SEM IMAGE SEGMENTATION USING MACHINE LEARNING

Project Report

Submitted in partial fulfillment of the requirement

For the award of the degree of

Bachelor of Technology in Electronics Engineering

By

Maha Zakir Khan

Shaista Tabrez

Pratima Singh

Supervisor

Prof. Mohammad Jawaid Siddiqui

DEPARTMENT OF ELECTRONICS ENGINEERING

ZAKIR HUSAIN COLLEGE OF ENGINEERING & TECHNOLOGY

ALIGARH MUSLIM UNIVERSITY ALIGARH

ALIGARH-202002 (INDIA)

JUNE, 2020

STUDENTS' DECLARATION

I hereby certify that the work which is being presented in this project report entitled "SEM Image Segmentation using Machine Learning" in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology and submitted in the Department of Electronics Engineering of the Zakir Husain College of Engineering & Technology, Aligarh Muslim University, Aligarh is an authentic record of my own work carried out during 3rd year of B. Tech. under the guidance of Prof. Mohammad Jawaid Siddiqui, Professor Department of Electronics Engineering, Aligarh Muslim University, Aligarh.

Maha Khan   Shaista Tabrez   Pratima Singh

Maha Zakir Khan   Shaista Tabrez   Pratima Singh

This is to certify that the above statement made by students is correct to the best of my knowledge.

(Prof. Mohammad Jawaid Siddiqui)

Project Guide

Date: 14: 04: 2022

Abstract:

In digital image processing and computer vision, image segmentation refers to the process of partitioning a digital image into multiple segments or related sets of pixels.

Research in image processing and segmentation techniques for SEM images is ongoing and a hot topic of interest for the researchers. Enormous efforts have been made to replace the human analysis of digital images by automated computational methods such as image processing and computer vision. The image processing methods (i.e., pixel and object-based assessments) determine a variety of spectral/optical features and devices/data properties on the images so that appropriate information could be extracted

However, these methods may not be appropriate when globally heterogeneous and locally anisotropic features exist such as those found in Scanning Electron Microscopy (SEM) Images. Thus, it is essential to have an adaptive and data-driven procedure to extract optimal information from individual SEM images

This project will introduce some basic image segmentation techniques driven by unsupervised machine learning techniques such as the Gaussian mixture model, k-means clustering, and template matching, and use them to segment a scanning electron microscopy image of graphene on a substrate. This paper gives brief overview of graphene, SEM images, Machine Learning, Neural Networks, introduces U-Net, a popular convolutional neural network commonly developed for image segmentation in biomedicine.

Acknowledgement

We wish to express our deepest gratitude and indebtedness to our mentor Prof. Mohammad Jawaid Siddiqui for introducing us to this topic and for his constant guidance throughout the project. He has always given us his valuable suggestion which helps us in timely completion of our project. He has always been cordial, attentive, responsible and supportive throughout all the highs and lows during the journey of our mini project. He has devoted a lot of his time for our work and took great pain to see us through. We have learnt a lot from him.

Table of Contents

Content

List of figures

List of Tables

List of Symbols

Π: pi

μ: Mean

Σ: Covariance

List of Abbreviations

SEM: Segmenting microscopy images

ANN: Artificial Neural Network

FCN: Fully Convolutional Network

CNN: Convolutional Neutral Network

U-net: U-Network

GMM: Gaussian Mixture Model

E-M: Expectation Maximization

GANs: Generative adversarial networks

Chapter 1: Introduction to Deep Learning

1.1: Introduction

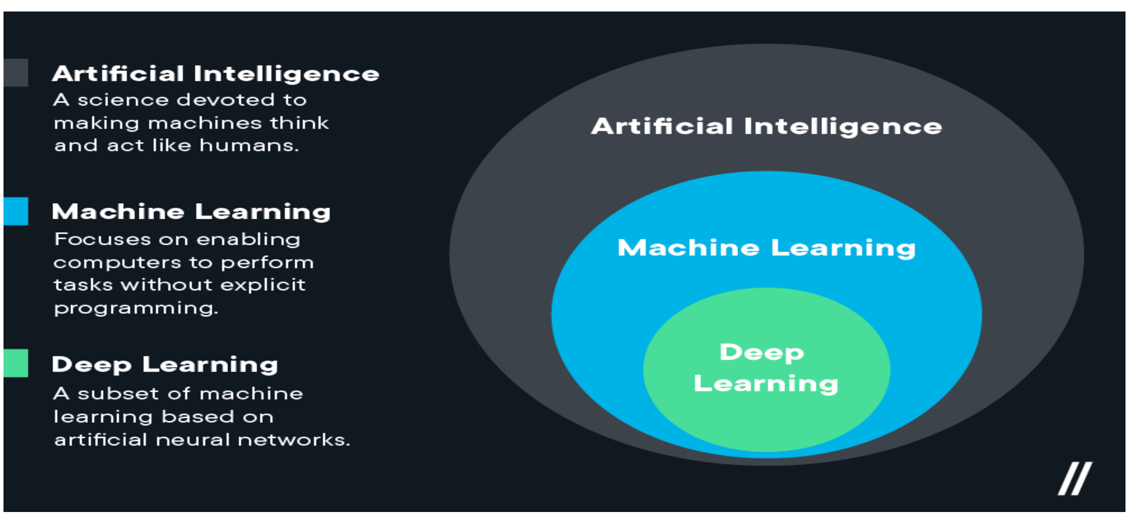
We have all heard about the term Artificial intelligence, but artificial intelligence is only a broader term that describes applications when a machine mimics "cognitive" functions that humans associate with other human minds, such as "learning" and "problem-solving".

1.2: Machine Learning

Machine Learning incorporates "classical" algorithms for various kinds of tasks such as clustering, regression or classification. Machine Learning algorithms must be trained on data. The more data you provide to your algorithm, the better it gets. The "training" part of a Machine Learning model means that this model tries to optimize along a certain dimension. In other words, the Machine Learning models try to minimize the error between their predictions and the actual ground truth values.

