Quantitative analysis of fluorescent images of glia cells using deep neural networks

Yewen Huang
Academic advisor: Dr. Demetrio Labate

University of Houston
Department of Mathematics

July 11, 2023

1. Background
   - Astrocytes and microglia in the CNS
   - Significance
   - Goals
2. Automated detection of glia cells
   - Deep learning cell detection approach based on YOLOv5
   - Application of YOLOv5
3. Detection results
   - Results of astrocyte detection
   - Results of microglia detection
4. Clustering microglia subpopulations
   - Introduction to the clustering algorithm
   - Application of the clustering method
5. Clustering results

1. Background

Astrocytes in the CNS

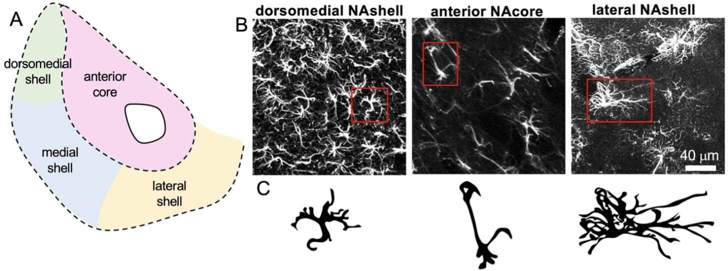
Astrocytes : subtype of glia cells

- The most abundant type of cells in the central nervous system (CNS)
- Complex star-shaped structure
- Diverse abilities and functions reflect by their complex morphology
- Morphological alterations may correlate to traumatic brain injury, infection, autoimmune responses, neurodegenerative diseases, etc
- Active role in neuronal development and function[1] , [2] :

[1] molofsky2012astrocytes.

[2] kruyer2023astrocyte.

Astrocytes in the CNS



Astrocyte cytoskeletal structure across **nucleus accumbens subregions** in rats (A-B). Heterogeneity in density (B) and morphology (C).

Nucelus accumbens : neural interface between motivation and action ; key role on reward, stress-related, drug self-administration behaviors.

Microglia in the CNS

Microglia : subtype of glia cells

- Responsible for immune surveillance, maintaining homeostasis
- Highly dynamic and heterogeneous population of cells
- Cell architecture more diffuse and complicated than astrocytes.
- Morphological alterations in response to injury or infection [3, 4]

[3] Fontainhas2011MicroglialMA.

[4] Davoust2008FromBM.

Microglia in the CNS

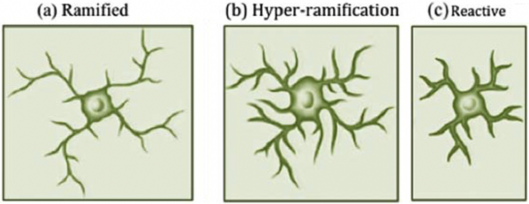


Figure – **Morphological alteration of microglia in different states**

(a) Ramified morphology associated with microglial “resting” state.
(b) Hyper-ramification of microglia (increased microglia process length and branching) in response to a stimulus, injury or inflammation.
(c) Activation microglia in response to tissue damage, injury, or pathological events in the CNS (increased cell body size, thicker processes, altered branching patterns)

Quantitative analysis in microscopy images of glia cells

- Morphological alterations in astrocytes and microglia are closely correlated to their role in the CNS.
- Need to automatically **detect** glia cells and **quantify** morphological alterations
  - to understand precise role in the CNS
  - to advance new therapeutics

Existing image analysis algorithms

- Automated or semi-automated intensity thresholding strategies, i.e., FIJI software[5]
  - performance is inconsistent
  - highly depends on cell density, noise level
- Learning-based algorithms, i.e., FindMyCell algorithm[6]
  - need large training set
  - detection performance is data-dependent

[5] Schindelin2012FijiAO.

[6] Suleymanova2017ADC.

Goals

Main goal : To quantify cellular alterations in high-resolution fluorescent microscopy images of glia cells.

Subgoals (discussed in this talk) :

- Automated detection of GFAP-labeled astrocytes in fluorescent images[7]
- Automated detection of IBA1-labeled microglia in fluorescent images
- Identification of microglia subpopulations in images of spinal cord injury

[7] **Huang2022AutomatedDO**.

2. Automated detection of glia cells

Automated glia cell detection based on YOLOv5

Why YOLOv5 ?

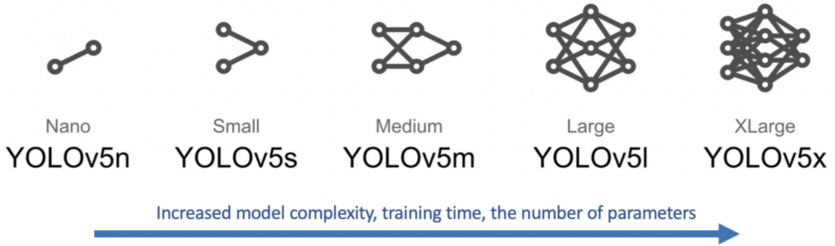
Powerful deep learning platform for **object detection**.

