E4 Project : MALIS

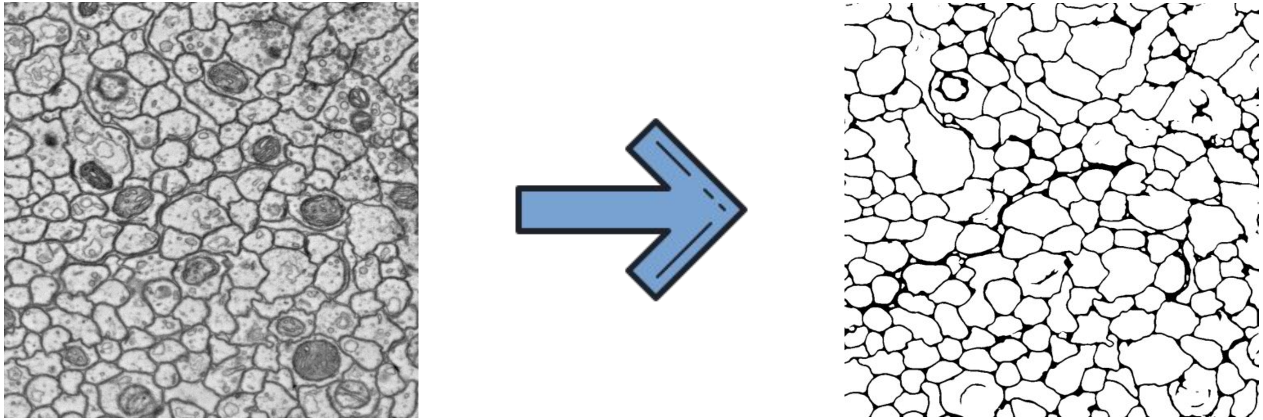
Work available at : *github.com/garridoq/malis-project*

