

DeepCut

Unsupervised Segmentation using Graph
Neural Networks Clustering

Anna Charles & Nolan Sisouphanthong

1.Context

Introduction

DeepCut ?

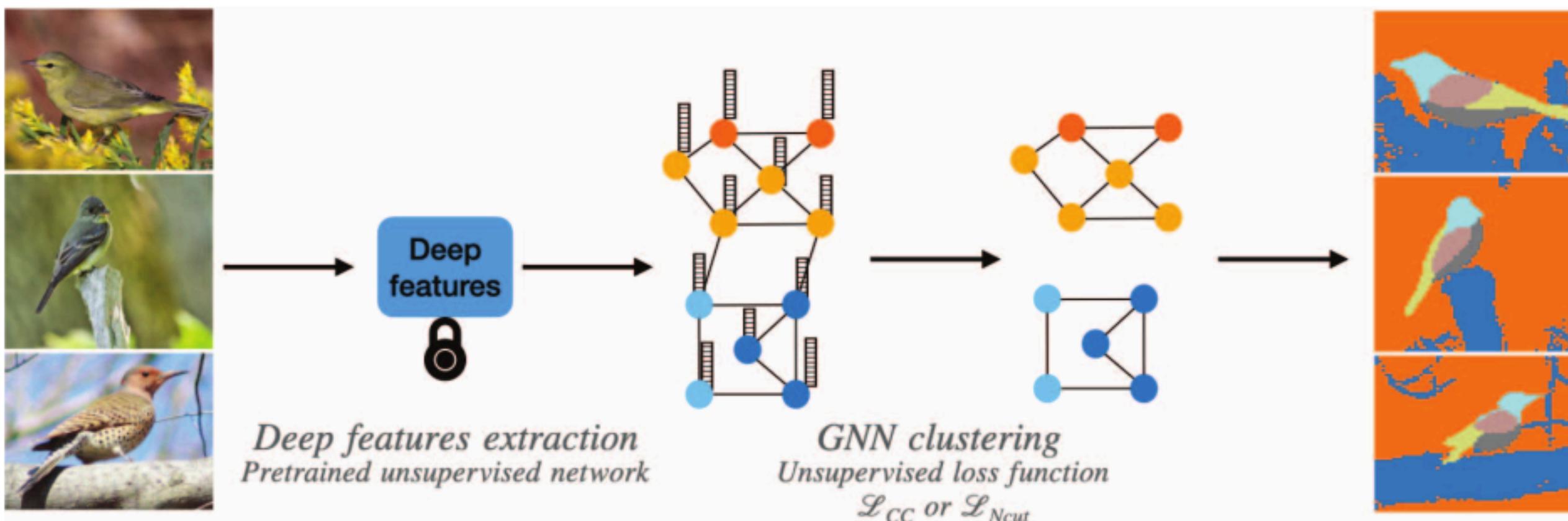
Unsupervised method to perform image segmentation

How ?

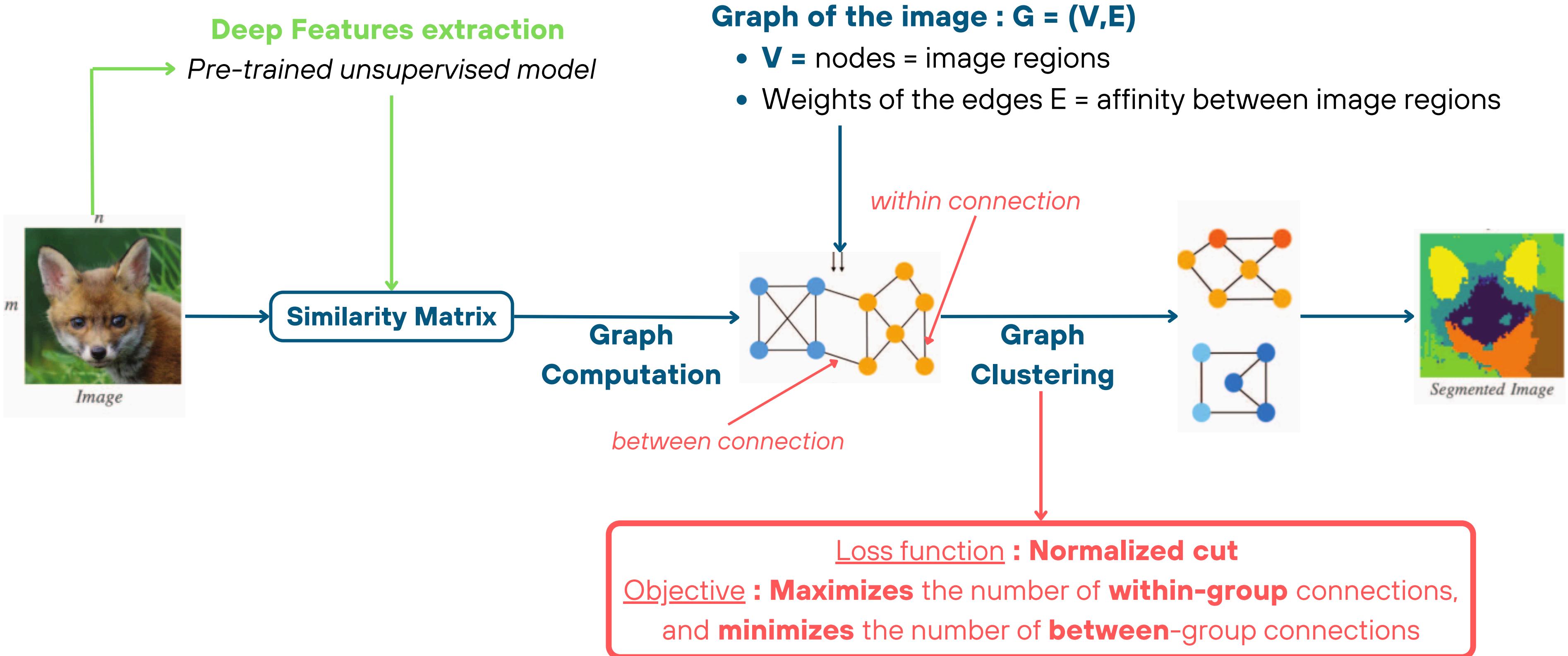
Graph Neural Networks (GNNs) + classical graph clustering

Advantages ?

