Unsupervised clustering methods for image segmentation: application to scanning electron microscopy images of graphene

Aagam Shah, Darren Adams, Sameh Tawfick, Elif Ertekin

University of Illinois at Urbana-Champaign
Graphene: Microscopy Images
Image Segmentation in General

- Image segmentation is a way of separating an image into regions containing shared attributes.
- In our case, we will separate graphene from the substrate.
Automated Segmentation

- Goal: given an image, analyze each pixel to determine whether it corresponds to graphene or something else.
- Humans can usually recognize graphene after seeing one or two images (e.g. contrast, hexagonal edges) but quantifying many hundreds of images takes time.
- Automated segmentation can help identify important characteristics such as:
  - Percent area covered
  - Crystalline Quality (hexagons)
Today’s topics: two approaches to image segmentation

• Template matching
• K-means

Both of these require pre-processing the image in the same way.
Pre-processing

• Divide an image into windows
• Each window represents a vector of pixel intensities
• Plot the pixel intensities
• They generally form clusters
Pre-processing

• In reality, we have a 2D window of $n \times n$ pixels
• We flatten them out to make the vector

$\mathbf{v} = \langle p_1, \ldots, p_{n^2} \rangle$

Set of all intensity vectors
$I = \{ v_1, \ldots, v_m \}$
Template Matching

Idea: select area that looks like graphene and screen for similar looking areas

• Step 1: Select the “template”, flatten and vectorize it.
• Step 2: Plot it on the intensity vector plot
• Step 3: For all other parts of the image, measure how close they are to the template on the intensity vector plot
• Step 4: If the distance is within a threshold, classify as “graphene”. If not, then “not graphene”.

Parameters:
• Template position
• Template size
• Threshold (or distance)
K-Means: Pre-processing

• Recall preprocessing: We divide the image into windows, flatten them to make pixel intensity vectors and plot the vectors on a high-dimensional graph.

• In k-means, we also control the number of pixels moved between two tiles (stride length).

\[
\begin{align*}
I & = \{v_1, \ldots, v_m\} \\
v & = \langle p_1, \ldots, p_{n^2} \rangle \\
\text{Flatten} \\
\end{align*}
\]
K-Means (unsupervised clustering method)

• Main idea: divide the pixels into clusters by partitioning the pixel intensity plot using Voronoi polyhedra

• Algorithm
  • Start with map of pixel intensities from before
  • Initialize centroids
  • Step 1: construct Voronoi polyhedral around centroids
  • Step 2: calculate new centroids by averaging all points within a centroid’s Voronoi polyhedron
  • Repeat steps 1 and 2 until polyhedral are optimally selected.
  • For each cluster, assign it a label: graphene or not graphene
K-Means (unsupervised clustering method)

- Main idea: divide the pixels into clusters by partitioning the pixel intensity plot using Voronoi polyhedra

- Algorithm
  - Start with map of pixel intensities from before
  - Initialize centroids
  - Step 1: construct Voronoi polyhedral around centroids
  - Step 2: calculate new centroids by averaging all points within a centroid’s Voronoi polyhedron
  - Repeat steps 1 and 2 until polyhedral are optimally selected.
  - For each cluster, assign it a label: graphene or not graphene
K-Means (unsupervised clustering method)

• Algorithm
  • Initialize centroids
  • Step 1: construct Voronoi polyhedral around centroids
  • Step 2: calculate new centroids by averaging all points within a centroid’s Voronoi polyhedron
  • Repeat steps 1,2.
  • Assign labels to the clusters

Parameters:
• Number of clusters
• Tile size
• Stride length
K-Means (unsupervised clustering method)

- **Advantages:**
  - No need to select template or threshold
  - Fast, memory efficient

- **Drawbacks:**
  - Need to select number of centroids (clusters)
  - Can suffer from concave shaped blobs
SEM Image Processing Tool

Visit https://nanohub.org/tools/gsaimage

Or

Look for “SEM Image Processing Tool” on nanoHUB.
Exercise

Open the following notebooks on Google Colab:

https://github.com/ertekin-research-group/image-segment/blob/master/bin/Template_Matching.ipynb

https://github.com/ertekin-research-group/image-segment/blob/master/bin/K-Means.ipynb

!git clone https://github.com/nanoMFG/nanohub_workshop_2021.git