Quentin GARRIDO
Tiphanie LAMY-VERDIN
Raphaël LAPERTOT
Josselin LEFEVRE
Annie LIM

Supervisor : Laurent NAJMAN

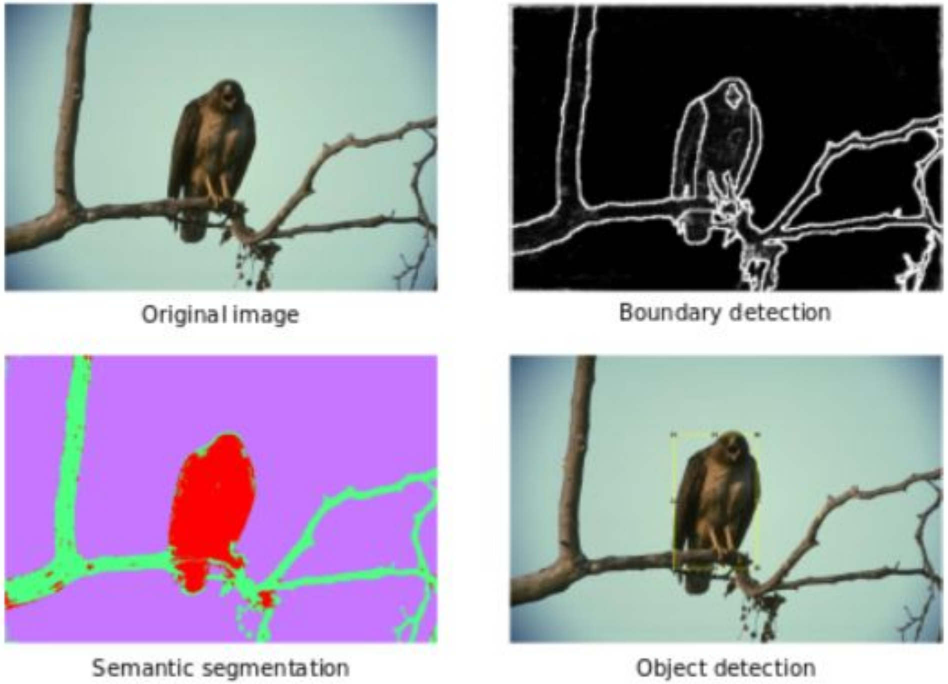
Introduction

Drosophila connectome segmentation



⇒ **Brain architecture**

Image segmentation



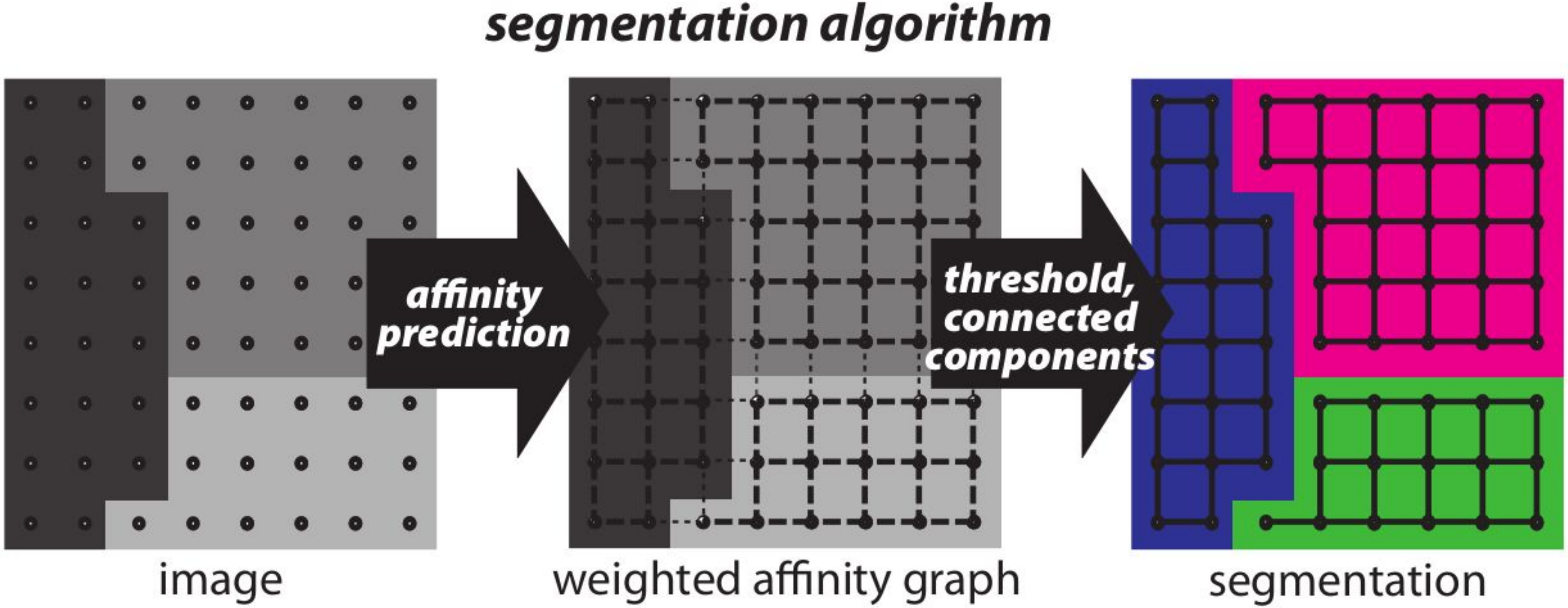
Source : ImageJ

Summary

- Method’s presentation
- Our implementation
- Results
- Project organization

Method’s presentation

S. Turaga *et al.*, “Maximin affinity learning of image segmentation,” 2009



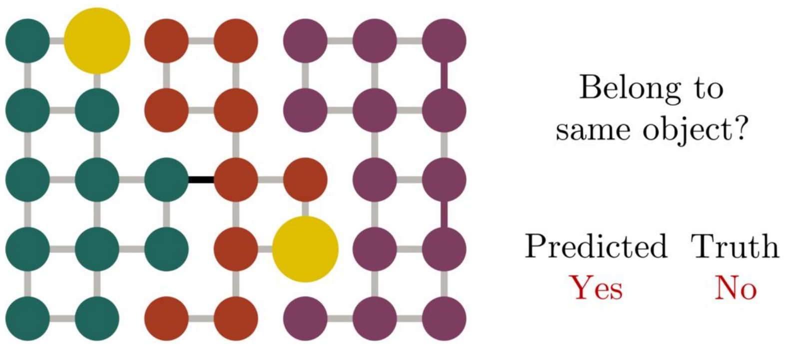
How to predict affinity ?

Various methods :

- Distance between pixels
- Any kind of classifier that is smooth
- Neural Networks

Rand Index

$$1 - RI(\hat{S}, S) = \binom{N}{2}^{-1} \sum_{i<j} \left| \delta(s_i, s_j) - \delta(\hat{s}_i, \hat{s}_j) \right|$$



Computing the Maximin affinity

- Maximin path : $$P^*_{ij} = \arg\max_{P \in \mathcal{P}_{ij}} \min_{\langle k,l \rangle \in P} A_{kl}$$

- Maximin affinity : $$A^*_{ij} = \max_{P \in \mathcal{P}_{ij}} \min_{\langle k,l \rangle \in P} A_{kl}$$

Property : Any path in a MST is a Maximin path

Computing the Maximin affinity

First paper approach :

Process:

- Pick a single pair of random pixels i,j
- Find the maximin affinity between them

Data from the MST is not used fully

S. Turaga, *et al.* , "Maximin affinity learning of image segmentation," 2009

Computing the Maximin affinity

Second paper approach : Constrained MALIS

We want to compute a loss over all pairs of points, which gives us :

$$L(s,a) = \sum_{u,v \in F} l(\delta(u,v), a(\mathrm{mm}(u,v))).$$

This can be reduced to a quasilinear computation :

$$w_P(e) = |\{(u,v) \in F^2 \mid \delta(u,v) = 1,\ e = \mathrm{mm}(u,v)\}|.$$

$$w_N(e) = |\{(u,v) \in F^2 \mid \delta(u,v) = 0,\ e = \mathrm{mm}(u,v)\}|$$

$$L(s,a) = \sum_{e \in \mathrm{MST}(G)} w_P(e) l(1, a(e)) + w_N(e) l(0, a(e))$$

Constrained Malis loss

Computing wp and wn :

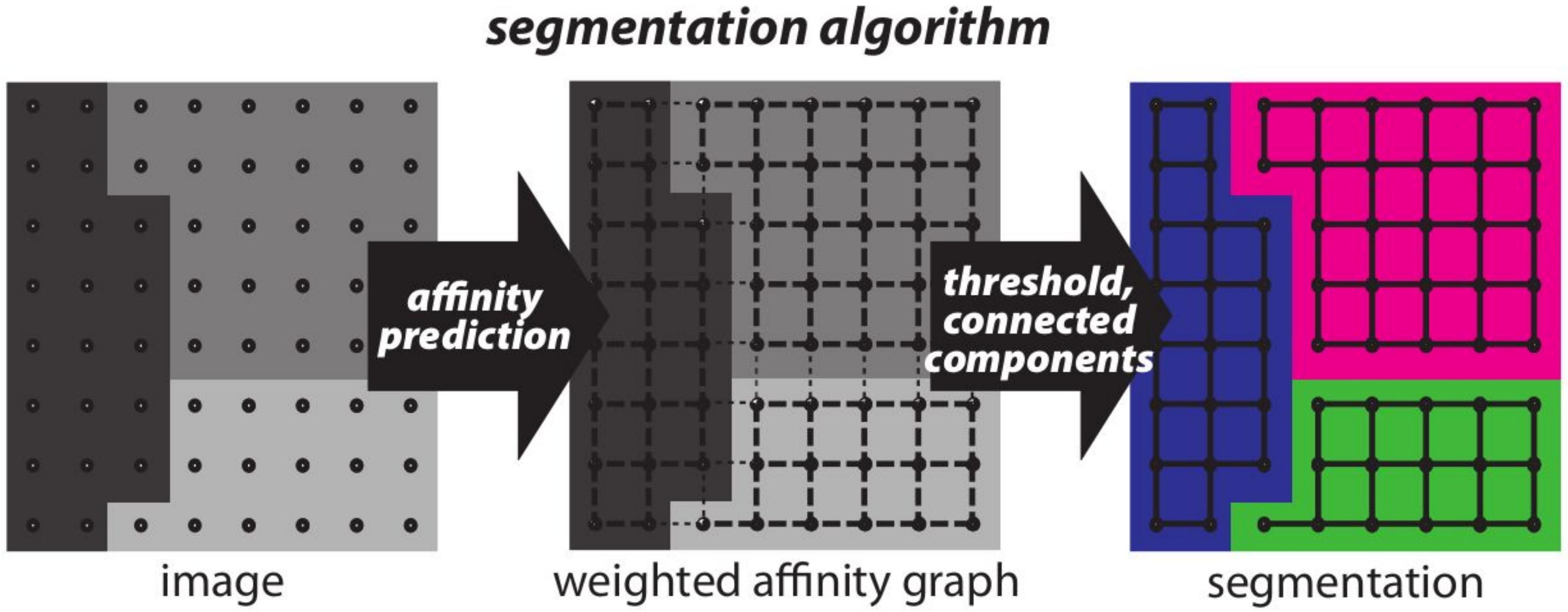
Adding an edge e to our MST merges two trees T_1 and T_2

Then $$w_p = \sum_{i \in \{1, \ldots k\}} |V^i_{T_1}||V^i_{T_2}|$$

$$w_n = \sum_{i \neq j \in \{1, \ldots k\}} |V^i_{T_1}||V^j_{T_2}| = |V_{T_1}||V_{T_2}| - w_p$$