- Flexibility : can handle together images of different dimension/color
- Customization :
  - customization of network architecture
  - selection of different model size
  - data augmentation
  - hyper-parameter optimization
- Convenience : implemented in PyTorch, a powerful Python-based scientific computing package

Different types of YOLOv5 model

YOLOv5 P5 models : design for images with smaller dimension, i.e. 640 × 640

YOLOv5 P6 models : design for images with larger dimension, i.e. 1280 × 1280


Nano
YOLOv5n
Small
YOLOv5s
Medium
YOLOv5m
Large
YOLOv5l
XLarge
YOLOv5x
Increased model complexity, training time, the number of parameters

Network architecture framework

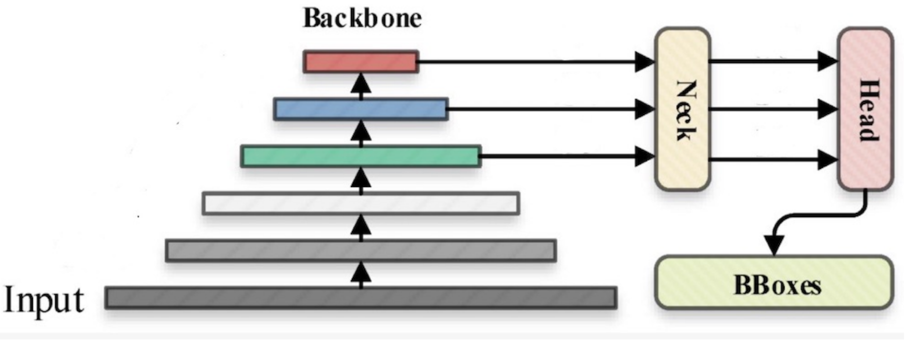
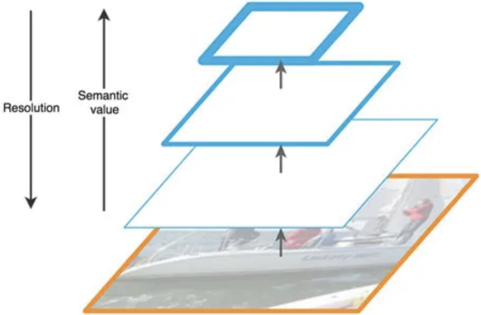


Figure – **Network Pipeline :** Input images are first processed by the **Backbone** for feature extraction (different feature size), then are fed to the **Neck** for feature fusion and finally processed by the **Head** network to generate a model and output detection results (class, score, location, size)

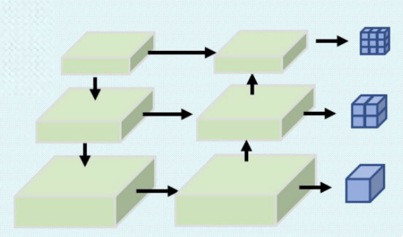
Backbone

It captures features of different sizes from the input image, and discovers hierarchical representations of the image by incorporating both low-level features such as edges and textures and higher-level global context.



Neck

Aggregate feature map of different size, and to improve the model's ability to detect objects of various scales.



- Up-bottom pathway : fusing features from the high-level to propagate fine-grained details
- Botton-up pathway : fusing features from the bottom level to enhance the lower-resolution feature maps

Head

Head : to generate different size of features maps to achieve multiscale prediction

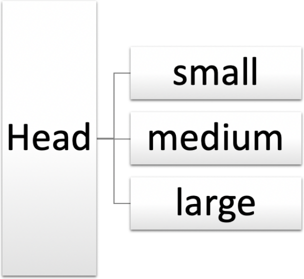


Figure – 3 detection layers, with different detection layers being responsible for predicting bounding boxes and classes at a specific scale.

Image preparation

To train the network, one has to generate labeled images where the objects of interest are identified by **rectangular boxes**.

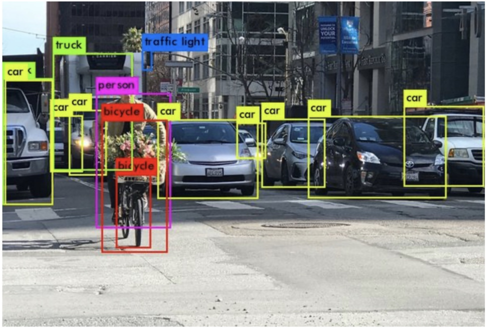


Figure – **Multi-class image dataset** : YOLOv5 can detect up to 80 different classes of objects.