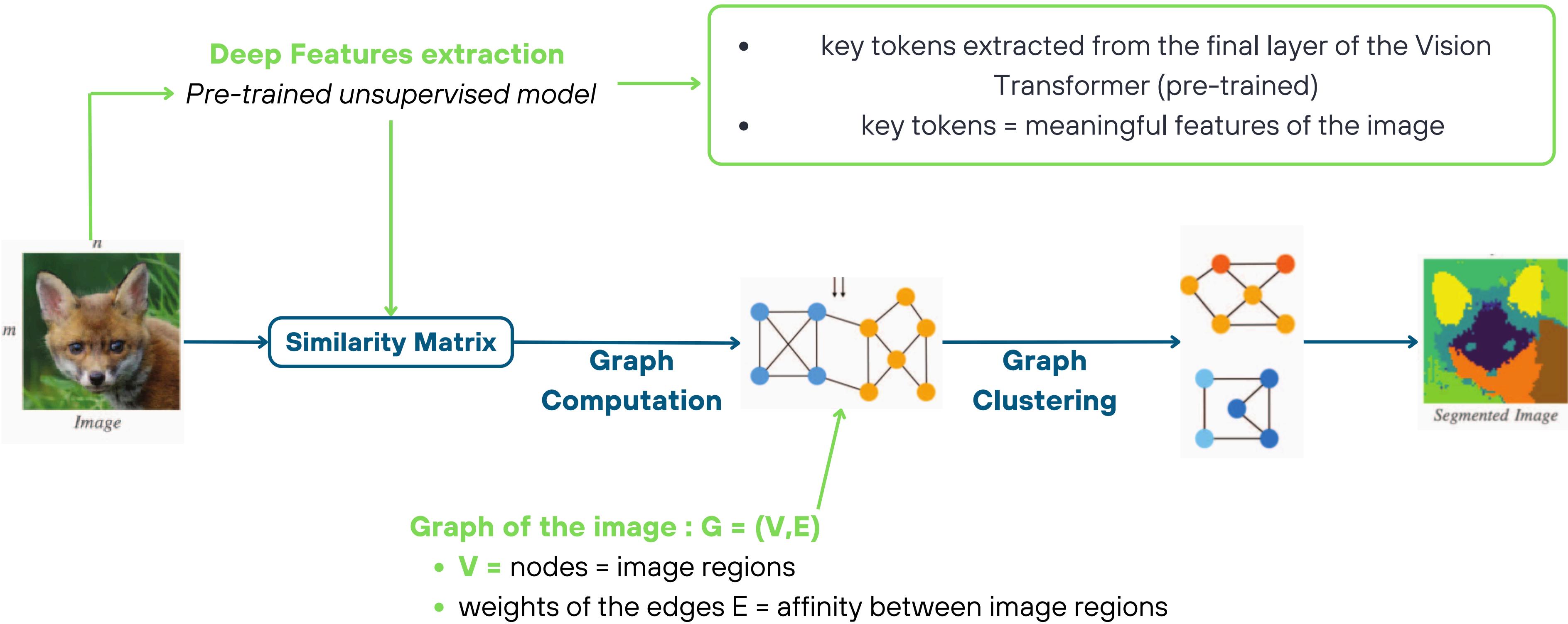
- “K-less” clustering --> **flexibility**
- Does not require annotated data --> **less costly**
- Utilization of deep features of the images --> more accurate results



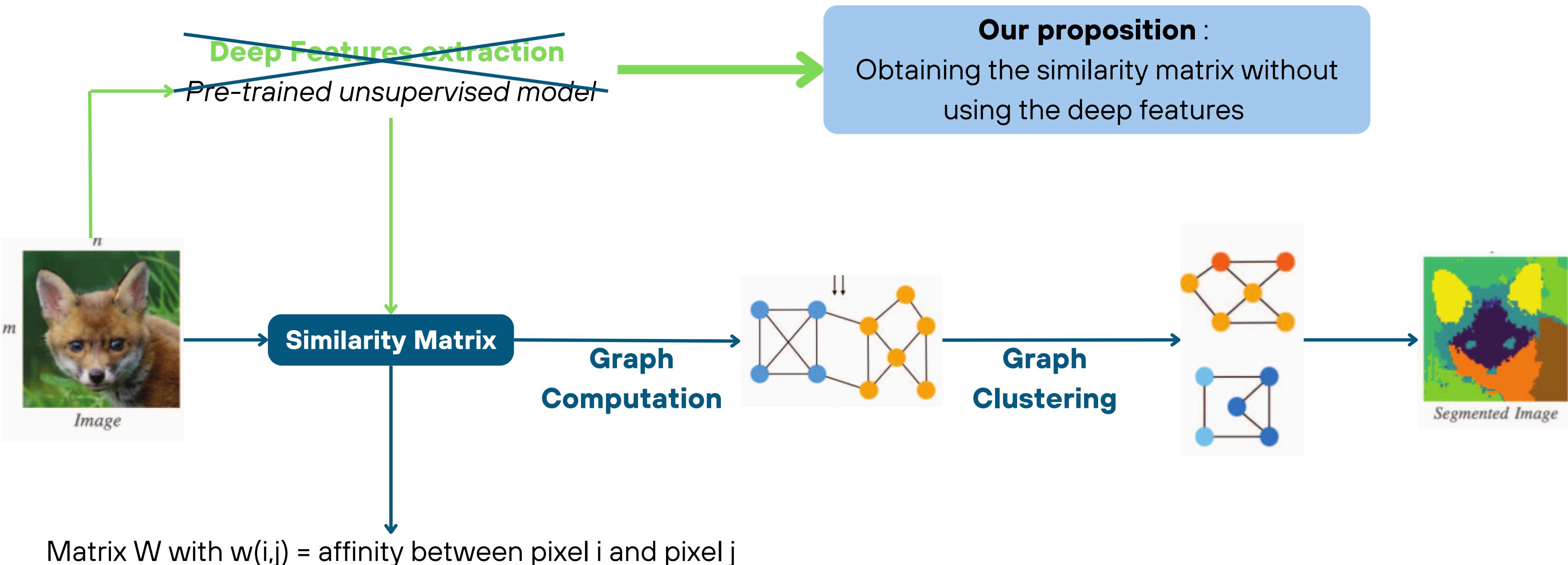
2. DeepCut architecture



2. DeepCut architecture



3. Our proposition



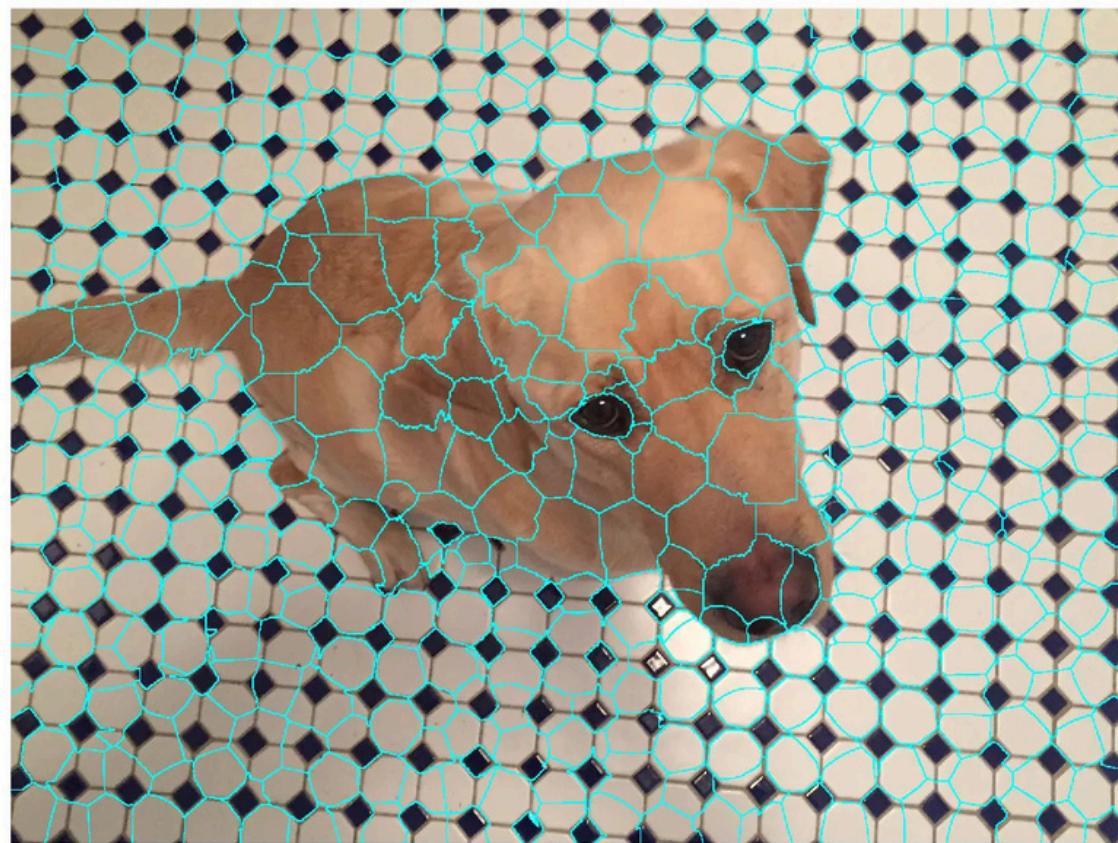
4. Computation of the similarity matrix

Our proposition :

Obtaining the similarity matrix using only the information given by the pixels
Can we still get a good segmentation ?