J. Funke *et al.*, "Large Scale Image Segmentation with Structured Loss Based Deep Learning for Connectome", July 2019.

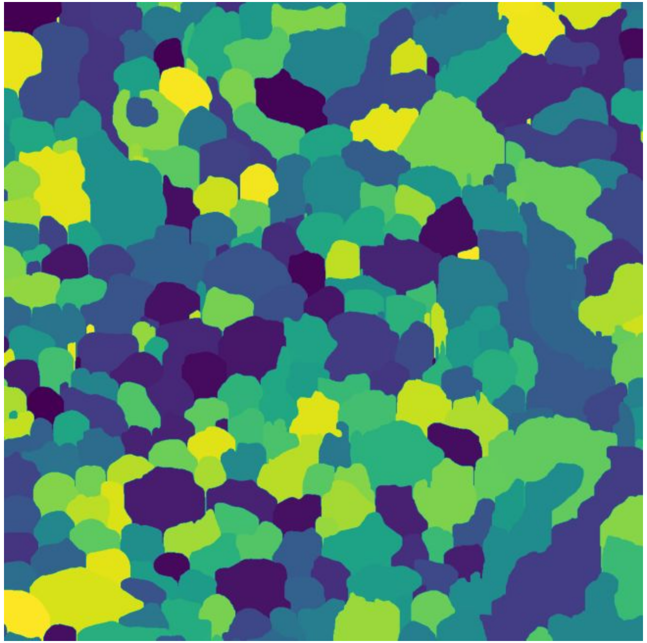
MALIS' method



Additional post processing

Examples

- Watershed
- Watershed + deletion of small regions
- Mumford Shah + cut
- **Watershed + agglomeration (as used in the improved version of MALIS)**



Result of a watershed

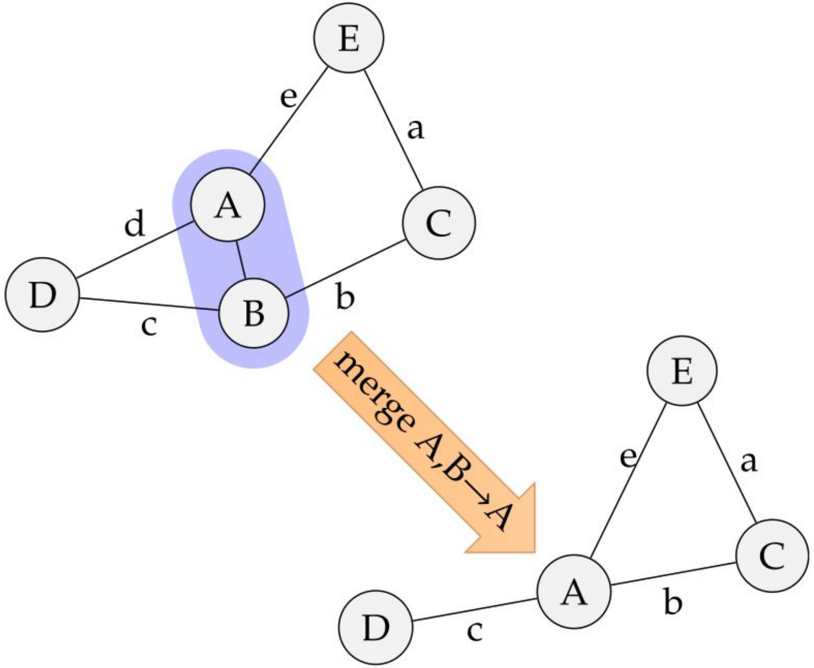
J. Cousty *et al.*, “Watershed cuts: minimum spanning forests, and the drop of water principle.”, 2009.