Image preparation

In our case, we only want to detect one object class

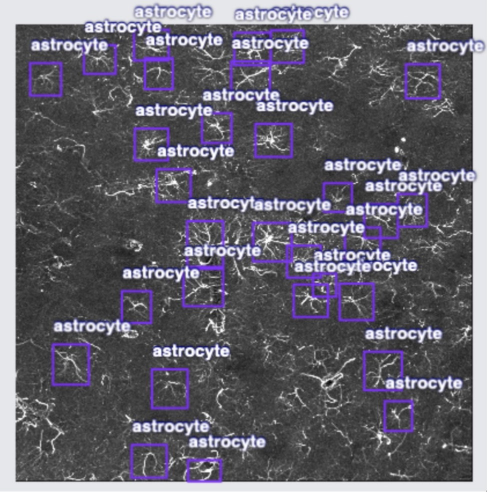


Image preparation

To increase the training size, YOLOv5 employs data augmentation where new training images are obtained by applying transformations to the original training set.

The user can opt out some transformation types.

- Data augmentation transformations
  - **Random Affine Transformations :** including random rotation, scaling, translation, and shearing of the images.
  - **Random Horizontal/Vertival Flip**
  - **MixUp Augmentation :** a method that creates composite images by taking a linear combination of two images and their associated labels.
  - ...

Output

- The predictions from different detection layers of the Head of YOLOv5 are combined to produce the bounding box coordinates for the detected objects together with **detection scores** and **class labels** (if you have multi-class)
- Utilization of post-processing technique, non-maximum suppression[8], to eliminate redundant or overlapping bounding box predictions and retain only the most reliable and non-overlapping bounding boxes.

[8] Neubeck2006EfficientNS.

Loss function

The loss function is critical for the success of object detection algorithms.

In YOLOv5, the loss function consists of 3 components

$$\text{Loss} = L_{box} + L_{obj} + L_{cls}$$

L_{cls} – **class loss :** measuring the error related to the class assignment

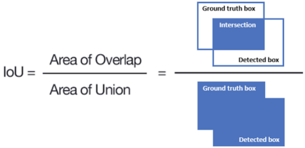
NOTE : In general, YOLOv5 can detect several object classes.
In our application, we only detect one class of cells at a time, e.g., astrocytes or microglia

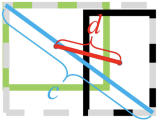
L_{box}

L_{box} – **location loss :** measuring the error related to the location and size of the bounding box based on Intersection over Union (IoU) :

$$L_{box} = 1 - IoU + D + \alpha V$$

where

- 



- $D = \dfrac{d^2}{c_2}$ is the normalized central point distance
- V is a penalty term accounting difference in aspect ratio and sizes between boxes, and α is a trade-off parameter

L_{obj}

L_{obj} – **objectness loss :** measuring the error in detecting whether an object is present in a particular grid cell or not, based on Binary Cross Entropy (BCE) loss

$$L_{obj} = t_0 \times (-\log(sigmoid(p_0)) + (1 - t_0) \times (-\log(1 - sigmoid(p_0)))$$

where

- t_0 = ground truth label,
  - $t_0 = 1$, if there is an object in the cell
  - $t_0 = 0$, if there is no object in the cell
- p_0 = predicted confidence score

Fitness function

YOLOv5 uses a Fitness Function for hyperparameter optimization

Fitness Function : a weighted combination of four loss metrics : (precision, recall, mAP@0.5[9], mAP@0.5 :0.95[10]).

- hyper-parameter evolution process : compute the fitness score for each hyper-parameter of different values, higher scores of fitness indicate better performance

[9] mAP@0.5 : mean Average Precision at an IoU threshold of 0.5

[10] mAP@0.5 :0.95 : mean Average Precision calculated across a range of IoU thresholds, typically from 0.5 to 0.95 with a step size of 0.05.

Hyper-parameter evolution

- 30 hyper-parameters : i.e.
  - initial learning rate
  - image rotation
  - image HSV (Hue Saturation Value)
  - momentum
- Application of Genetic Algorithm[11] for hyper-parameter optimization
- The best hyper-parameter selection comes with the maximization of the fitness score within a number of generations

[11] Katoch2020ARO.

