



MULTI/HYPER-PLEXED IMAGES

- Use a pixel classifier to segment epithelium and stroma, save it and run it
- Define a small annotation and run stardist to segment nuclei
- Load geojson file to get all nuclei segmented
- Train an object classifier to identify positive cells for each marker
- Export the results as one csv file for each cell type



MULTI/HYPER-PLEXED IMAGES

- Install Cytomap
- Load data as separate csv files
- Define neighborhoods
- Cluster neighborhoods into regions
- Visualize the proportions of cell types per region

The screenshot shows a GitLab repository page for the project 'CytoMAP'. The sidebar on the left lists project navigation options: CytoMAP (selected), Project information, Repository, Files (highlighted), Commits, Branches, Tags, Contributors, Graph, Compare, Locked Files, and Issues (with 19 notifications). The main content area displays the commit history for the 'master' branch under the 'Repository' tab. The commits are listed in reverse chronological order. The first commit is a recent one by Caleb Stoltzfus, dated 8 months ago, titled 're-compiled correct version 1.4.21'. Below this, there are three older commits related to installer files:

Name	Last commit
..	
CytoMAP_Installer_WindowsV1.4.1.exe	Updated Standalone installer
CytoMAP_Installer_WindowsV1.4.10.exe	Re-Compiled CytoMAP version 1.4.10
CytoMAP_Installer_WindowsV1.4.11.exe	Compiled version 1.4.11

CITATIONS

- P. Bankhead *et al.* **QuPath: Open source software for digital pathology image analysis.** *Scientific Reports* (2017). <https://doi.org/10.1038/s41598-017-17204-5>
- U. Schmidt *et al.* **Cell Detection with Star-convex Polygons.** *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)* (2018). <https://arxiv.org/abs/1806.03535>
- N.F. Greenwald *et al.* **Whole-cell segmentation of tissue images with human-level performance using large-scale data annotation and deep learning.** *Nature Biotechnology* (2021). <https://doi.org/10.1038/s41587-021-01094-0>
- T. Péicot *et al.* **A deep learning segmentation strategy that minimizes the amount of manually annotated images.** *F1000 Research* (2022) <https://doi.org/10.12688/f1000research.52026.2>
- T. Péicot *et al.* **Deep learning tools and modeling to estimate the temporal expression of cell cycle proteins from 2D still images.** *PLOS Computational Biology* (2022)

VIDEO TUTORIALS

- [QuPath installation, data and script downloading](#)
- [Project creation and annotations](#)
- [Stain deconvolution](#)
- [Pixel classification \(epithelium/stroma for H&E images\)](#)
- [Nuclei segmentation \(watershed\) and DAB positive cells](#)
- [Nuclei segmentation \(stardist\) and DAB positive cells \(thresholding\)](#)
- [Visualization of fluorescence images](#)
- [Pixel classification \(epithelium/stroma for fluorescence images\)](#)
- [Nuclei segmentation \(stardist\)](#)
- [Object classification for marker identification](#)