1.3: Deep Learning

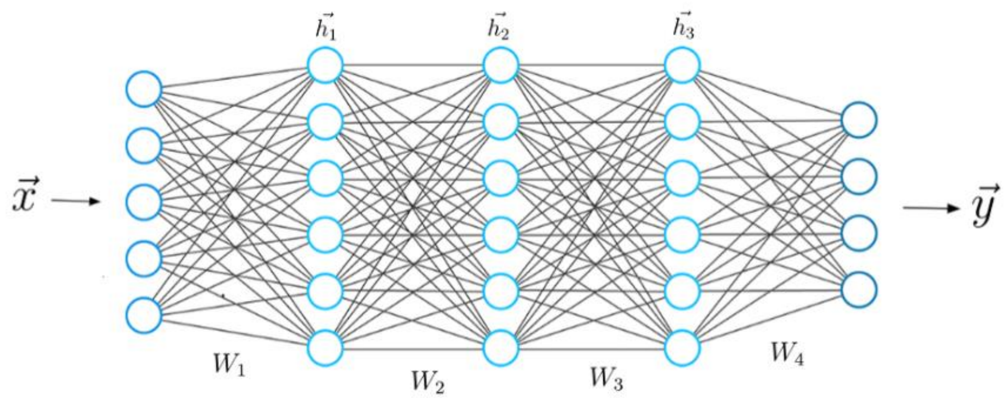
Deep Learning is a very young field of artificial intelligence based on artificial neural networks. It can be viewed again as a subfield of Machine Learning since Deep Learning algorithms also require data in order to learn to solve tasks. Therefore, the terms of machine learning and deep learning are often treated as the same. However, these systems have different capabilities.



[Source: https://flatironschool.com/blog/deep-learning-vs-machine-learning/]

Fig 1.1 Relationship between Artificial Intelligence, Machine Learning and Deep Learning

Deep Learning uses a multi-layered structure of algorithms called the neural network:



Neural Network, Source: www.deeplearning-academy.com/ai-wiki

Fig 1.2

Artificial neural networks (ANN) have unique capabilities that enable Deep Learning models to solve tasks that Machine Learning models could never solve.

Chapter 2: Image segmentation

2.1: Introduction

Image Segmentation is a branch of digital image processing which focuses on partitioning an image into different parts according to their features and properties. It is the process by which a digital image is partitioned into various subgroups (of pixels) called Image Objects, which can reduce the complexity of the image, and thus analyzing the image becomes simpler.

2.2: Approaches for different segmentation techniques:

Now that we know the different approaches and kinds of techniques for image segmentation, we can start discussing the specifics. Following are the primary types of image segmentation techniques:

2.2.1: Thresholding Segmentation

2.2.1.1: Simple Thresholding

In this method, the image's pixels are replaced with either white or black. If the intensity of a pixel at a particular position is less than the threshold value, it is replaced with black and if it's higher than the threshold, it is replaced with white.

2.2.1.2: Adaptive Thresholding

In this method, threshold value is kept different for different sections of image. This method works well for images having different backgrounds and conditions which effect their properties. An algorithm is used that segments the image into smaller sections and calculates the threshold value for each of

2.2.2: Edge-Based Segmentation

Edge-based segmentation algorithms work to detect edges in an image, based on various discontinuities in grey level, color, texture, brightness, saturation, contrast etc. It's aim is to reach at least a partial segmentation, where all local edges are grouped into a new binary image where only edge chains that match the required existing objects or image parts are present.

2.2.3: Region-Based Segmentation

Region-based segmentation algorithms divide the image into sections with similar features. These regions are only a group of pixels and the algorithm find these groups by first locating a seed point which could be a small section or a large portion of the input image. After finding the seed points, a region-based segmentation algorithm would either add more pixels to them or shrink them so it can merge them with other seed.

2.2.4: Watershed Segmentation

The watershed methods consider the gradient of image as topographic surface. The pixels having more gradient are represented as boundaries which are continuous.

2.2.5: Clustering-Based Segmentation Algorithms

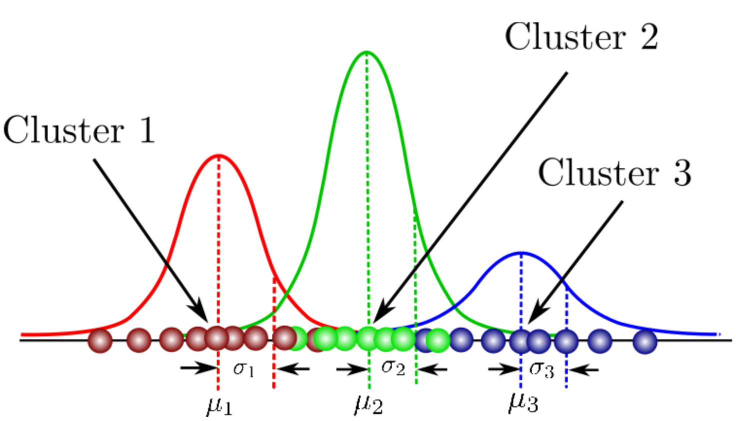
Suppose we have N number of Unlabeled Multivariate Datasets of various Animals like Dogs, Cats, birds etc. The technique to segregate Datasets into various groups, on basis of having similar features and characteristics, is being called Clustering. The groups being formed are being known as Clusters. Clustering Technique is being used in various Field such as Image recognition, Spam Filtering, etc.

2.2.5.1: K means

It follows a simple procedure of classifying a given data set into a number of clusters, defined by the letter "k" which is fixed beforehand.

2.2.5.2: Gaussian Mixture Model

Gaussian mixture model involves the mixture (i.e. superposition) of multiple Gaussian distributions, meaning values are more likely around mean over extremes. Every distribution is multiplied by a weight π to account for the fact that we do not have an equal number of samples from each category. Since, we're dealing with probabilities, the weights should have a sum of 1 after addition.



[Source: https://towardsdatascience.com]

Fig 2.1

The number of clusters specifies the number of components in the GMM. Each Gaussian in the mixture carries some parameters which are-

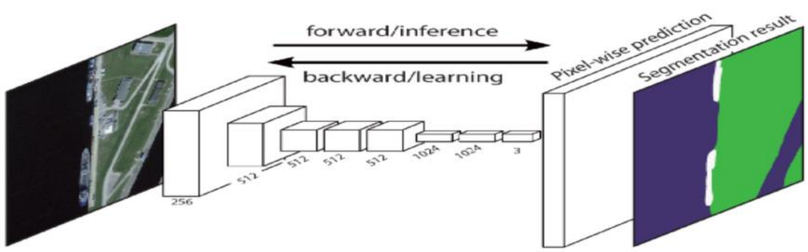
1. A mean μ that defines its center.

2. A covariance Σ that defines its width. This would be equivalent to the dimensions of an ellipsoid in a multivariate scenario.
3. A mixing probability π that defines how big or small the Gaussian function will be.

2.2.6: Neural Networks for Segmentation

2.2.6.1: Fully Convolutional Network (FCN):

The original Fully Convolutional Network (FCN) learns a mapping from pixels to pixels, without extracting the region proposals. The FCN network pipeline is an extension of the classical CNN. The main idea is to make the classical CNN take as input arbitrary-sized images. FCNs only have convolutional and pooling layers which give them the ability to make predictions on arbitrary-sized inputs.