Image Datasets

Images

- Astrocytes :
  - BBBC dataset[12]
  - Kruyer dataset[13] : GFAP-labeled fluorescent images from 5 different subregions of the nucleus accumbens of adult rats
- Microglia : Geoffroy dataset[14] : IBA1-stained images of microglia from spinal cord region of adult rats

Labels

TXT format files storing the coordinates of the rectangular boxes containing the objects of interest, each label is a vector of the form

$$(class, x\_center, y\_center, width, height)$$

[12] Publicly available images from the Broad Bioimage Benchmark Collection

[13] Images from Dr Kruyer, Department of Neuroscience, University of Cincinnati

[14] Images from Dr Geoffroy, Texas A&M

Dataset

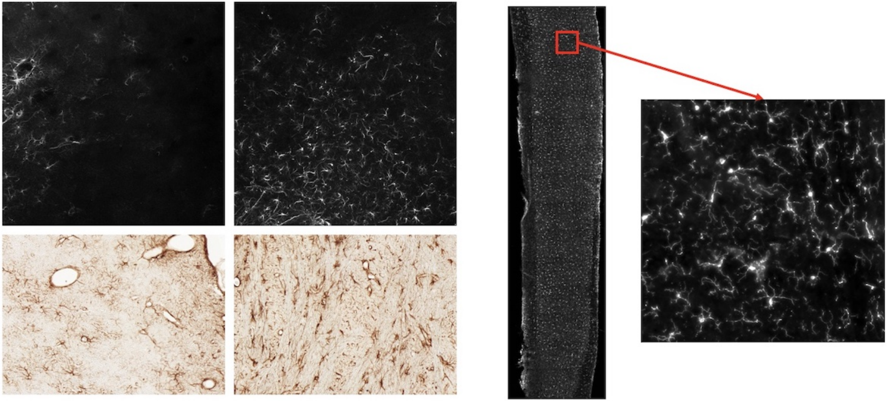


Figure – **Image datasets.** Two representative images from the Kruyer dataset (left top row) and two representative images BBBC dataset (left bottom row). The Geoffroy dataset (middle) and an example of extracted patch (right).

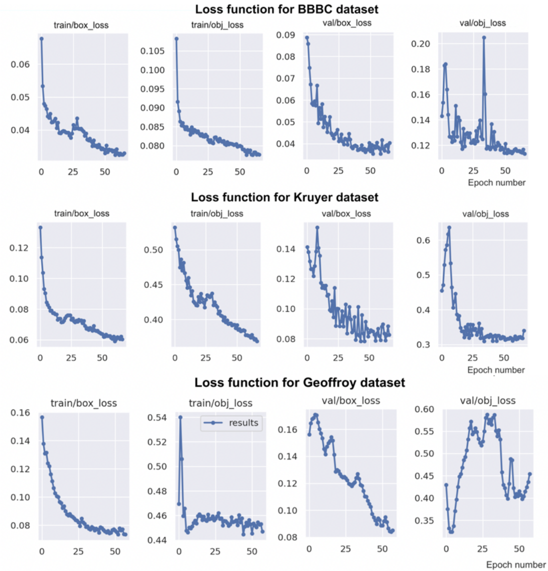
Dataset split

	BBBC	Kruyer	Geoffroy
training	783	169	59
valid	225	21	10
test	31	21	10
total	1039	211	79
average cell per image	15	25	36
cell range perimage	(10, 30)	(6, 65)	(20, 50)

Training setting

	epochs	batch size	hyperparameters	Optimizer
BBBC	66	32	evolution	Adam
Kruyer	67	16	evolution	Adam
Geoffroy	58	16	same as Kruyer dataset	Adam

Loss function



Evaluation metrics

- **Precision** : measure the proportion of correctly detected cells over all detected cells

$$\mathrm{P} = \frac{TP}{TP+FP}$$

- **Recall** : measure the proportion of correctly detected cells over the total number of cells

$$\mathrm{R} = \frac{TP}{TP+FN}$$

- **Dice coefficient** : measure the overall effectiveness

$$\mathrm{DC} = \frac{2TP}{2TP+FP+FN}$$

- **PR (precision-recall) curve** : the precision (y-axis) of a model varies as a function of recall (x-axis). An assessment of the overall performance by illustrating the trade-off between the recall and precision.

3. Detection results

Astrocyte detection results

We considered other astrocyte detection methods for comparison

- Adaptive thresholding[15] : Fiji software
- GESU-net[16] : an algorithm for astrocyte detection and segmentation
- FindMyCell[17] : a deep learning platform (It was applied only to BBBC dataset)

[15] Healy2018ThresholdbasedSO.

[16] Kayasandik2020AMD.

[17] Suleymanova2017ADC.