Step 1: Choice of the image regions -> nodes of the graph

Step 2 : Calculation of the affinities between image regions -> weights of the graph



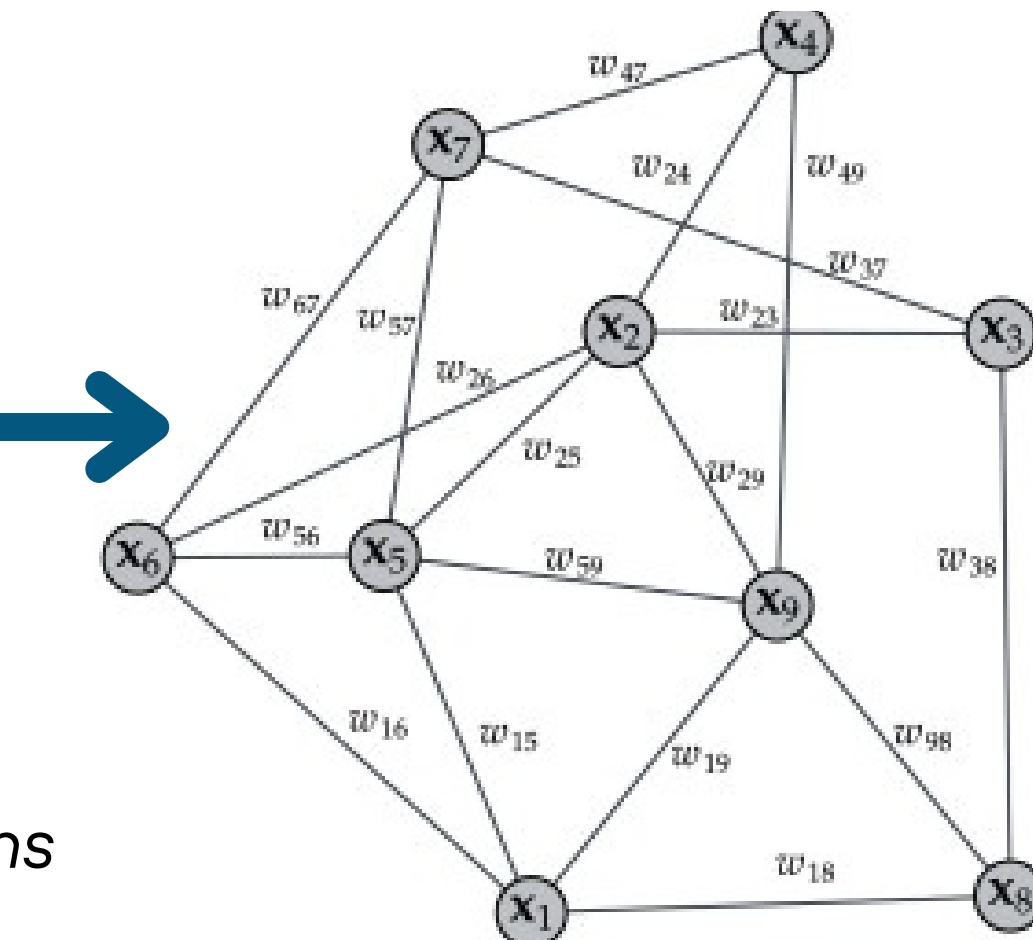
$$W = \begin{pmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,n} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n,1} & w_{n,2} & \cdots & w_{n,n} \end{pmatrix}$$

W : matrix of affinities

n = number of super-pixels/image regions

Step 1

Step 2



4. Computation of the similarity matrix

Step 1:

Method 1: SLIC algorithm : segmenting the image into K regions, “super-pixels”

SLIC Algorithm :

Initialization : K initial super-pixel centers, spaced evenly across the image.

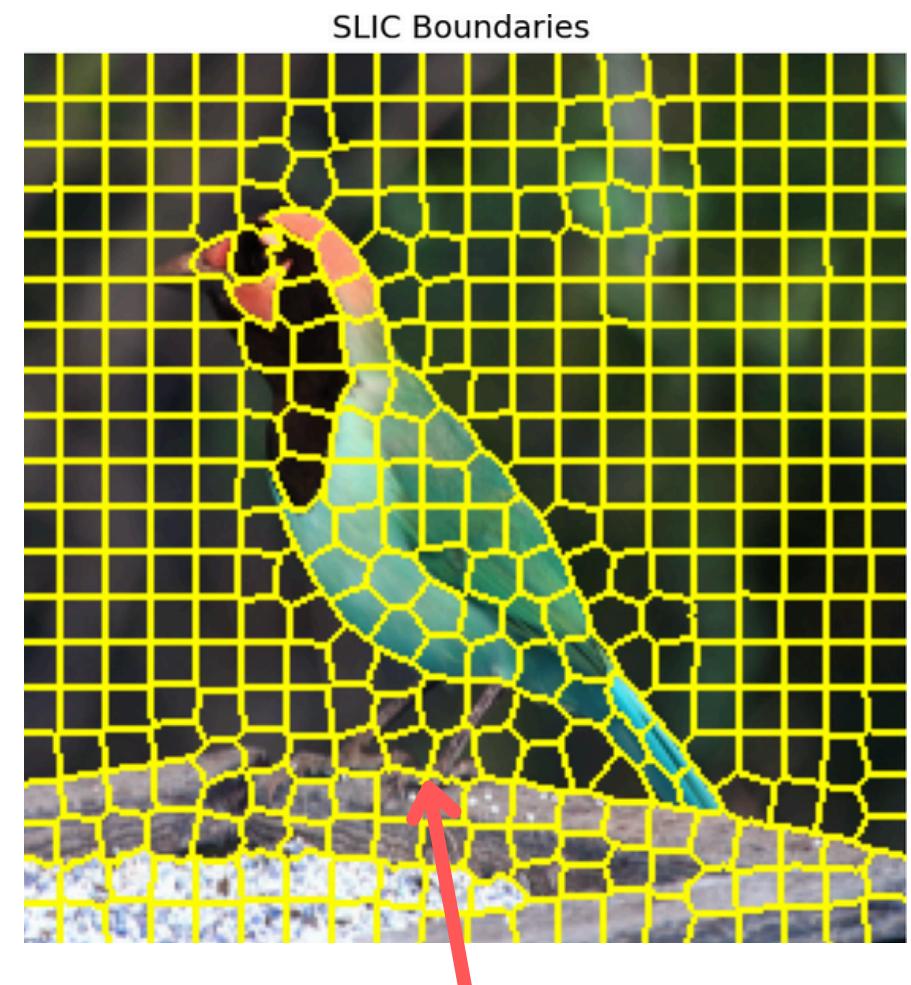
Each super-pixel is in a 5-dimensional space:

- 2D spatial coordinates (x, y)
- color information (L, a, b)

Iteration : Each pixel is assigned to the nearest cluster center based on a weighted **combination of color distance and spatial distance**

+ centers updated

-> Reduces the image size while preserving meaningful features



Pixels in the same super-pixel share similar visual properties

4. Computation of the similarity matrix

Step 2 : Calculation of the affinities between image regions



Affinity between 2 regions represents how similar or dissimilar two regions are :

$$W = \begin{pmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,n} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n,1} & w_{n,2} & \cdots & w_{n,n} \end{pmatrix}$$

W : matrix of affinities

n = number of super-pixels/image regions

$$w_{i,j} = \exp\left(-\frac{|c_i - c_j|^2}{\sigma}\right)$$

c_i, c_j : the color mean of i and j

i, j : the image regions :