[Source: https://towardsdatascience.com]

Fig 2.9 FCN Architecture

One issue in this specific FCN is that by propagating through several alternated convolutional and pooling layers, the resolution of the output feature maps is down sampled. Therefore, the direct predictions of FCN are typically in low resolution, resulting in relatively fuzzy object boundaries.

2.2.6.2: U-Net:

The U-Net was developed by Olaf Ronneberger et al. for Bio Medical Image Segmentation.

Some other architectures: AlexNet and ResNet.

2.6: Applications of image segmentation

1. Medical Imaging
2. Traffic Control Systems
3. Self-Driving Cars
4. Locating objects in satellite image

Chapter 3: Machine Learning for SEM Image Segmentation in Materials Science

3.1: Introduction

A SEM image is formed by beam of electrons focused to a few billionths of meter that is swept across a surface of a sample in a series of stacked rows until 2D image is formed. Scanning electron microscopy (SEM) images are typically used to observe the growth results of a synthesis experiment, such as areal coverage, nucleation density, shape, quality of graphene domains. While the visual inspection of images can sometimes be sufficient to determine the quality of graphene, it is desirable to determine quantitative metrics as well. Quantitative metrics can provide easier comparison between experimental results and are useful as response variables when attempting to predict optimal recipes. To calculate these metrics, we need to segment the image and each pixel needs to be classified as 'graphene' or 'not-graphene'.

Note: Graphene is a single layer (monolayer) of carbon atoms, tightly bound in a hexagonal honeycomb lattice. It is an allotrope of carbon in the form of a plane of sp2-bonded atoms. It is monoatomic in thickness and has small particle sizes and an extremely high surface area; these characteristics have exceptional outcomes on its properties.

3.2: Using Template Matching to segment SEM Image of graphene

Template matching is a method in which we search similar template in our source image by sliding template image from left to right and top to bottom over the source image and where ever in the source image similar template are found a rectangular box is drawn.

3.2.1: Python implementation of Template Matching:

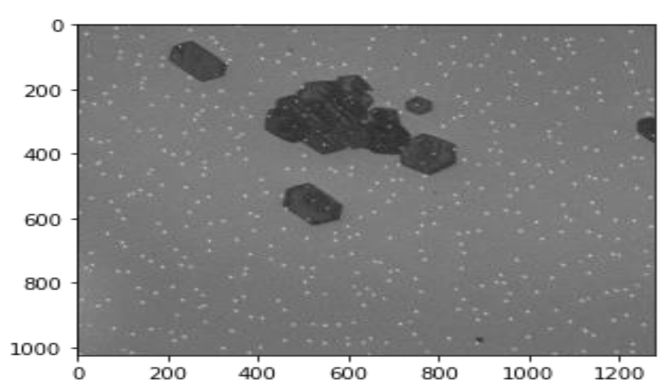


Fig 3.1 SEM image of graphene

[The SEM data-set of graphene are taken from the site:
https://proxy.nanohub.org/weber/2016635/1eek9Jh5fcnS3011/2/tree/data]

Rectangular boxes shows where the matching template similar is found in the source image.

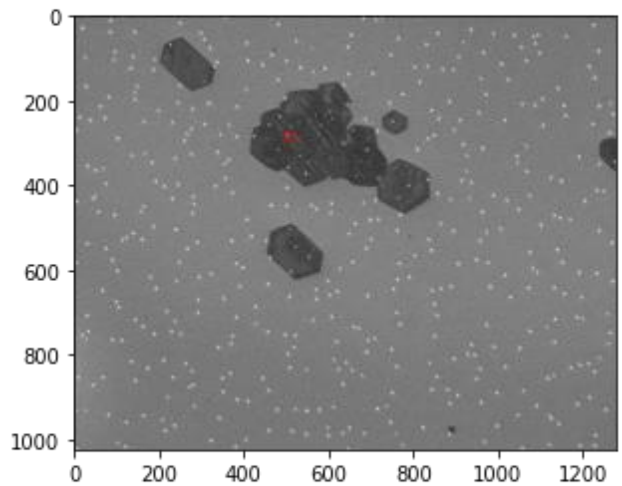
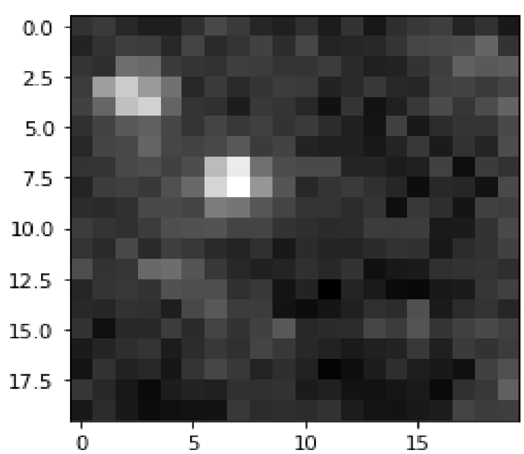


Fig 3.2 The small red box is the template used

Using resolutions of the template that is width and height of the template we will find matching template in the source image.



3.3: Zoomed in view of the template

All the bright spots in the image shows best matches of the matching template found in the source image whereas the darker region shows the area where the template does not match very well with the source image.