Astrocytes detection performance

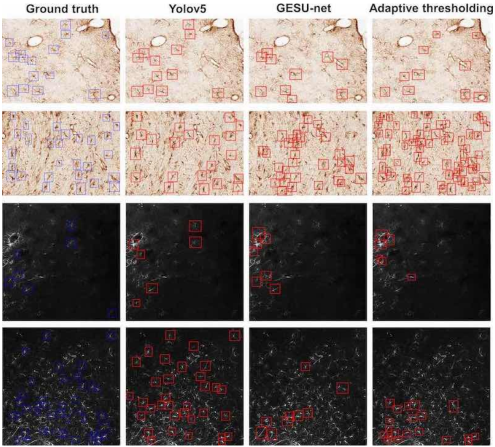


Figure – **Astrocytes detection performance.** Detection performance from different algorithms shown on 2 representative images from BBBC dataset (top two rows) and 2 representative images from Kruyer dataset (bottom two rows).

PR curve

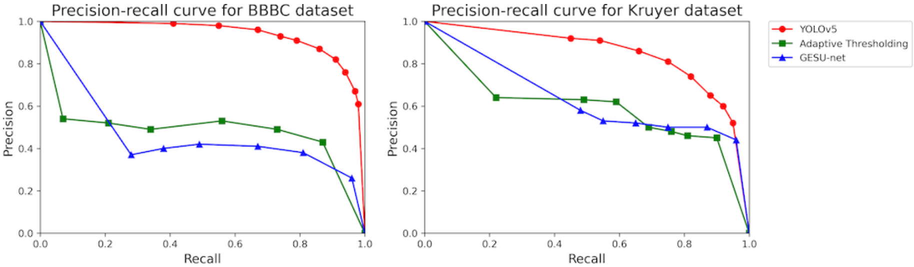


Figure – **Precision-recall curves.** Precision-recall curves illustrate the astrocyte detection performance of different algorithms on the BBBC (left panel) and Kruyer (right panel) image datasets.

Detection performance comparison

Dataset	YOLOv5			Adaptive Thresholding			GESU-net		
	R	P	DC	R	P	DC	R	P	DC
BBBC (484 cells)	0.87	0.86	0.86	0.73	0.49	0.58	0.84	0.38	0.52
Kruyer (565 cells)	0.75	0.81	0.78	0.59	0.62	0.60	0.87	0.50	0.63

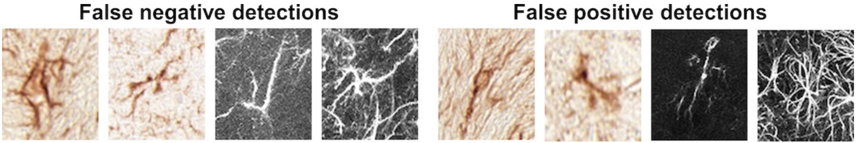
Table – Astrocyte detection performance corresponding to best Dice coefficient of our algorithm based on YOLOv5, Adaptive Thresholding and GESU-net.

Dataset	YOLOv5			FindMyCell		
	R	P	DC	R	P	DC
BBBC (188 cells)	0.85	0.78	0.82	0.70	0.86	0.77

Table – Astrocyte detection performance corresponding to best Dice coefficient of our algorithm based on YOLOv5 and FindMyCell.

Discussion for astrocyte detection results

- The YOLOv5-based algorithm for astrocyte detection provides a powerful tool for the quantitative analysis of astrocytes.
- The algorithm can be easily adapted to other types of glial cells.
- Some representative examples of misclassified cells resulting from our algorithm on the test images
  - cell morphology is rare or the cell is similar to objects often assigned to the background
  - cell could have been missed in the ground truth by the image annotators


False negative detections

False positive detections

Microglia detection performance

Application of YOLOv5-based methods on microglia dataset achieves P= 0.91, R = 0.76, DC = 0.83

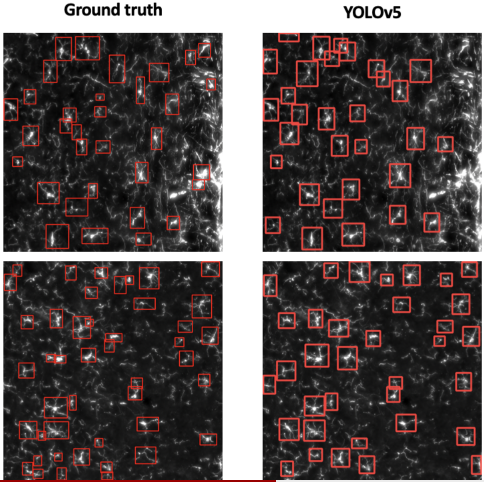


Figure – **Microglia detection performance.** Microglia detection performance on two representative images from the Geoffroy dataset.