Agglomeration of objects

- Compute a Region Adjacency Graph

- Edge values are the affinity between both objects

- Merge object with smallest edge value between them

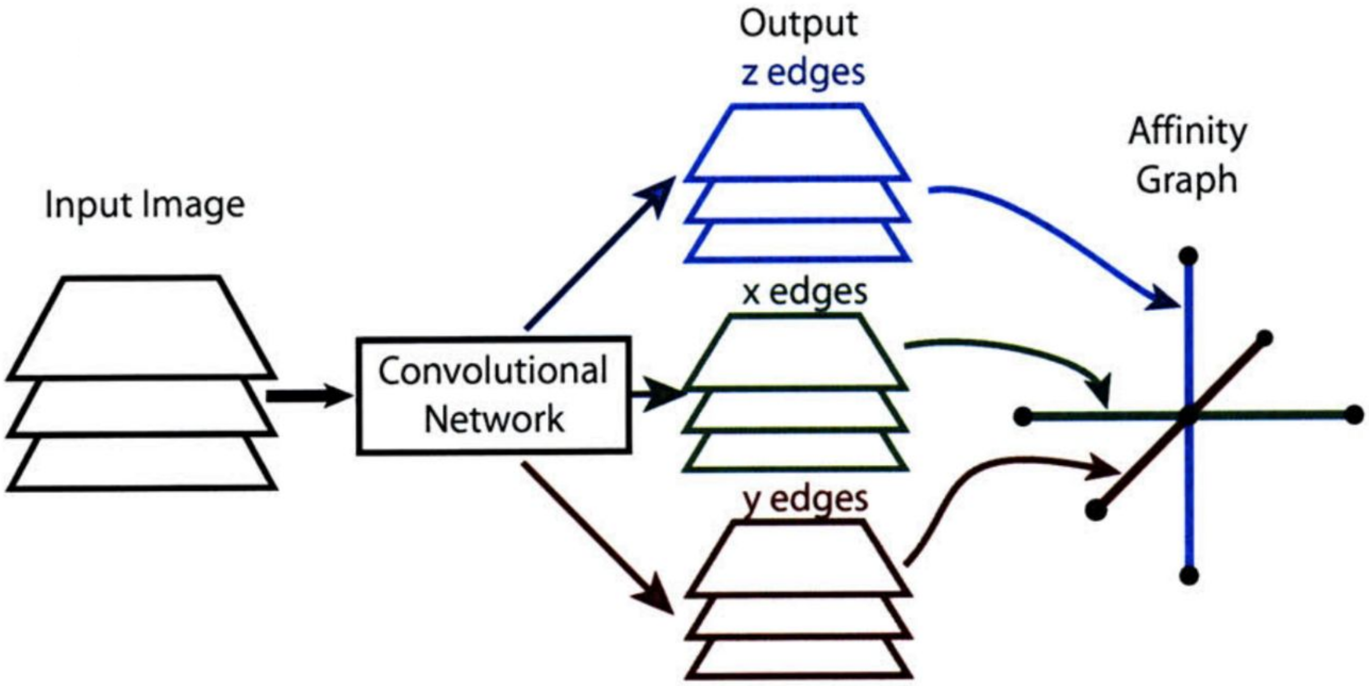


J. Funke *et al.*, “Large Scale Image Segmentation with Structured Loss Based Deep Learning for Connectome”, July 2019.

Our implementation

- Pytorch for the neural network

- Higra to compute the loss



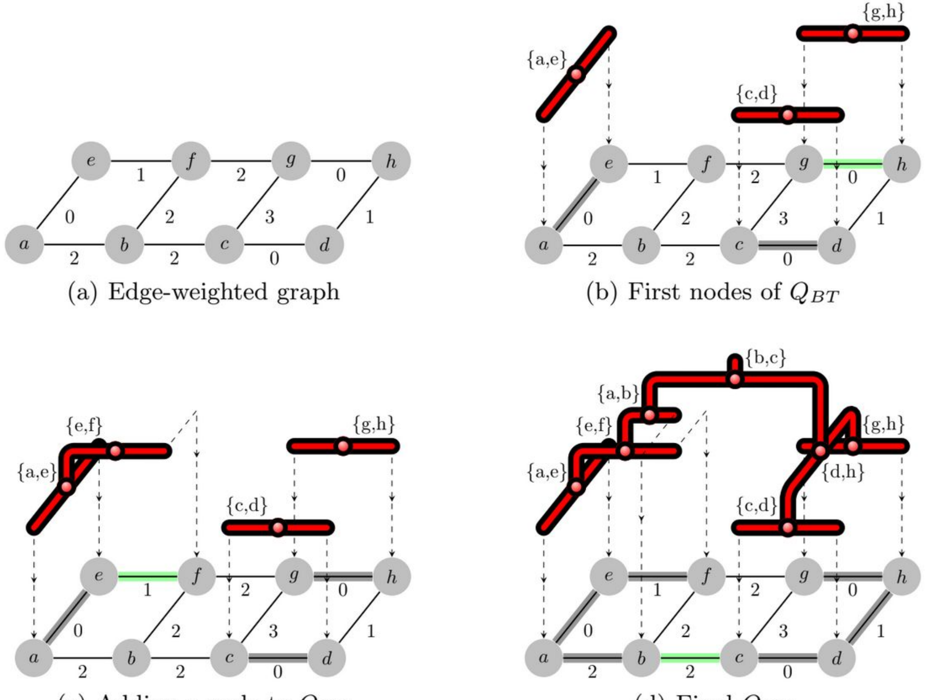
S. Turaga, “Learning image segmentation and hierarchies by learning ultrametric distances”, Apr. 2010.

Using a MST and a BPT

The BPT contains all the data for computing

$$w_p = \sum_{i \in \{1,\ldots k\}} |V_{T_1}^i||V_{T_2}^i|$$

and w_n



(a) Edge-weighted graph

(b) First nodes of Q_{BT}

(c) Adding a node to Q_{BT}

(d) Final Q_{BT}

S. Turaga, "Learning image segmentation and hierarchies by learning ultrametric distances", Apr. 2010.

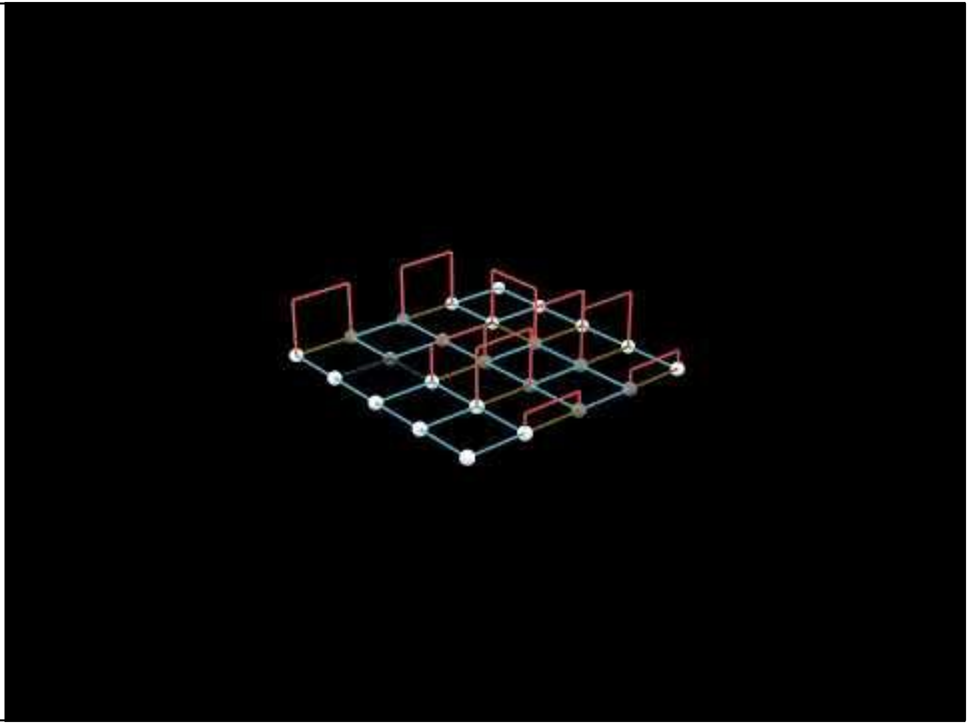
Visualizing the construction of the MST & BPT