Now we can match the template to each tile in the image

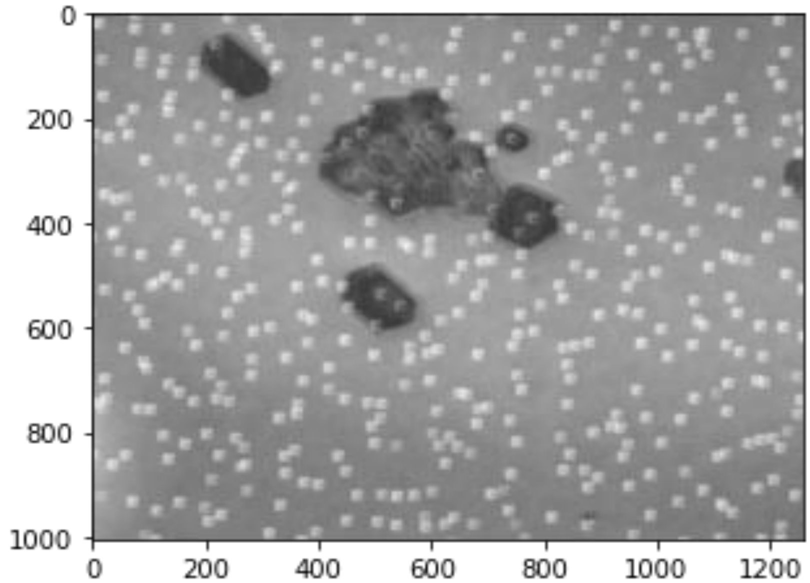


Fig 3.4 Image showing all similar template in SEM image

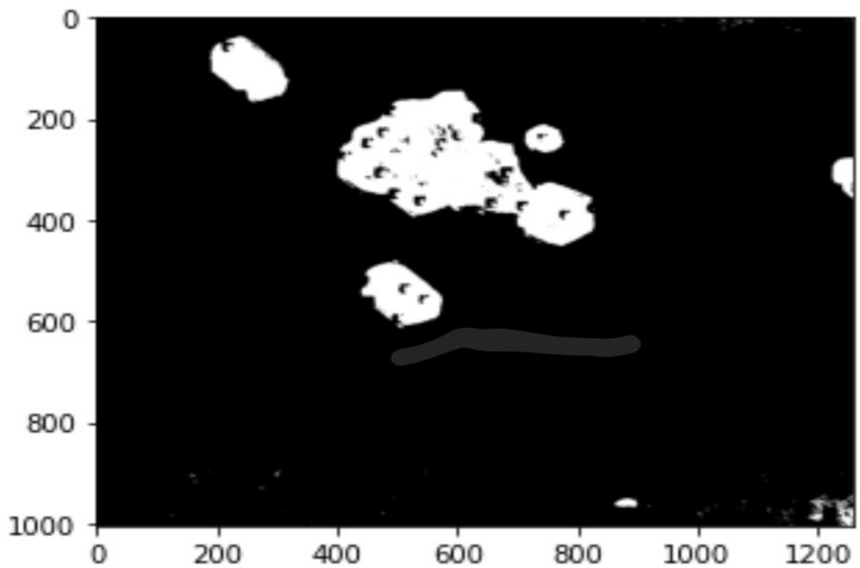


Fig 3.5 Masked Image

3.2.2: Effect of parameter variation:

3.2.2.1: Varying the template size:

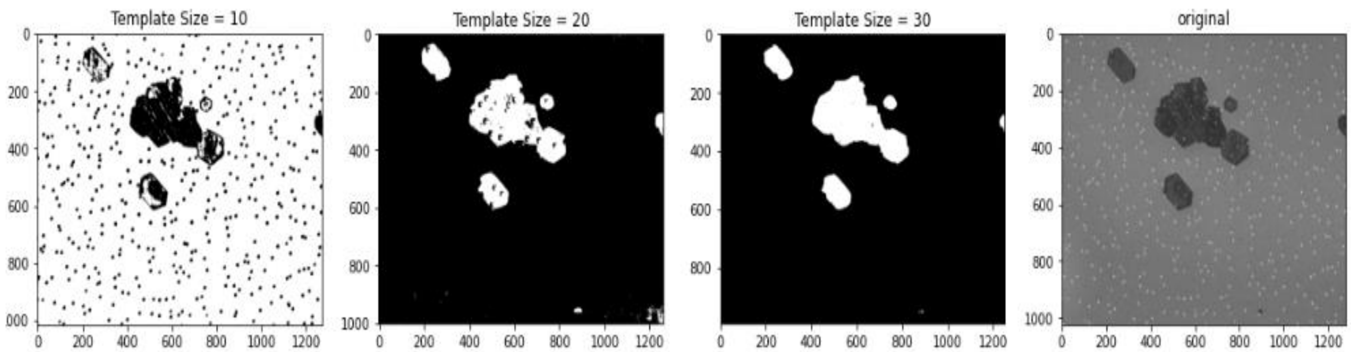


Fig 3.6 Template matching with different template size

3.2.2.2: Varying the threshold:

We make use of thresholding to find matching templates. Thresholding basically means probability of finding a best match of our image in the source image Threshold value changes with different images.

Difference between segmented images having threshold value 0.2, 0.25 and 0.30

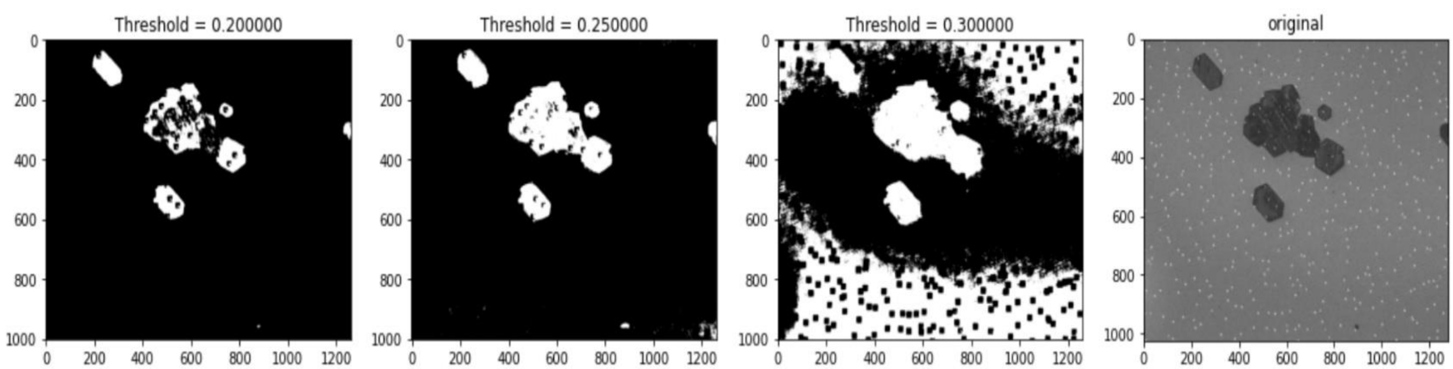


Fig 3.7 Template matching with different threshold value

3.2.3: Uses:

Template matching can be used as a pipeline in conducting object detection for machine learning models and deep learning models. It can be used in face recognition and eye detection in facial image. It is used in biological science such as Nuclear Agriculture and Molecular Biology.

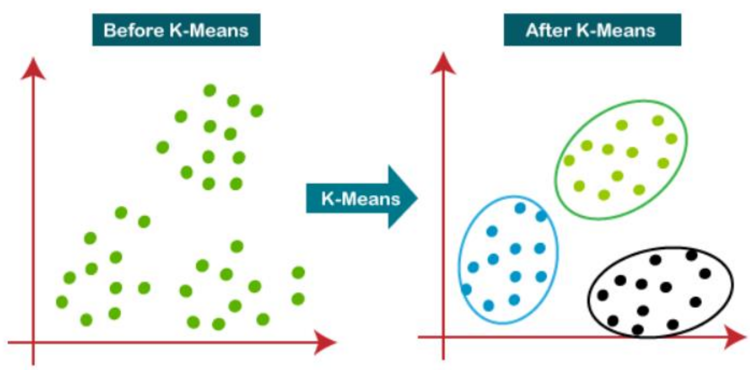
3.2.4: Limitations of Template Matching:

1. Templates are not rotation or scale invariant.

2. Slight change in size or orientation variations can cause problems.
3. It often uses several templates to represent one object.
4. Template Matching requires high computational power because the detection of large patterns of an image is very time taking.