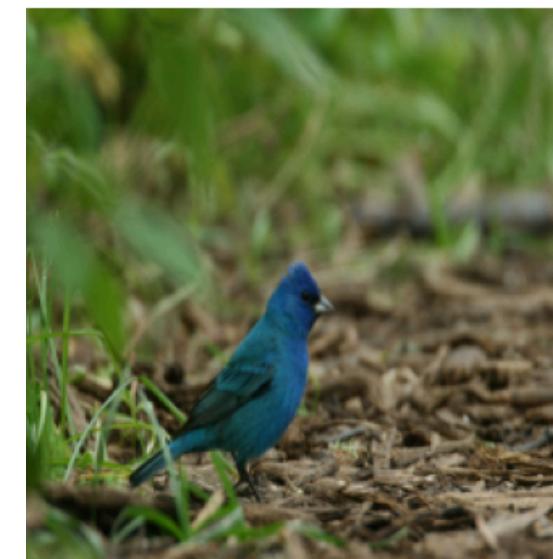
- **super-pixels or pixels**
- represent the nodes of the graph



Function `graph.rag_mean(mode='similar')` from the module `scikit-image`

4. Computation of the similarity matrix

Alternative: Using directly pixels and downsampling the image for easier computation



400x400 image

Each pixel is its own
image region

size of the matrix = $(400 \times 400)^2 = 256 \times 10e8$

Working directly on **pixel** can be tricky

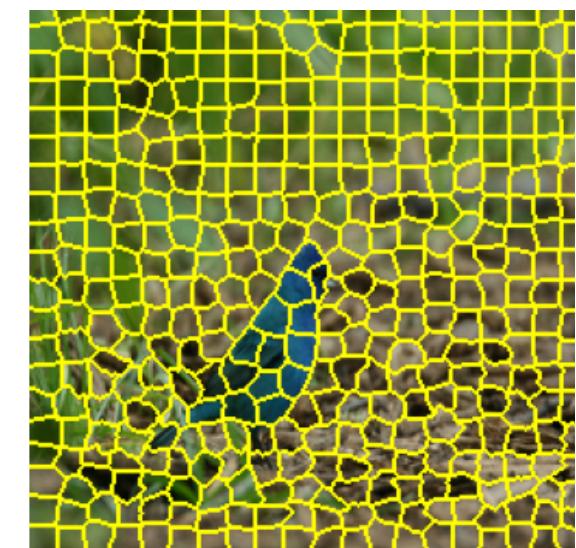
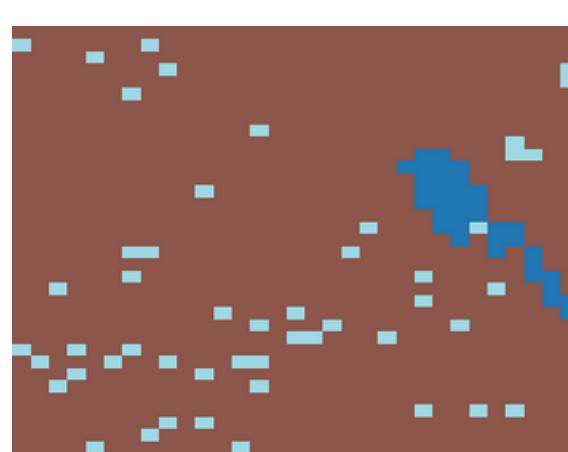
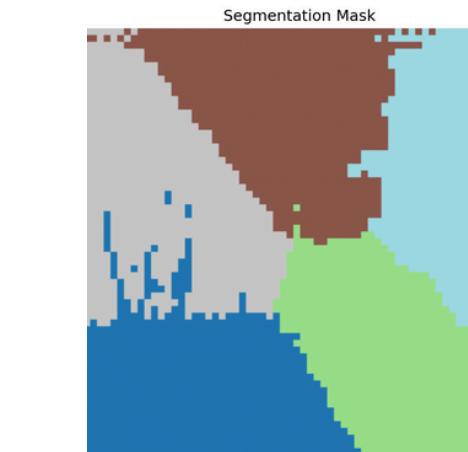
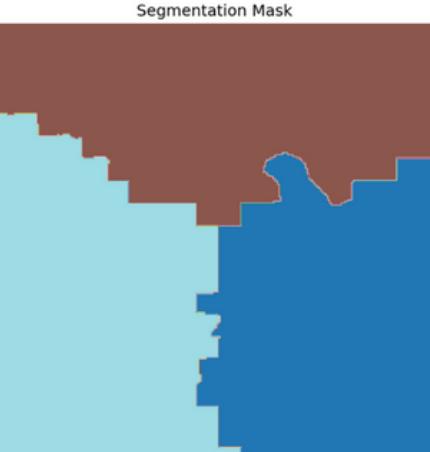
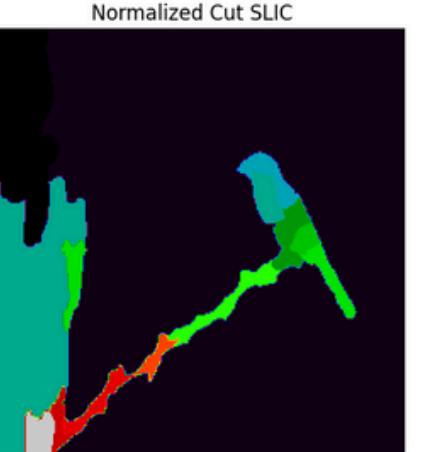