Light blue : Low affinity

Dark blue : High affinity

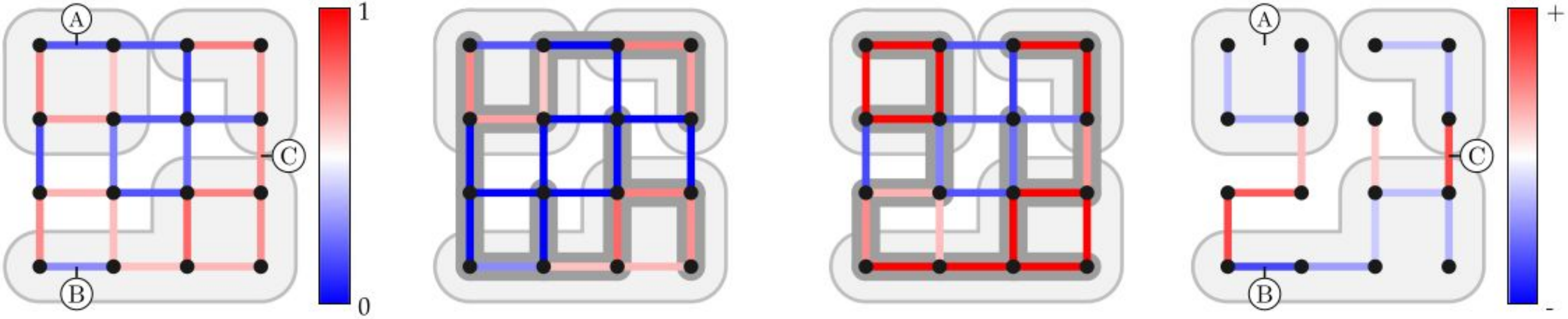
Yellow : Maximum Spanning Tree

Red : Binary Partition Tree



Two pass loss

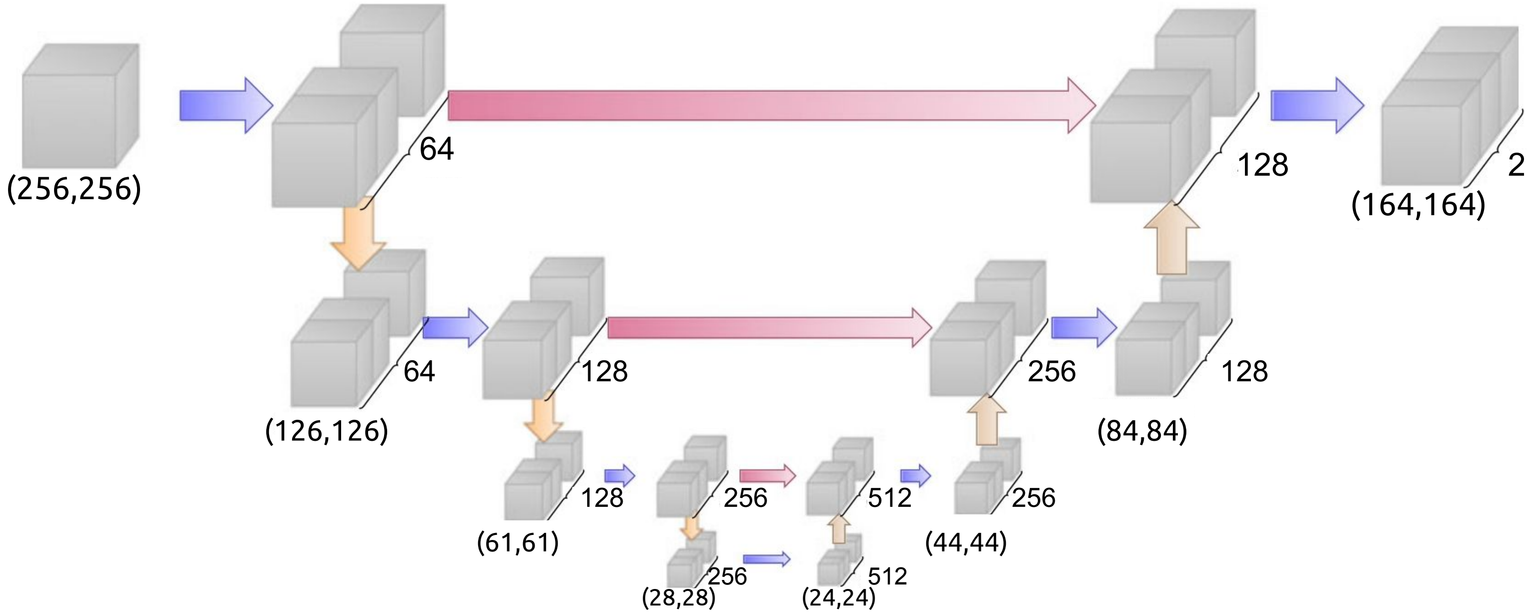
$$L(s, a) = \sum_{e \in \mathrm{MST}(G)} w_P(e) l(1, a(e)) + w_N(e) l(0, a(e))$$



(a) Predicted affinities. (b) Positive pass. (c) Negative pass. (d) Gradient of loss.