3.3: Using K-Means clustering to segment microscopy images

We will use a type of Stochastic Segmentation Technique known as K-means. K-Means is a relatively quick and memory efficient method to cluster images. There is no need to select any template or threshold. All data points are partitioned into k number of clusters, each of which is represented by its centroids (prototype). The centroid of a cluster is often a mean of all data points in that cluster. k-means is a partitioning clustering algorithm and works well with spherical-shaped clusters



[Source: www.analyticsvidhya.com/blog/2021/04/k-means-clustering-simplified-in-python]

Fig 3.8 Kmeans Clustering

3.3.1: The working of the K-Means algorithm:

1. Step-1: Select the value of K, to decide the number of clusters to be formed.
2. Step-2: Select random K points which will act as centroids.
3. Step-3: Assign each data point, based on their distance from the randomly selected points (Centroid), to the nearest/closest centroid which will form the predefined clusters.
4. Step-4: place a new centroid of each cluster.
5. Step-5: Repeat step no.3, which reassign each datapoint to the new closest centroid of each cluster.
6. Step-6: If any reassignment occurs, then go to step-4 else go to Step 7.
7. Step-7: FINISH

3.3.2: Python implementation of k-means

Initializing a randomly generated dataset with roughly four centers for testing

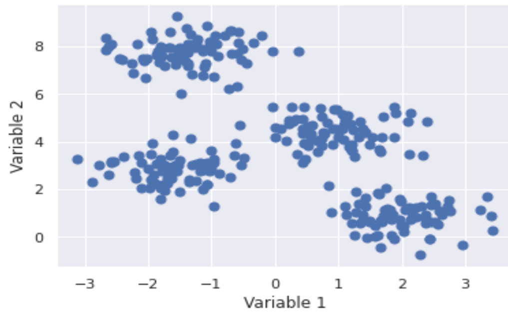


Fig 3.9 Plot of data points

Grouping individual points according to the number of clusters k

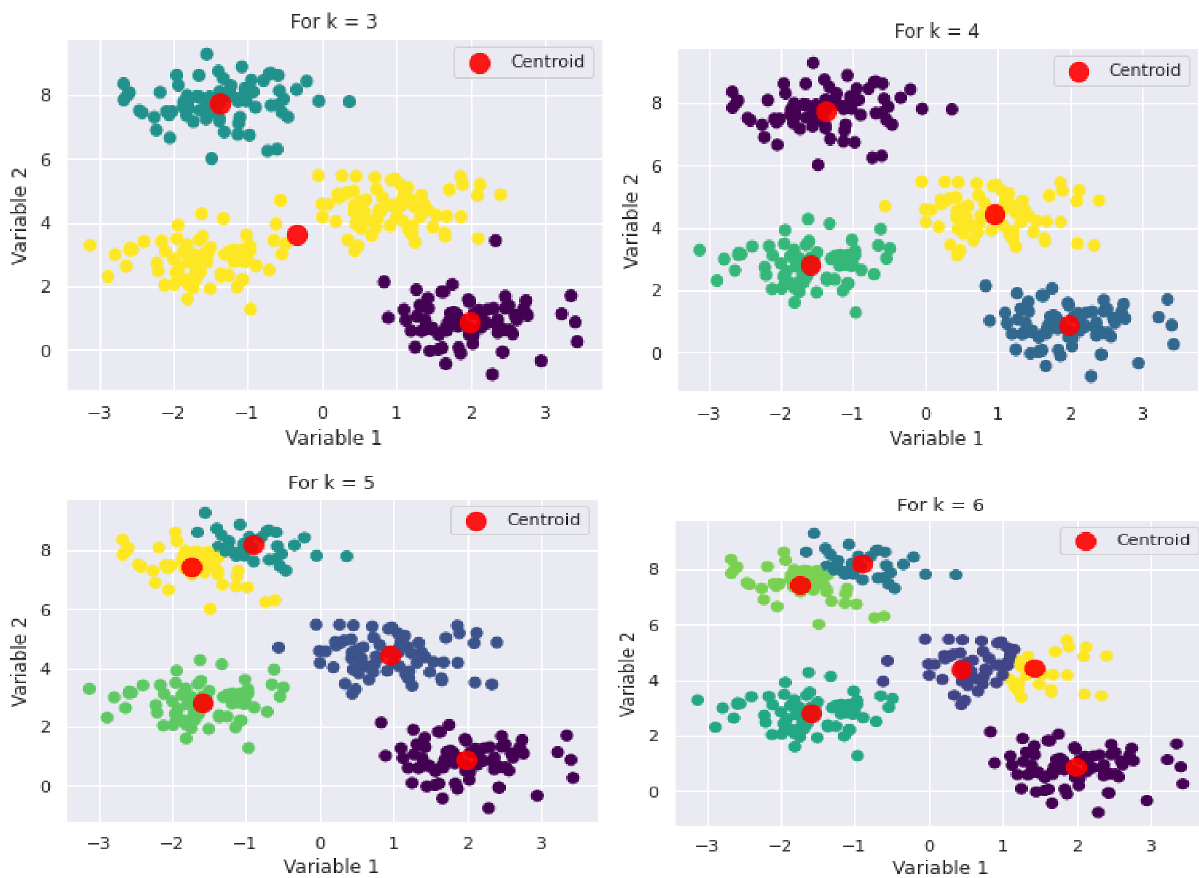
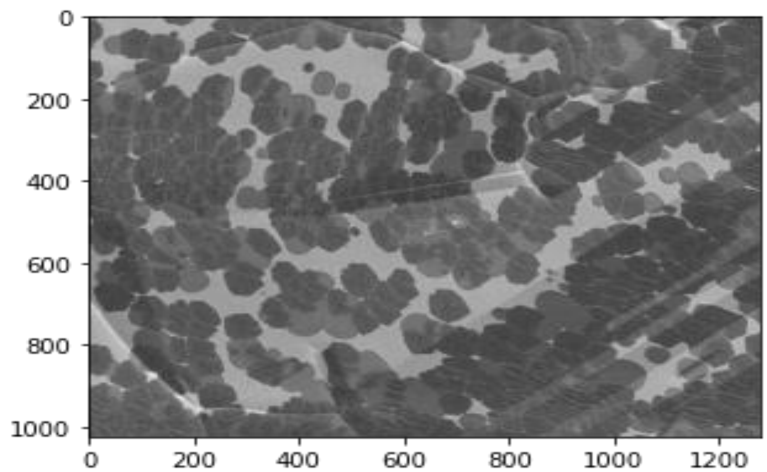


Fig 3.10 Plot of clustered data points (for k =3, 4, 5 and 6)

Let's perform k-mean clustering on the SEM image of graphene over a copper layer shown below:

We will use Kmeans and a variant known as MiniBatchKmeans to cluster our images

The MiniBatchKMeans is a variant of the KMeans algorithm which uses mini-batches to reduce the computation time, while still attempting to optimize the same objective/cost function. Mini-batches are subsets of the input data, randomly sampled in each training iteration. These mini-batches drastically reduce the amount of computation required to converge to a local solution. In contrast to other algorithms that reduce the convergence time of k-means, mini-batch k-means produces results that are generally only slightly worse than the standard algorithm.