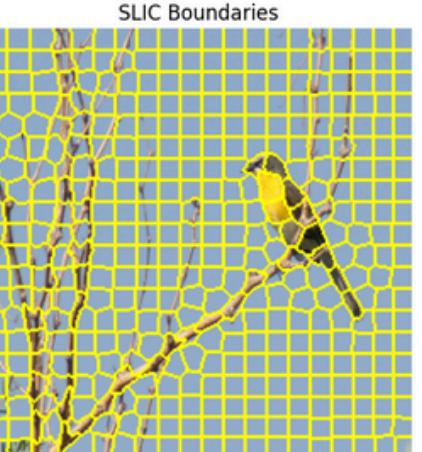
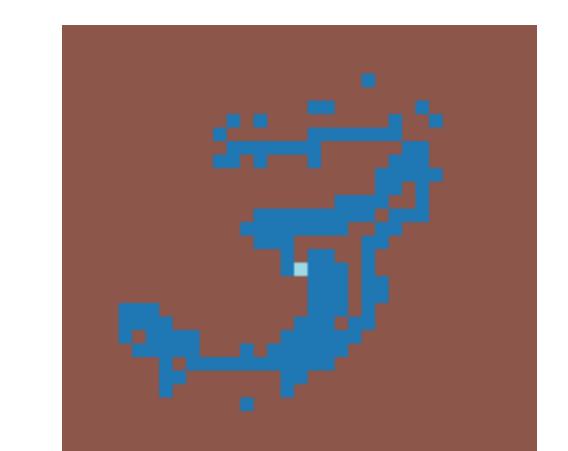
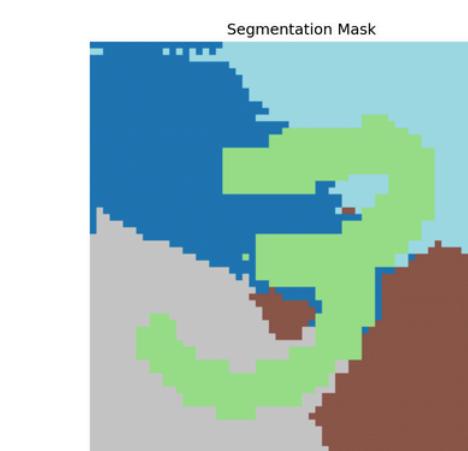
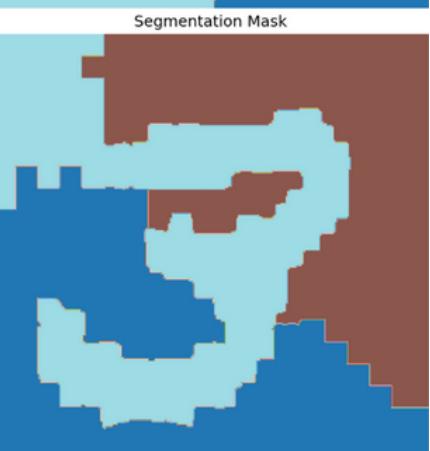
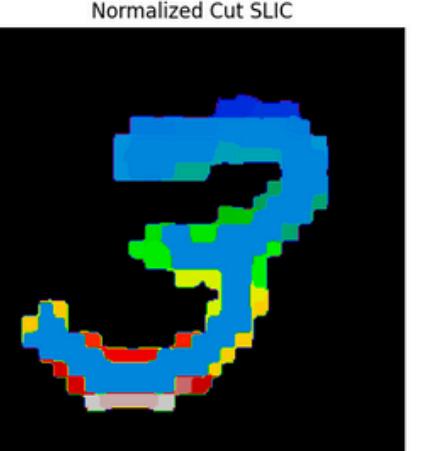
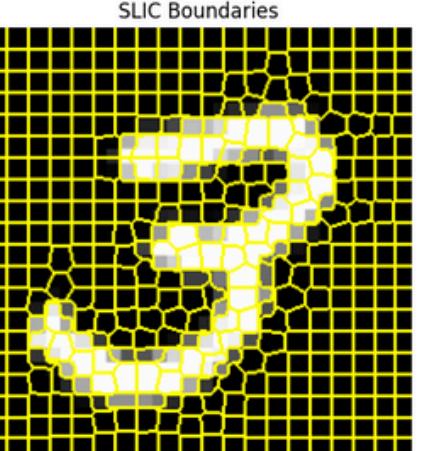
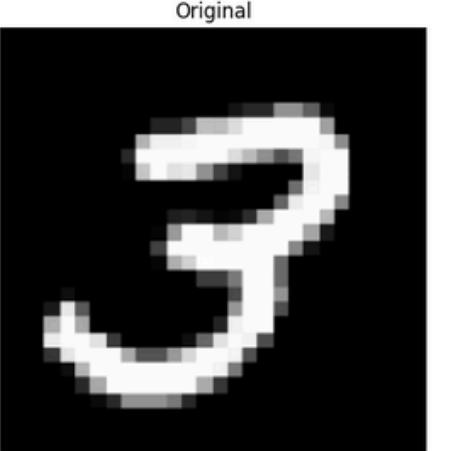
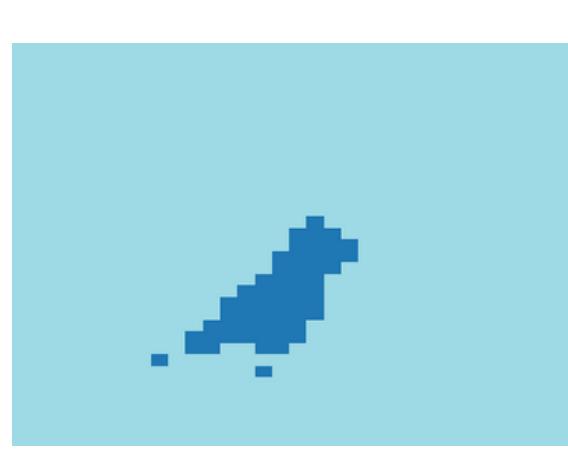
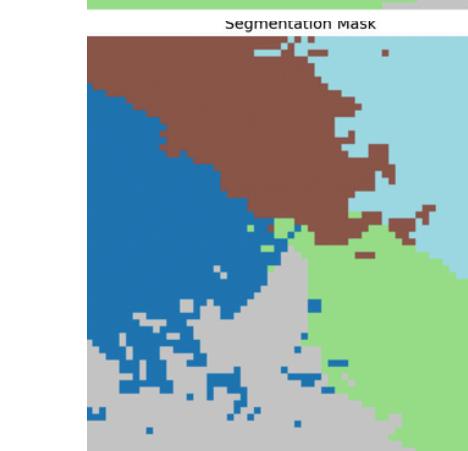
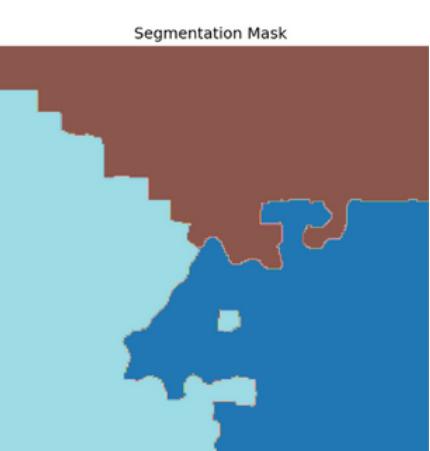
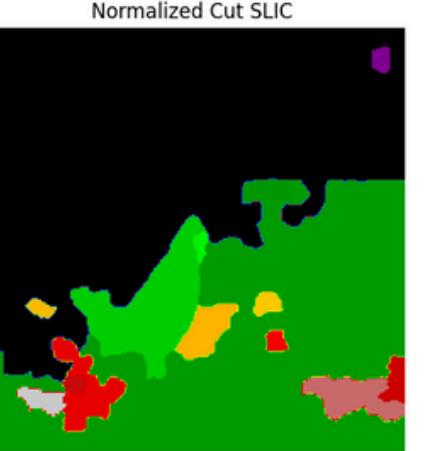
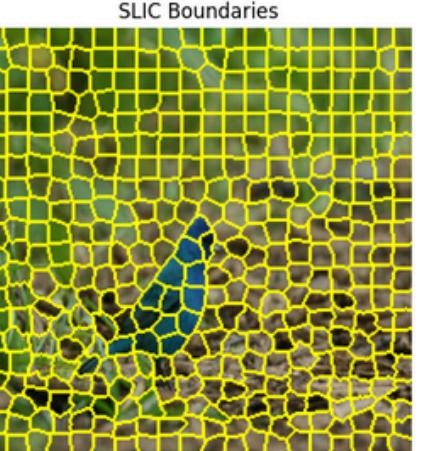
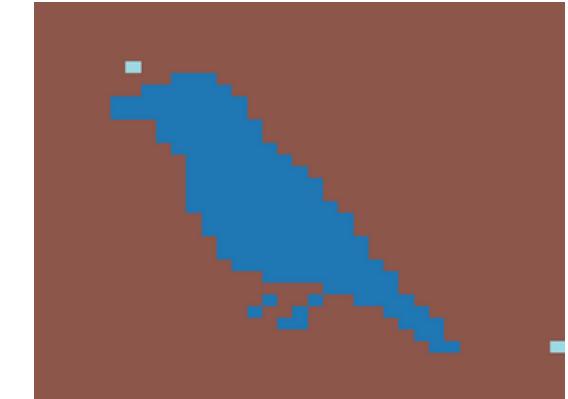
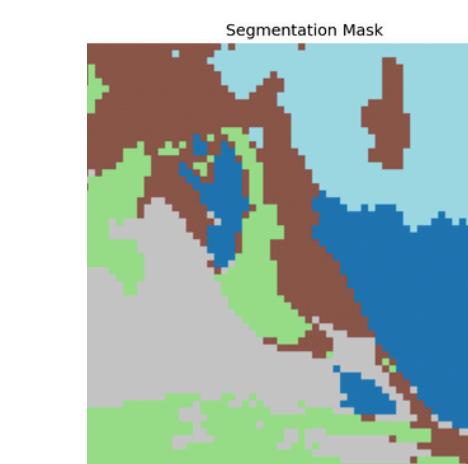
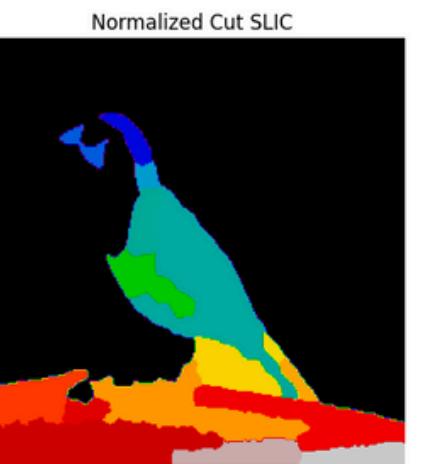
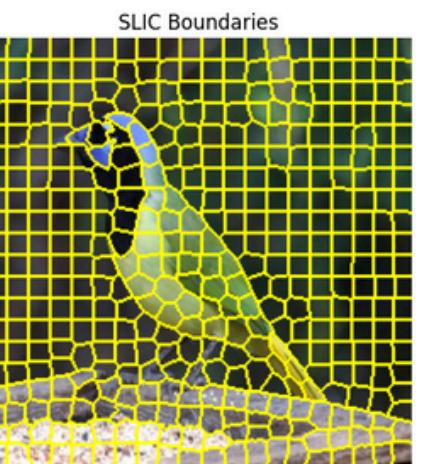
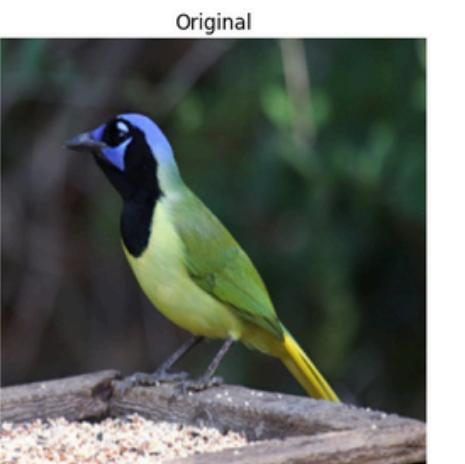