Microglia detection on spinal cord injury

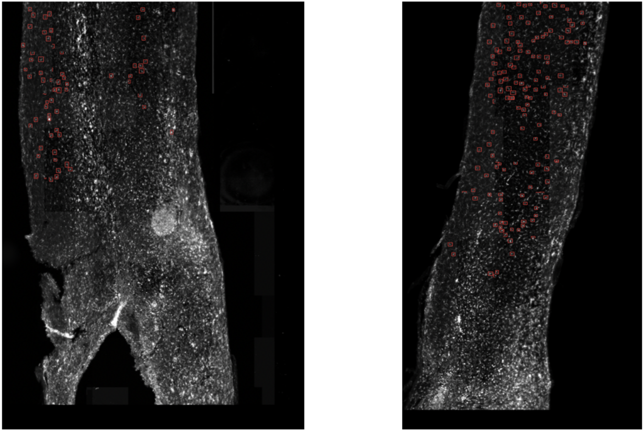


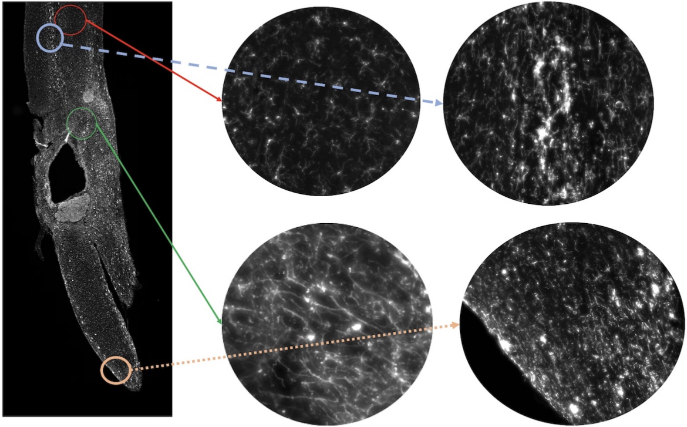
Figure – Application of the model trained on Geoffroy dataset to another two microglia images of spinal cord injury.

Discussion for microglia detection results

- Microglia detection results indicates that cell morphology in the Geoffroy dataset is comparable to the morphology of cells away from injury.
- In images with spine cord injury, microglia show morphological heterogeneity
- Microglia closer to injury is more dense and exhibit their more elongated shape
- Rather than working on single cells detection, we will examine local properties using image patches containing approximately a single cell to identify subpopulations of microglia.

4. Clustering microglia subpopulations

Microglia in the images of spinal cord injury



Microglia morphology is sensitive to spine cord injury - cells closer to the injury are morphologically different than cells away from the injury

Cell segregation

Can we segregate microglia subpopulations depending on proximity to injury ?

- Apply clustering algorithm
- Adapt features to incorporate rotation and translation invariance

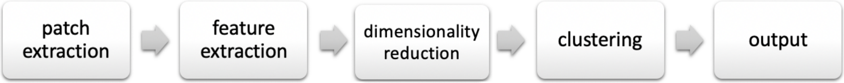


Figure – Pipeline of the microglia segregation algorithm

Patch extraction

- Estimate average cell size
- Extract patches containing about one cell
- Patch overlap allowed

Feature extraction using Kymatio

Kymatio[18] :

- Software library to compute wavelet scattering coefficients
- It captures image structural characteristics occurring over multiple scales
- It incorporates translation and rotation invariance

[18] **Andreux2018KymatioST**.

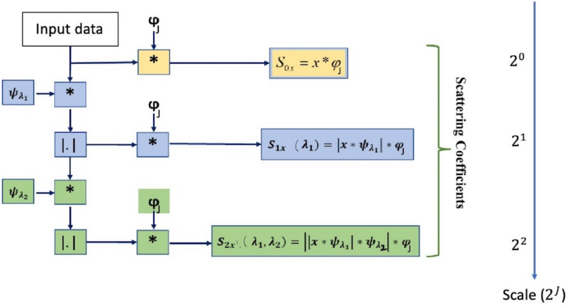
What is scattering wavelet transform ?

The scattering transform maps functions in $L^2(\mathbb{R}^2)$ into a set of coefficients associated with multiple scales 2^j, rotations θ, locations k

$$f \to \langle f, \psi_{j,\theta,k} \rangle$$

- Translation invariance : Use modulus nonlinearities are used to capture local information while also achieving local translation invariance
- Multi-Scale : Through a cascade of wavelet transforms and modulus operations, the transform captures multiscale information by iteratively decomposing input data at different scales and combining the resulting coefficients
- Rotation invariance : Compute the magnitude of the wavelet response at different orientation and then perform averaging across orientations to obtain rotation-invariant features.