Resolution of Image: (1024, 1280)

Fig 3.11 SEM image of graphene over copper substrate

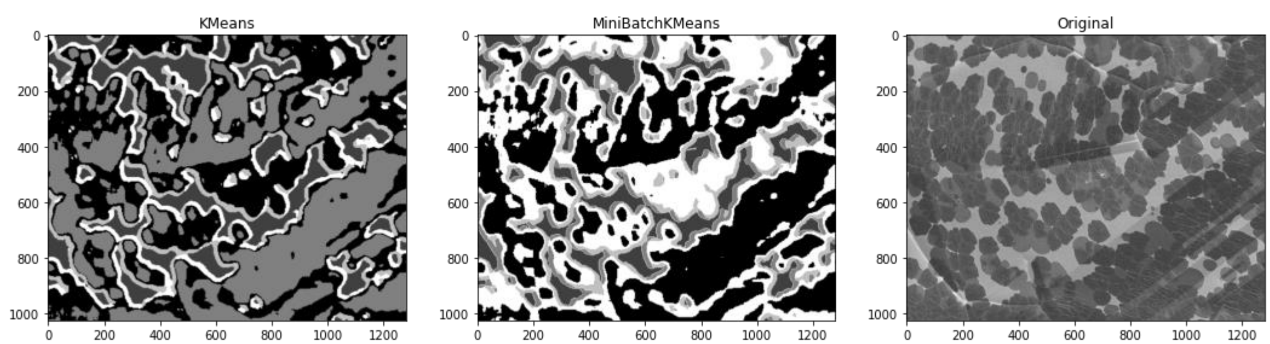


Fig 3.12 Segmented Images after performing clustering

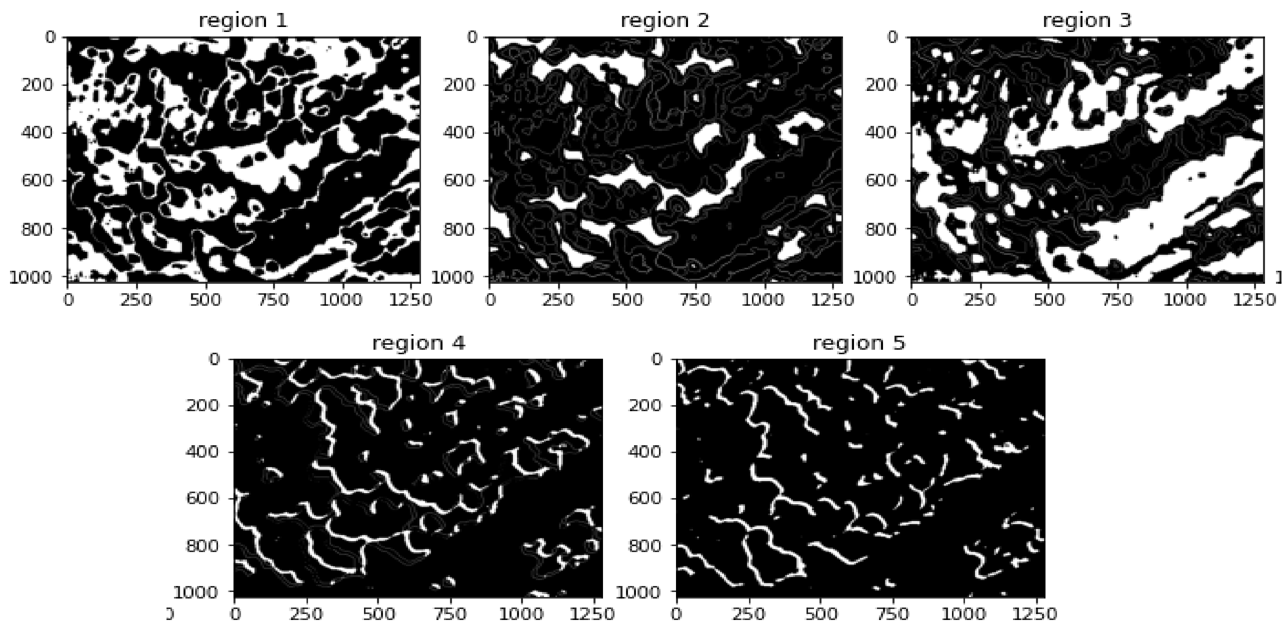


Fig 3.13 Segmented Images with each region shown separately

3.3.3: Effect of parameter variation:

3.3.3.1: Varying number of clusters:

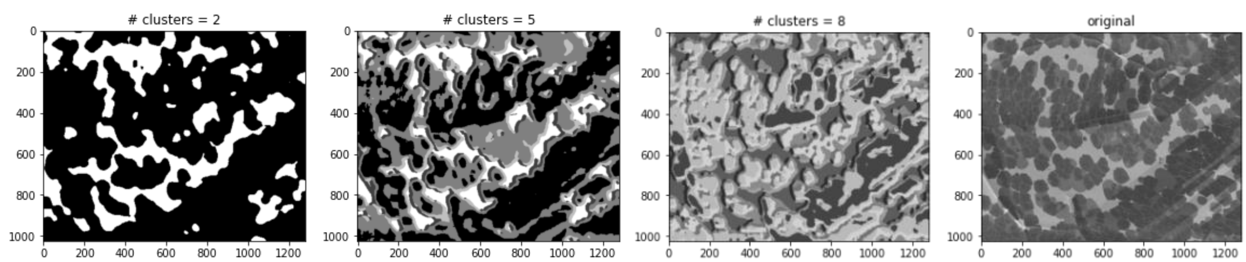


Fig 3.14 k-means with different cluster size

3.3.3.2: Varying stride length:

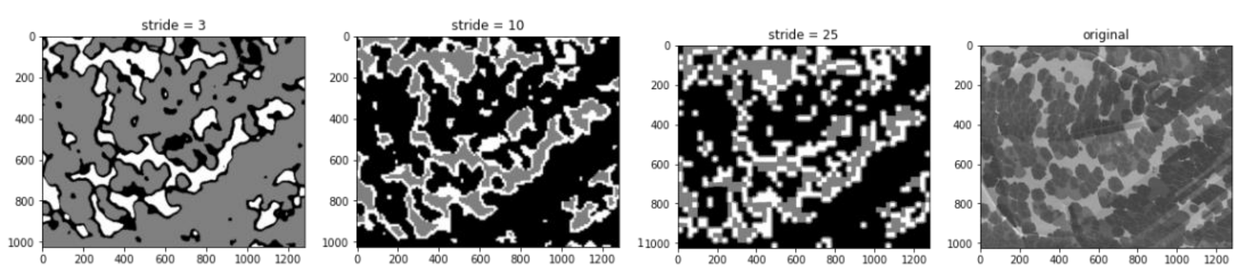


Fig 3.15 k-means with different stride length

3.3.3.3: Varying window size:

See the difference between having a window side length of 10, 30 and 50 pixels. Note the differences in the time taken to execute the segmentation.