Image segmented in 400 super-pixels

size of the matrix = 400



Original

Super-pixels

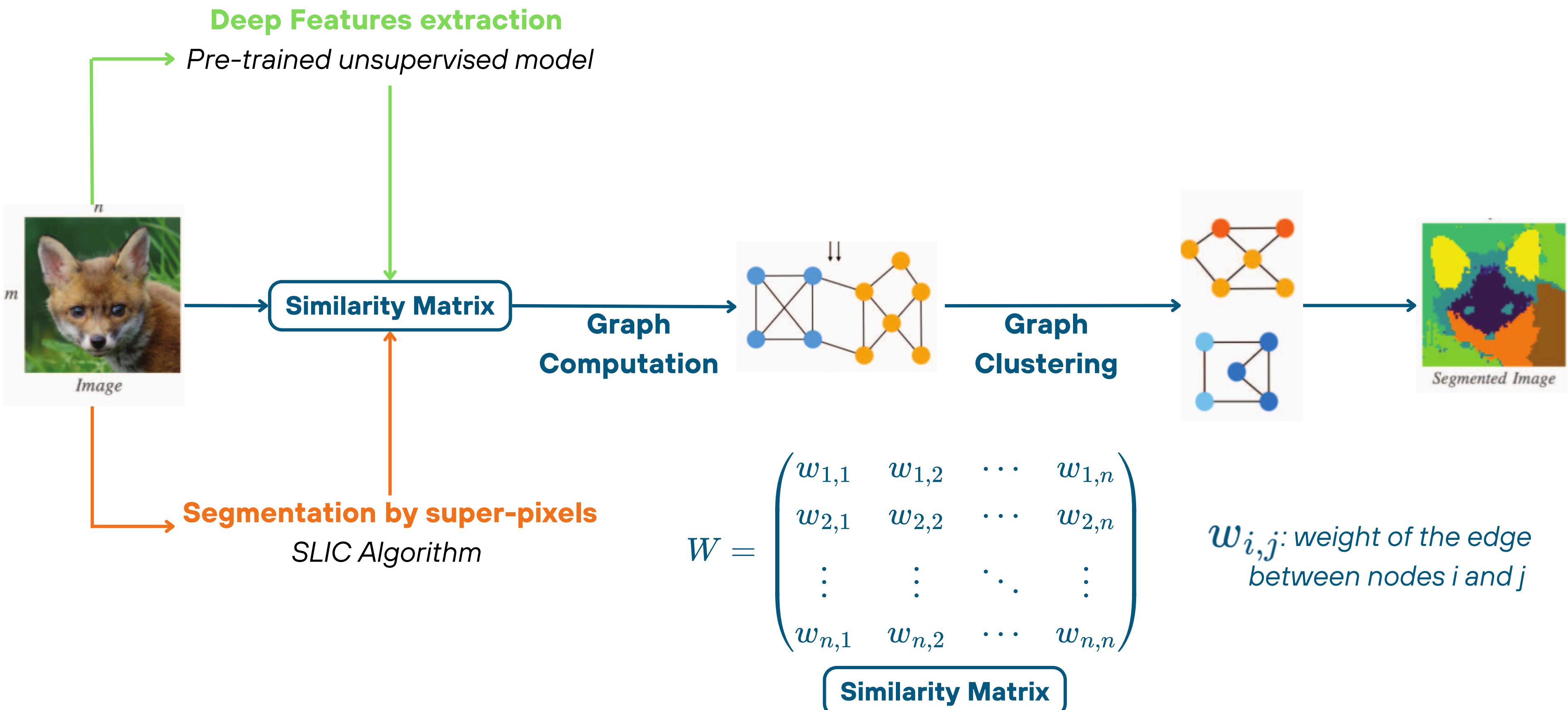
N-cut
results

GNN results with
super-pixels

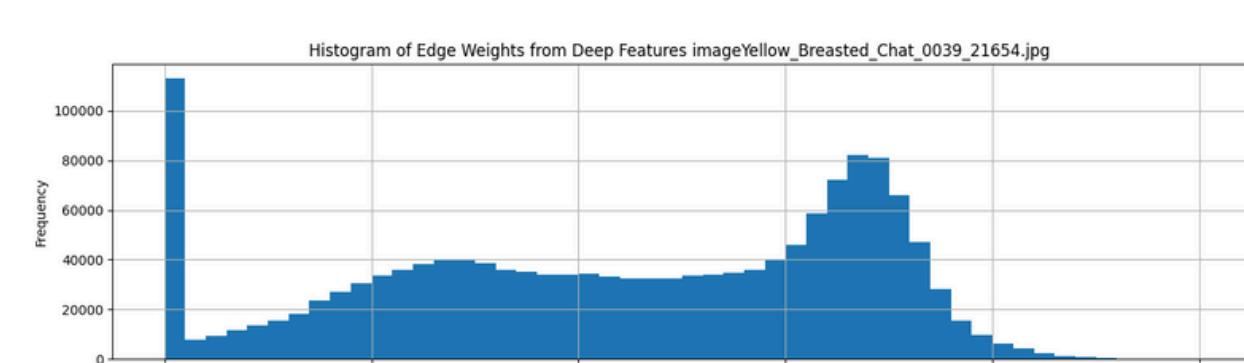
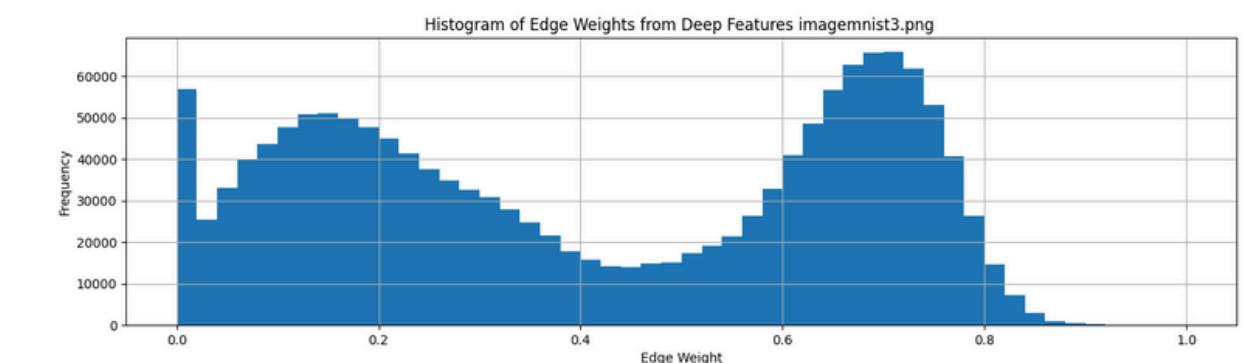
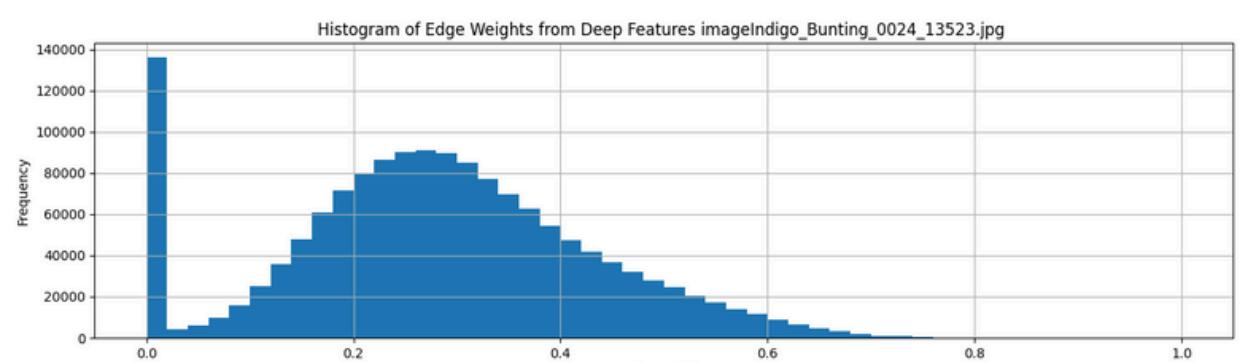
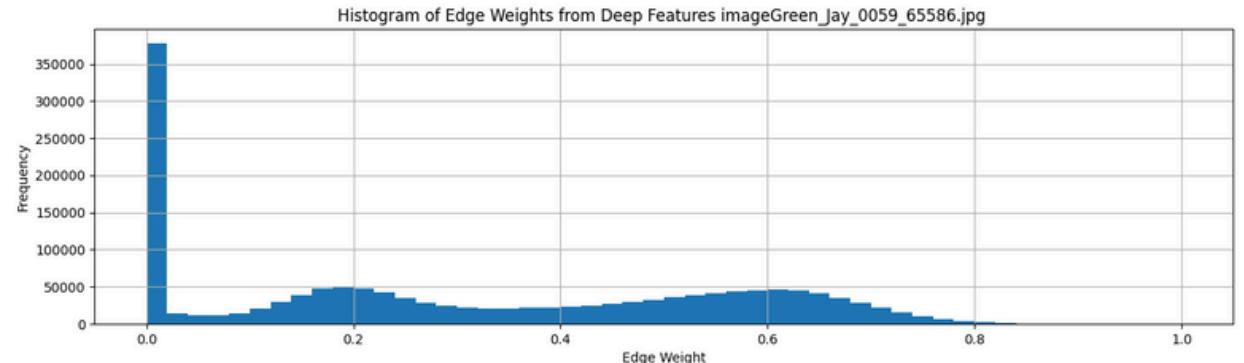