Diagram of second-order scattering wavelet transform

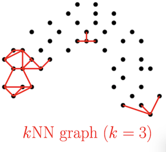


- * : convolution
- $|\cdot|$: non-linearity modulus operation, representing the magnitude of the high-frequency content
- φ_J : lowpass filter has spatial support that is proportional to 2^J
- ψ_{λ_i} : wavelet filter banks, with λ_i be the frequency index

Dimensionality reduction using Isomap

Isomap : manifold learning algorithm to preserve the geodesic distance between samples while reducing the dimension

1. Build distance matrix $M_{m,m}$ using Euclidean distance for each pair of the data.
2. Build a k-neighborhood graph G by connecting only "nearby points" with edges weighted by their Euclidean distances



kNN graph ($k = 3$)

3. In the graph G, solve the all pairs shortest path problem, set $M(i, j) =$ geodesic distance (v_i, v_j).
4. Apply Multidimensional Scaling (MDS)[19] to find low-dimension projection.

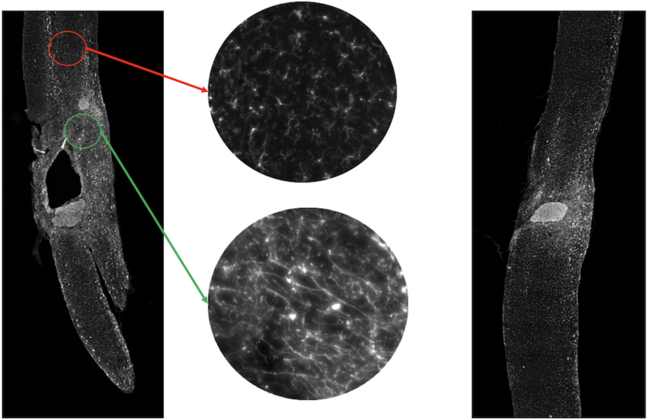
[19] Hout2013MultidimensionalS.

Clustering using GMM

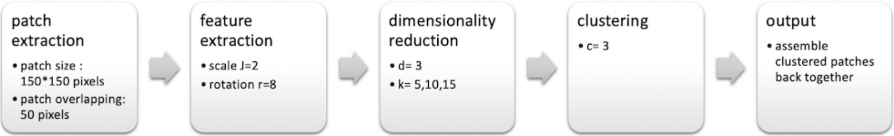
Gaussian mixture model (GMM) : a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters

Dataset

Two images of microglia from spinal cord injury



Parameter setting



patch extraction
- patch size : 150*150 pixels
- patch overlapping: 50 pixels

feature extraction
- scale J=2
- rotation r=8

dimensionality reduction
- d= 3
- k= 5,10,15

clustering
- c= 3

output
- assemble clustered patches back together

5. Clustering results

3 clustering results

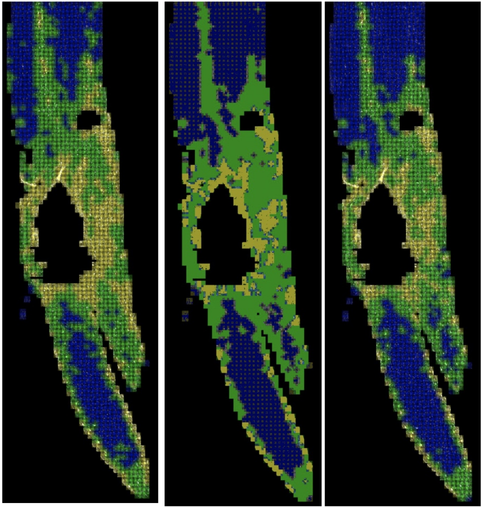


Figure – 3 clusters results for k=5, k=10, k=15 respectively

3 clustering results

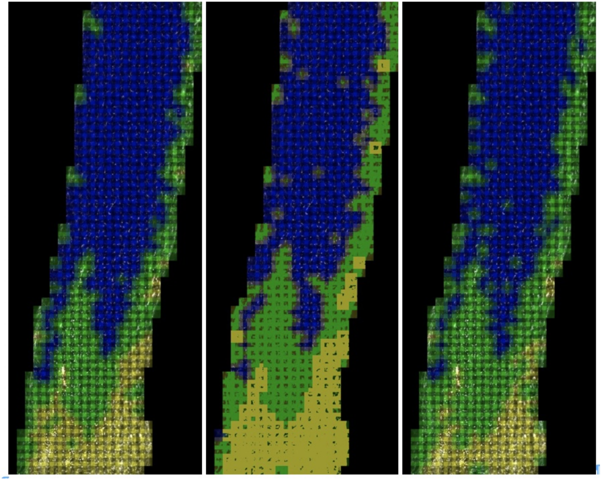


Figure – 3 clusters results for k=5, k=10, k=15 respectively

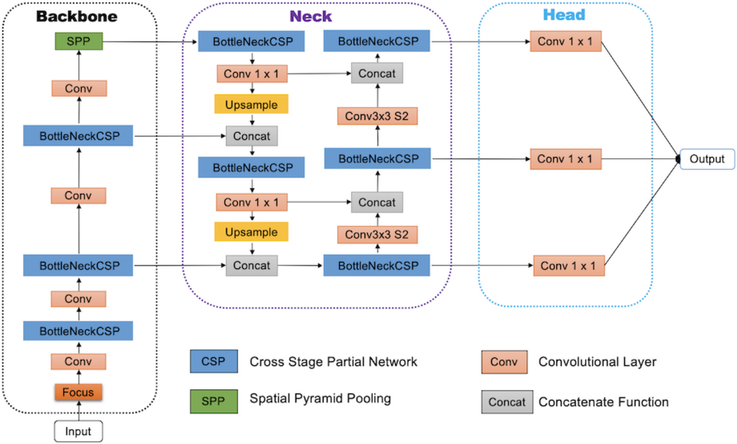
Discussion for the clustering results

- The clustering algorithm is able to segregate microglia subpopulations based on morphological characteristics
- Transition regions show mixture of populations
- Confirms the presence of distinct subpopulations.
- Further biological validation needed (work in progress)

Thank You !

Questions ?

YOLOv5 complete structure



CSP	Cross Stage Partial Network	Conv	Convolutional Layer
SPP	Spatial Pyramid Pooling	Concat	Concatenate Function

Backbone main blocks

BottleNeckCSP : extract features in different scale which is capable of capture both local and global context information by a lot of convolution layers.
SPP : capture features at different levels to produce multiscale representation by several max pooling operations with different size, and to improve the model's ability to handle objects of various scales.

Head section supplement information

Using the predefined anchor box with different scale and aspect ratio to do the detection in the three detection layers. In each detection layer, has 3 different anchor box in each grid cell.

Mish activation function

The Mish activation function used in the block helps in introducing non-linearity to learn more complex relationships between features, and preserving gradient information during backpropagation to lead to more stable and efficient training, and improving the model's performance. It is defined as :

$$Mish(x) = x \times \tanh(\log(1 + e^{x}))$$

Adam optimizer

Adam optimization combines the concepts of gradient descent optimization (momentum) and adaptive learning rates (RMSProp) to efficiently update the model's parameters during training, and is well suited for problems that are large in terms of data/parameters.

- First moment (momentum) : use the exponentially moving average of past gradients to accelerate the optimization process
- Second moment (RMSPorp) : use the exponentially moving average of past squared gradients to help track the magnitude of the gradients
- Use the first and second moments to adaptively adjust the learning rate for each parameter, and ensure a smooth and fast convergence

Translation invariance of the sattering wavelet transform

$$W[\lambda]f(x) = |f * \psi_{j,k}(x)|$$

$$W[\lambda]T_y f(x) = W[\lambda]f(x-y) = |f(x-y)*\psi_{j,k}| = T_y W[\lambda]f(x) = |T_y \psi_{j,k} * f|$$

And convolution commutes with translations

$$T_y(f * g) = (T_y f) * g = f * (T_y g)$$

The averaging filter plays a crucial role in the wavelet scattering network. When you create a wavelet scattering network, you specify the invariance scale. The network is invariant to translations up to the invariance scale. The support of the scaling function determines the size of the invariant in space.

Isomap's topological stability problem

Selecting k larger or smaller may result in "short circuits" between distant elements or may infer multiple disconnected components when building the neighborhood graph.

If different regions have different densities, or if there is considerable amount of noise, Isomap fails to reconstruct correctly the exact structure of the embedding.

Dimensionality reduction using b-Isomap

Degree-bounded Isomap (b-Isomap) :Using the degree-k-bounded minimum spanning tree[20] (k-MST) which restricts the degree of every vertex to be at most k.

1. Build the distance matrix $M_{m,m}$: for all elements v_i, $i \in [1..m]$, set $M(i,j) = rd(v_i, v_j)$.
2. Build the MST for the data described by $M_{m,m}$. Select a root node r for the tree.
3. Starting from r do recursively for all non-leaf nodes v : Assume that (v, v_1), (v, v_2), ..., (v, v_d), are the edges in increasing weight from v to its children. If $degree(v) > k$, replace the edges (v, v_2), (v, v_3), ..., (v, v_{d-k+1}) with the edges (v_1, v_2), (v_2, v_3), ..., (v_{d-k}, v_{d-k+1}). This generate k-MST.
4. In the graph defined by k-MST, solve the all pairs geodesic distance.
5. Apply MDS to find low-dimension projection.

[20] Minimum spanning tree (MST) : a subset of the edges of a connected, edge-weighted undirected graph that connects all the vertices together, without any cycles and with the minimum possible total edge weight.