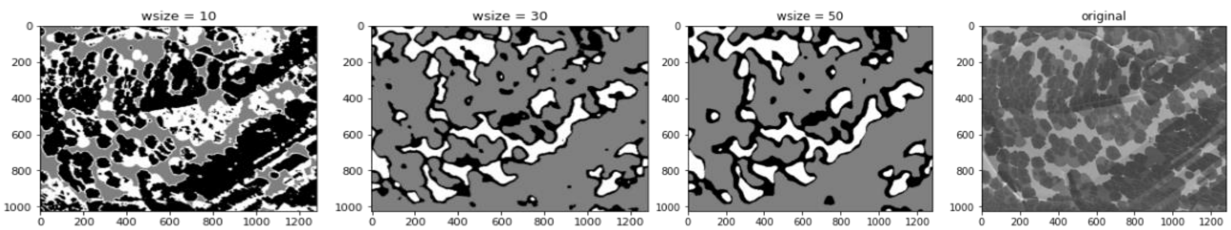


Fig 3.16 k-means with different window size

3.3.4: Accuracy of k-means clustering

As k-means is a clustering method (not classification), the accuracy should not be evaluated. This is because we do not train the model with class label data and therefore there will be inconsistency in between true class labels and predicted class labels. You may compare the scatterplot of original dataset and scatterplot after k-means clustering for evaluating the performance of k-means clustering.

3.3.5: Limitations of k-means clustering

1. In k-means clustering, it is essential to provide the number of clusters to form from the data. If the dataset is well-separated, it would be easy to identify the optimal number of clusters using the elbow method. But, if the data is not well-separated, it would be difficult to find the optimal number of clusters.
2. k-means clustering works well with spherical-shaped clusters of similar sizes. If there are arbitrary-shaped clusters, k-means may not work well.
3. The k-means can produce empty clusters (no points assigned to the cluster) depending on the initialization method and the value of k. In this case, you should try clustering with different values of k.

3.4: Using Gaussian-mixture clustering to segment microscopy images

3.4.1: Why Gaussian Mixture Model clustering technique is better than k-means clustering?

The limitations of k means as stated above can be addressed in GMM as it has these two important characteristics:

1. It measures the uncertainty in cluster assignment rather than just focusing on which one is closest.
2. It takes into account non-circular clusters as well.

3.4.2: Algorithm of Gaussian Mixture Model clustering technique

Step-1: Let's generate some sample data (Fig 3.17)

Step-2: The probabilities that a data point belongs to each of the K clusters. (Fig 3.18)

Step-3: The probabilistic clusters obtained after GMM clustering (Fig 3.19)

Step-4: We can visualize the uncertainty by, making the size of each point proportional to the certainty of its prediction (Fig 3.20)

Step-5: Let us visualize the locations and shapes of the GMM clusters by drawing spheres based on the GMM output. (Fig 3.21)

Step-6: Similarly, we can use the GMM approach to fit our stretched dataset; allowing for a full covariance the model will fit even very oblong, stretched-out clusters (Fig 3.22)

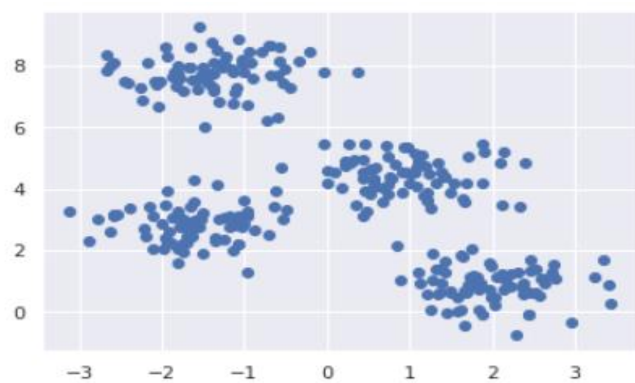


Fig 3.17 Plot of data points

```
[[0.972 0.026 0.    0.002]
 [0.    0.    1.    0.   ]
 [0.    1.    0.    0.   ]
 [0.    0.    1.    0.   ]
 [0.999 0.001 0.    0.   ]]
```