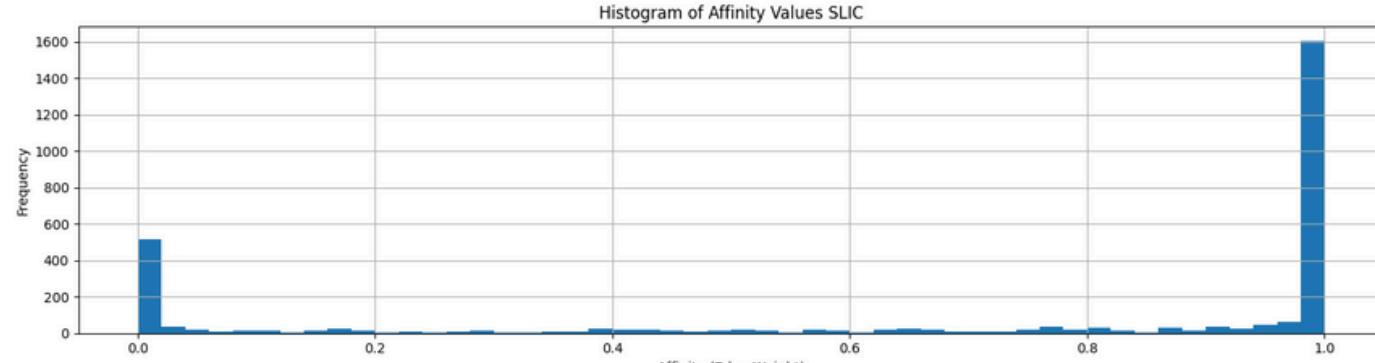
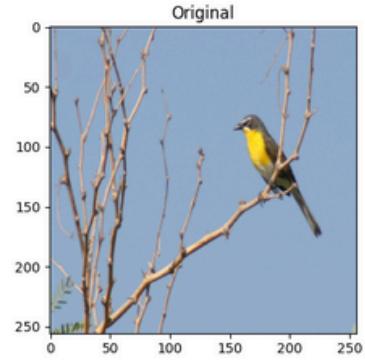
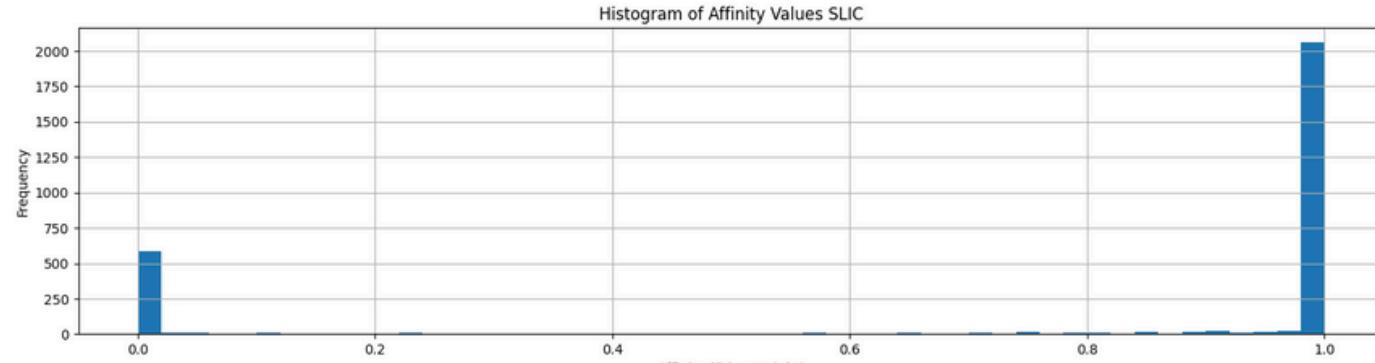
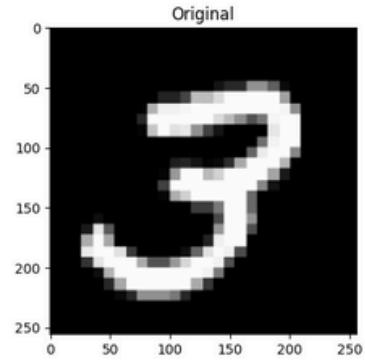
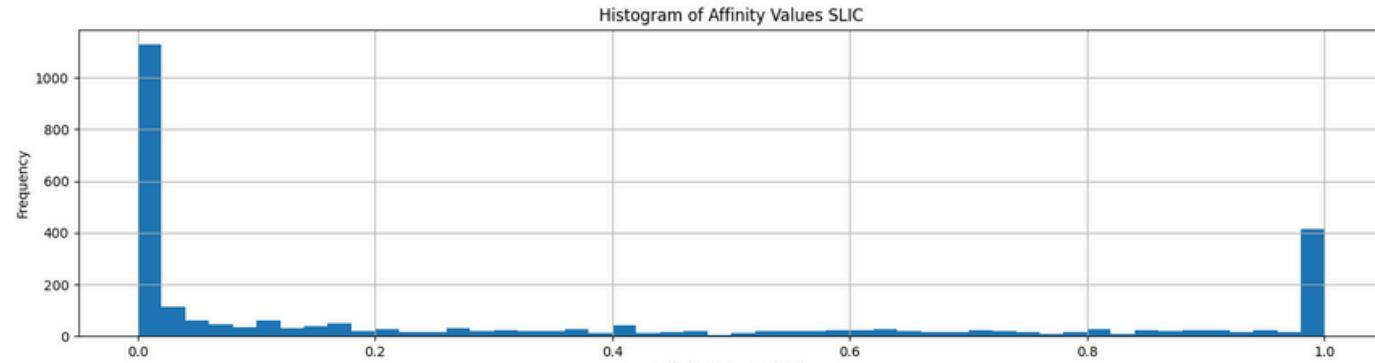
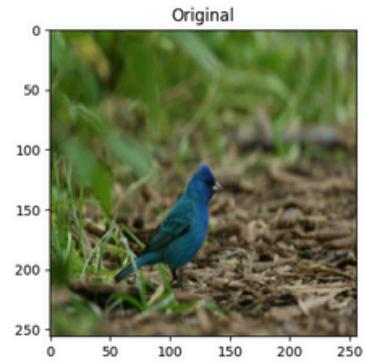
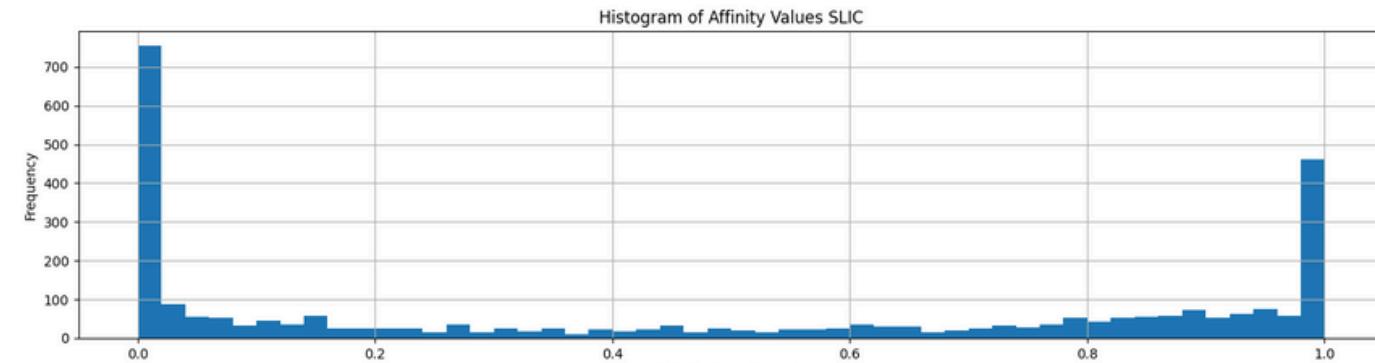
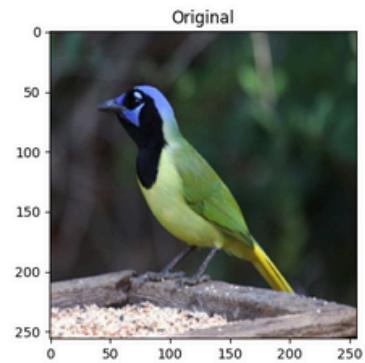
GNN results with
pixels