J. Funke *et al.*, "Large Scale Image Segmentation with Structured Loss Based Deep Learning for Connectome", July 2019.

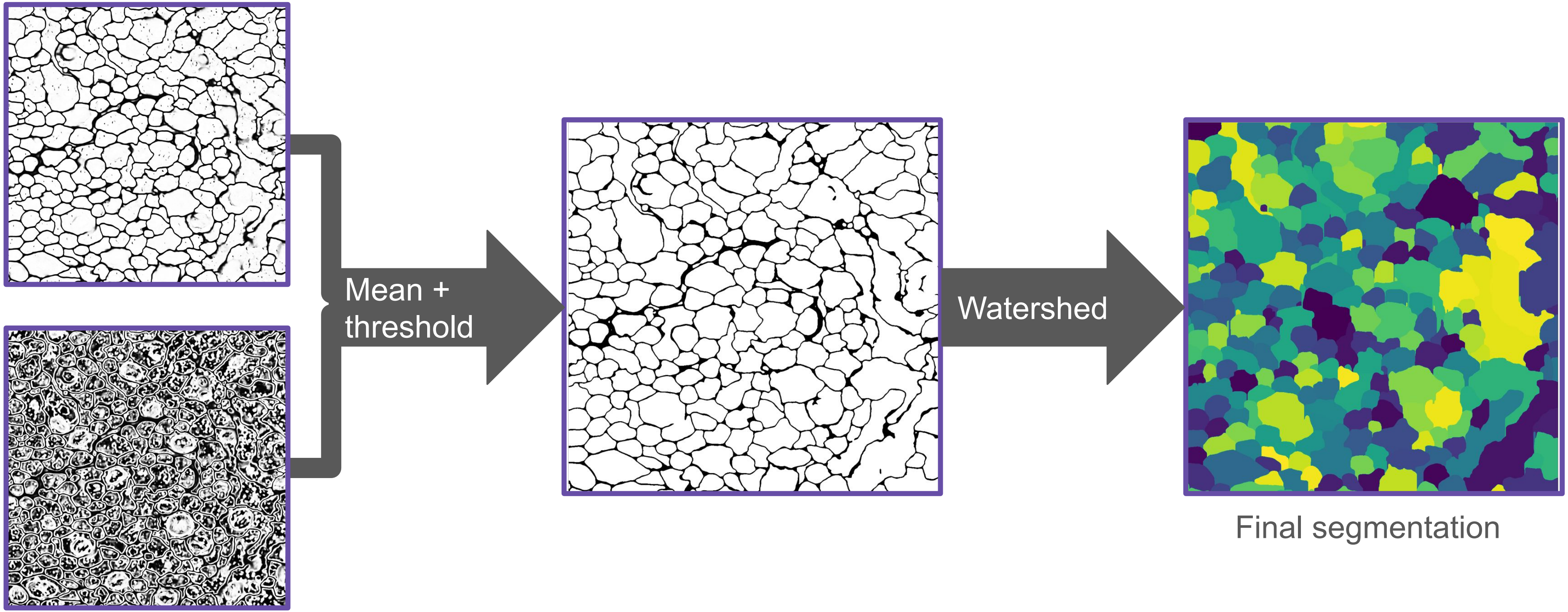
Results: U-net architecture



Architecture used for the experiments

J. Funke *et al.*, “Large Scale Image Segmentation with Structured Loss Based Deep Learning for Connectome”, July 2019.

Results: Post processing used

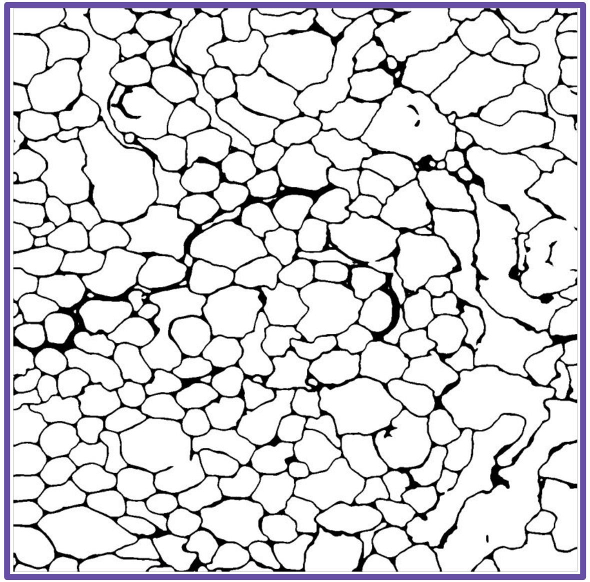


Affinity predictions

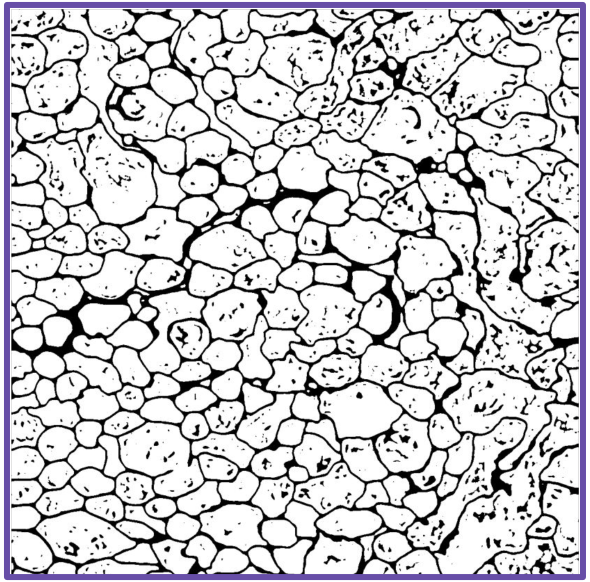
Final segmentation

Results: Post processing used

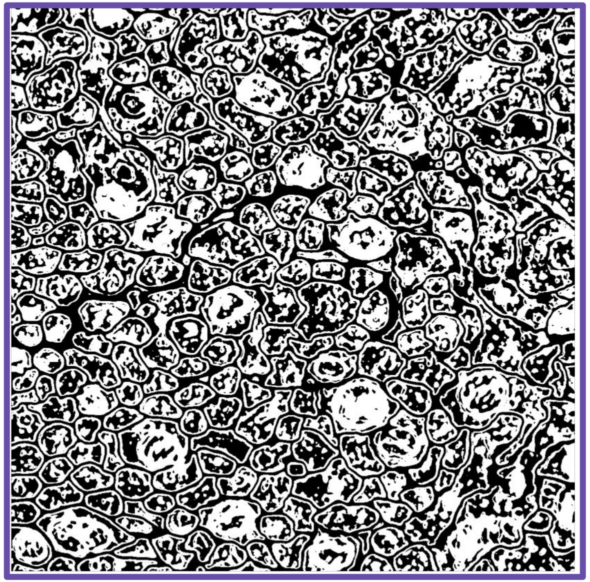
Influence of threshold value



$\theta = 0.25$



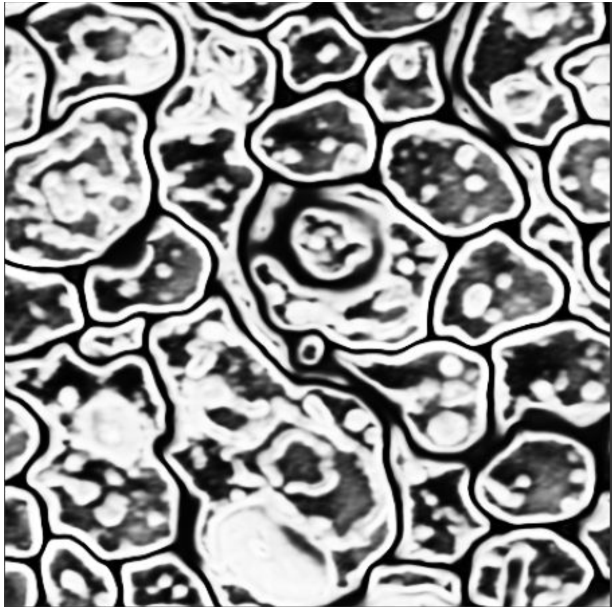
$\theta = 0.5$



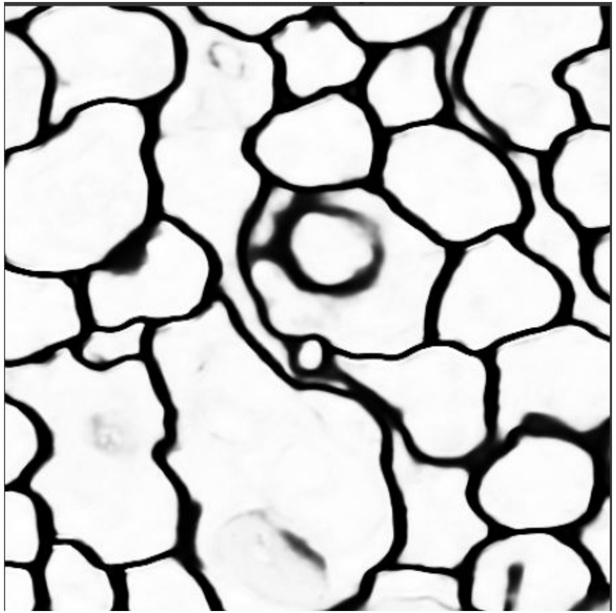
$\theta = 0.75$

Discrepancies between affinities

Prediction difference between X and Y axis:

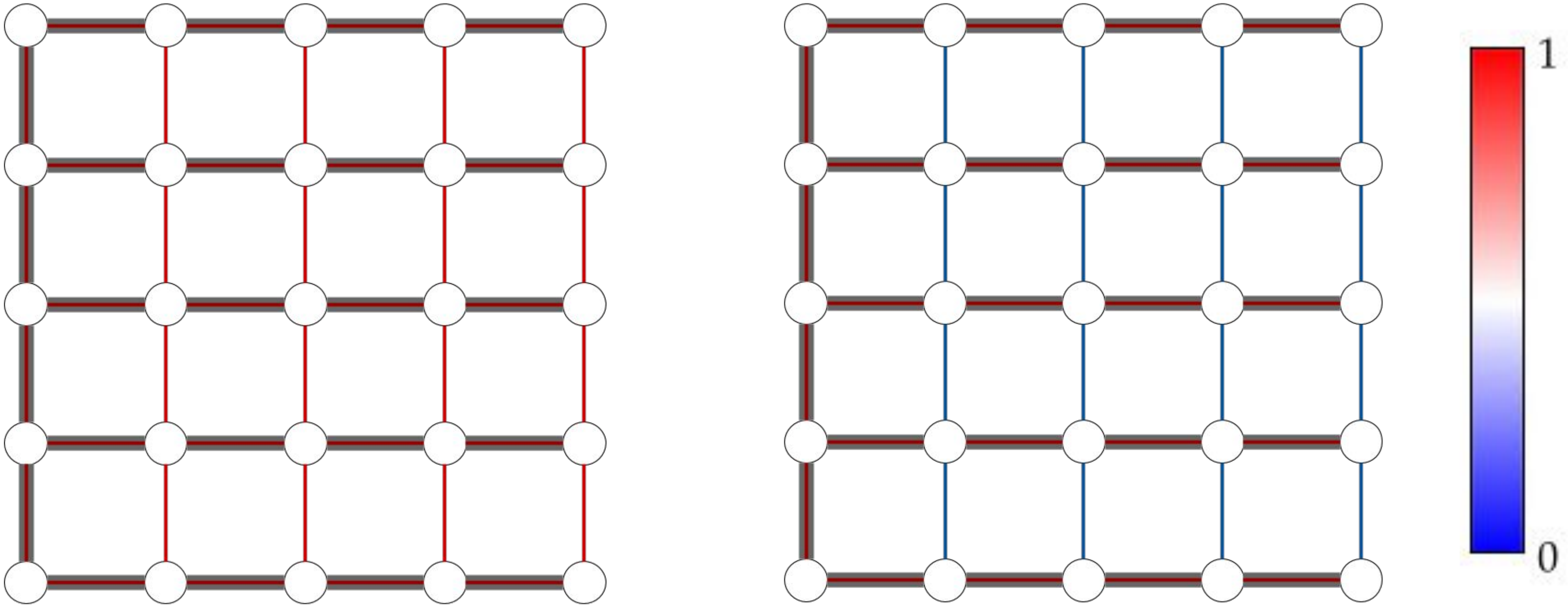


Affinity along the x axis



Affinity along the y axis

Graphs with the same MALIS loss



Both are valid, and give the same loss, but the left one is more desirable

Results: CREMI dataset

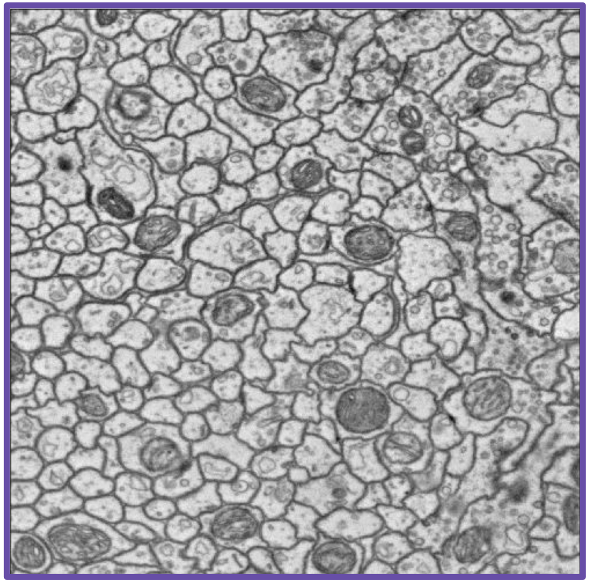
Dataset presentation :