DeepCut results

6. Similarity matrix



6. Similarity Matrix Values



Affinity values using super-pixels

Affinity values using deep features

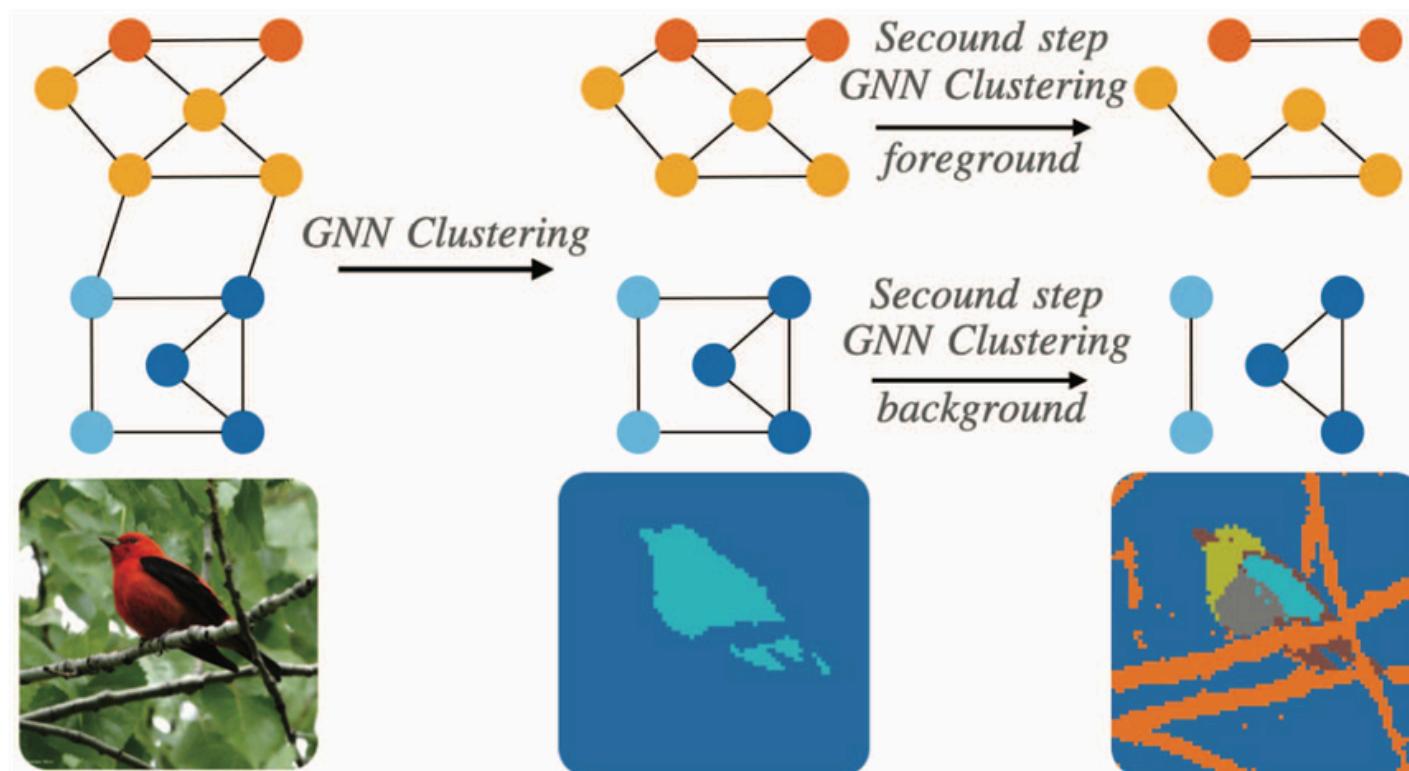
7. Improvements

Using correlation clustering loss (with negative affinities) :

- Positive affinities encourage patches to be in the same cluster
- Negative affinities push patches into separate clusters

Evaluate on a dataset to have numerical results

2 steps clustering



- Step 1 : foreground - background segmentation
- Step 2 : object part segmentation