Image size :
1250x1250x125

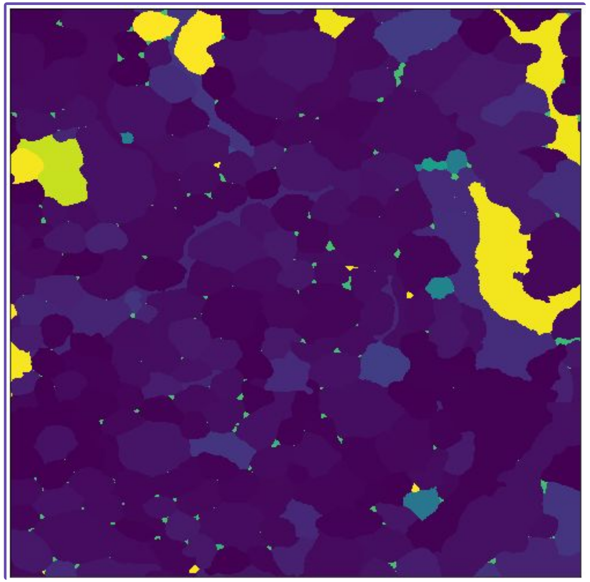
Number of images used : 1

Split train/test : 100/25

Evaluation using a python 2 library



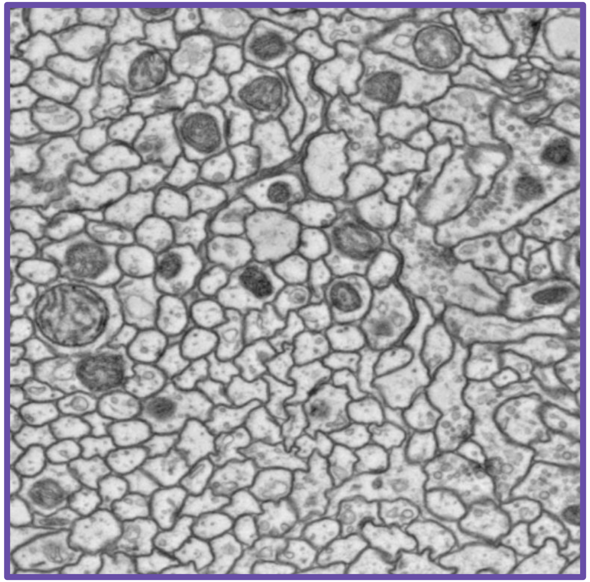
Original image



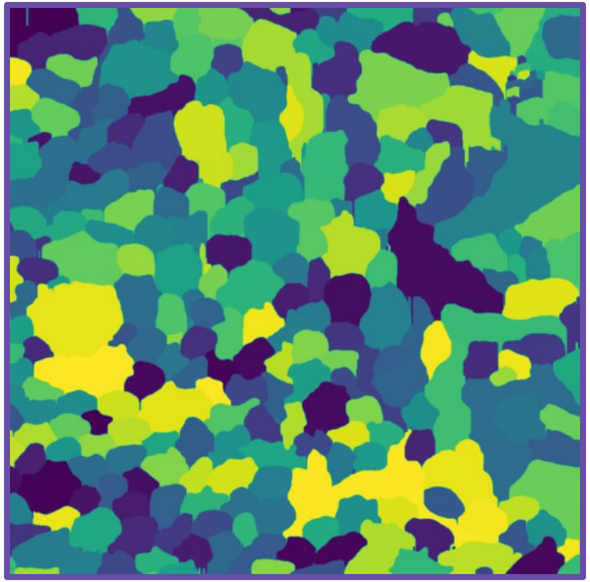
Groundtruth

Results: CREMI dataset

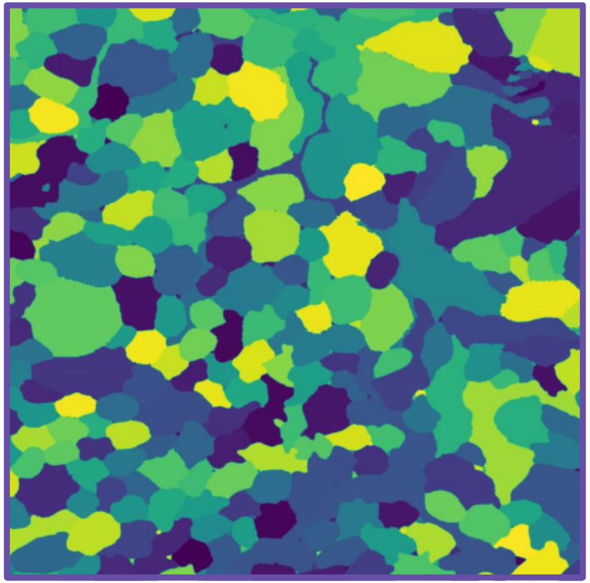
Example results : CREMI volume A (Test set)



Original image



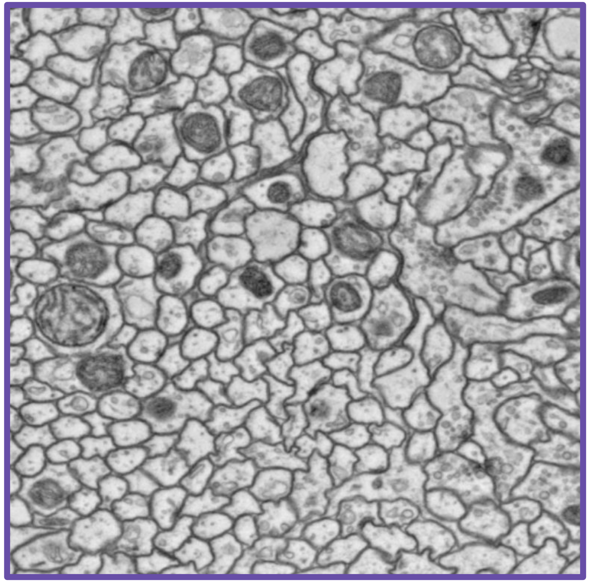
Our segmentation



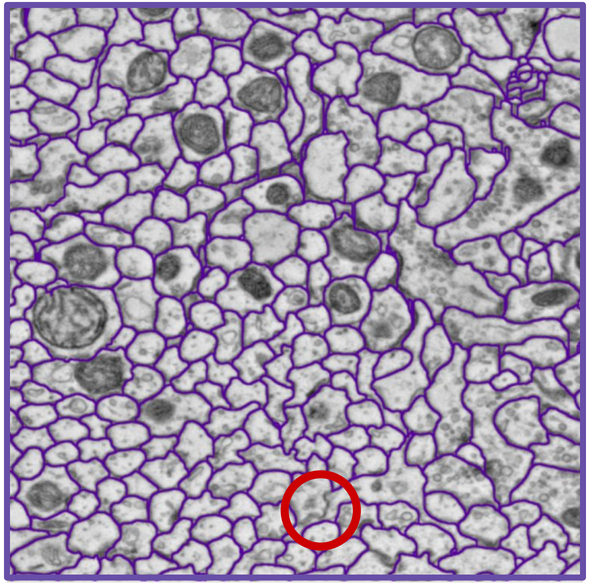
Groundtruth

Results: CREMI dataset

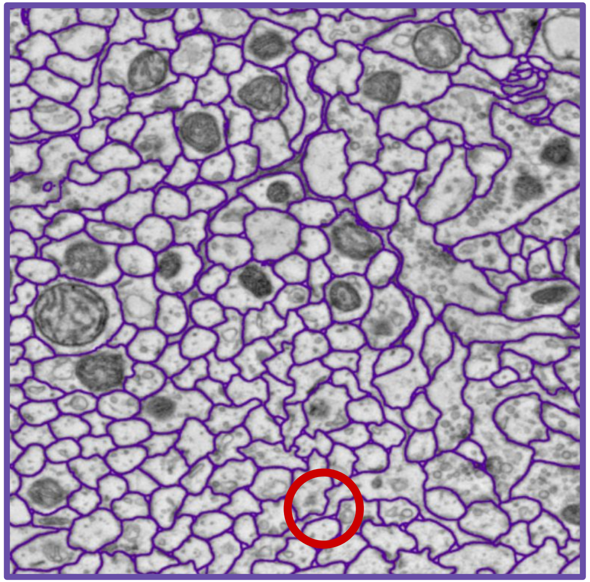
Example results : CREMI volume A (Test set)



Original image



Our segmentation



Groundtruth

Results: Metrics used

Rand index :

$$R(X,Y) = \binom{N}{2}^{-1} \sum_{i<j} |\delta(X_i, X_j) - \delta(Y_i, Y_j)|$$

VOI merge and split :

$$VOI(X,Y) = \underbrace{H(X|Y)}_{\text{VOI split}} + \underbrace{H(Y|X)}_{\text{VOI merge}}$$

CREMI score :

$$CREMI = \sqrt{VOI(X,Y) \times (1 - R(X,Y))}$$

Results: CREMI dataset Training set

Numerical results

	Rand index	VOI merge (lower is better)	VOI split (lower is better)	CREMI score (lower is better)
FCN + MALIS	0.61	1.25	1.03	0.94
U-Net	0.58	0.91	**0.52**	0.78
U-Net MALA	**0.83**	**0.46**	0.54	**0.42**

Results: CREMI dataset Test set

Numerical results

	Rand index	VOI merge (lower is better)	VOI split (lower is better)	CREMI score (lower is better)
FCN + MALIS	0.53	1.57	1.38	1.18
U-Net	0.52	1.07	**0.47**	0.87
U-Net MALA	**0.80**	**0.50**	0.57	**0.46**

SOTA (in 3D)	*0.89*	*0.115*	*0.339*	*0.221*

Project organization

- Meetings (Hangouts)
- Weekly reports
- Trello
- Jupyter notebooks
- Social networks

Difficulties encountered

- Scientific papers
- Basic knowledge
- Higra
- PyTorch
- Quarantine

Conclusion

Work available at : *github.com/garridoq/malis-project*