Fig 3.18 Matrix of data points

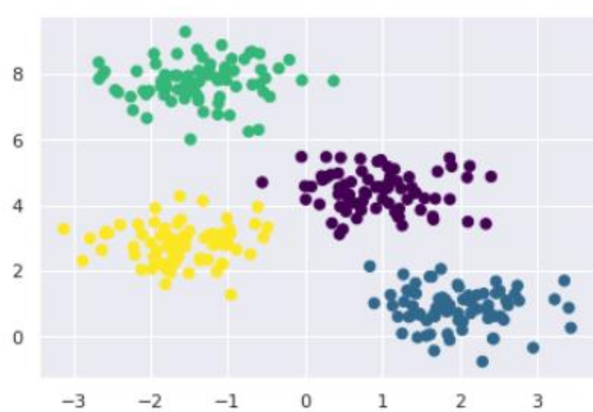


Fig 3.19 Probabilistic clusters of data points

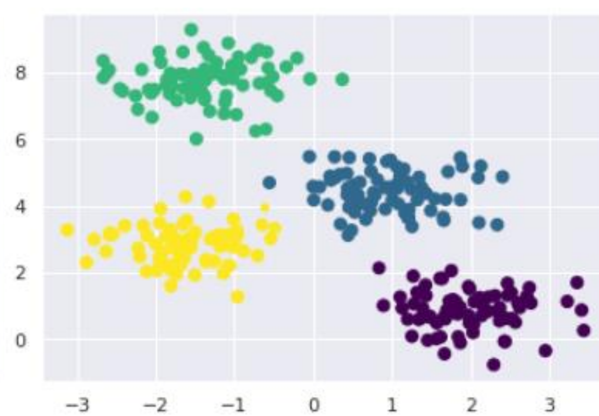


Fig 3.20 Uncertainty of data points

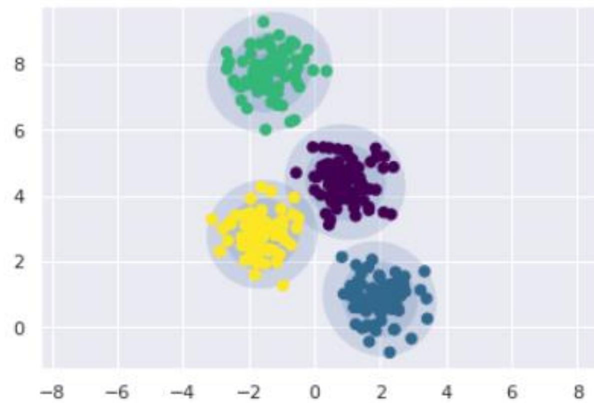


Fig 3.21 GMM Clusters

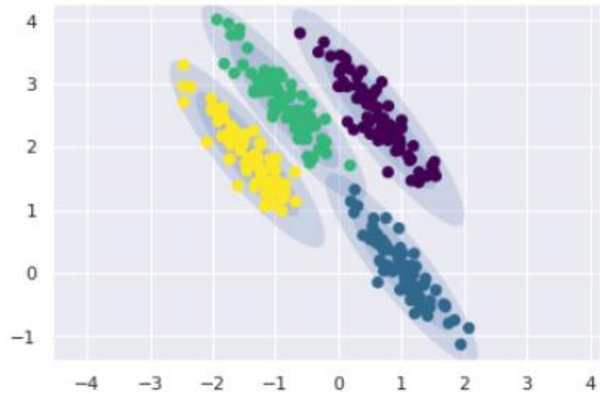


Fig 3.22 Stretched-out GMM clusters

3.4.3: Implementing Gaussian Mixture Model to Segment SEM images:

It uses an expectation–maximization(E-M) approach which does the following:

1. Choose starting guesses for the location and shape
2. Repeat until converged:
    a. *E-step*: for each point, find weights encoding the probability of membership in each cluster
    b. *M-step*: for each cluster, update its location, normalization, and shape based on *all* data points, making use of the weights

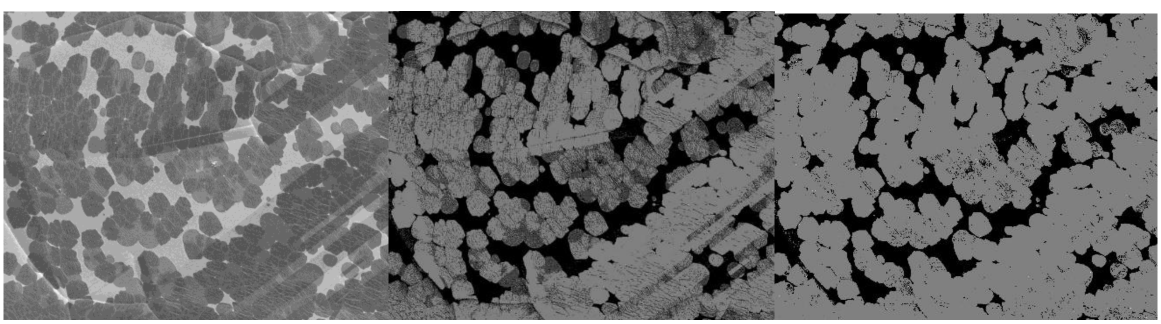
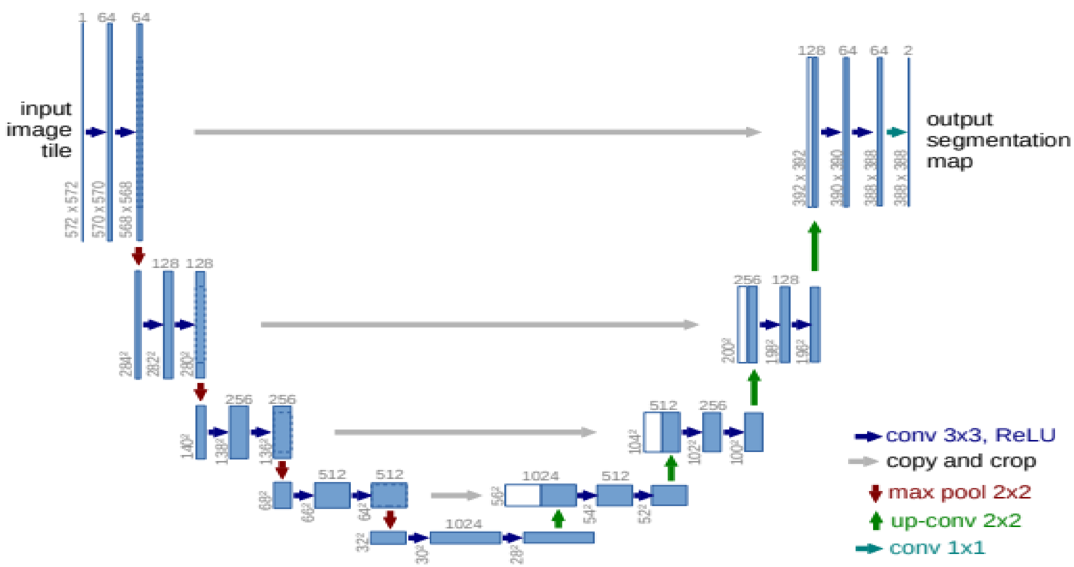


Fig 3.24: Original and segmented graphene image for n components=3&2 respectively

3.5: Implementing a U-Net model to segment microscopy images

The architecture of U-Net contains two paths. First path is the contraction path (also called as the encoder) which is used to capture the context in the image. The encoder is just a traditional stack of convolutional and max pooling layers. The second path is the symmetric expanding path (also called as the decoder) which is used to enable precise localization using transposed convolutions. Thus, it is an end-to-end fully convolutional network (FCN), i.e., it only contains Convolutional layers and does not contain any Dense layer because of which it can accept image of any size. U-Net is dedicated to solving this problem. The reason it is able to localize and distinguish borders is by doing classification on every pixel, so the input and output share the same size.



[source: U-Net: Convolutional Networks for Biomedical Image Segmentation by Olaf Ronneberger, Philipp Fischer, and Thomas Brox]

Fig 3.25 U-Net Architecture

U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations

First in sight, it has a “U” shape.

In the contracting path, the image is going to be downsized and increases in depth with each convolution operation

In the expansive path, the image is going to be upsized to its original size.

3.5.1: Let us build a U-net model to train over graphene SEM images:

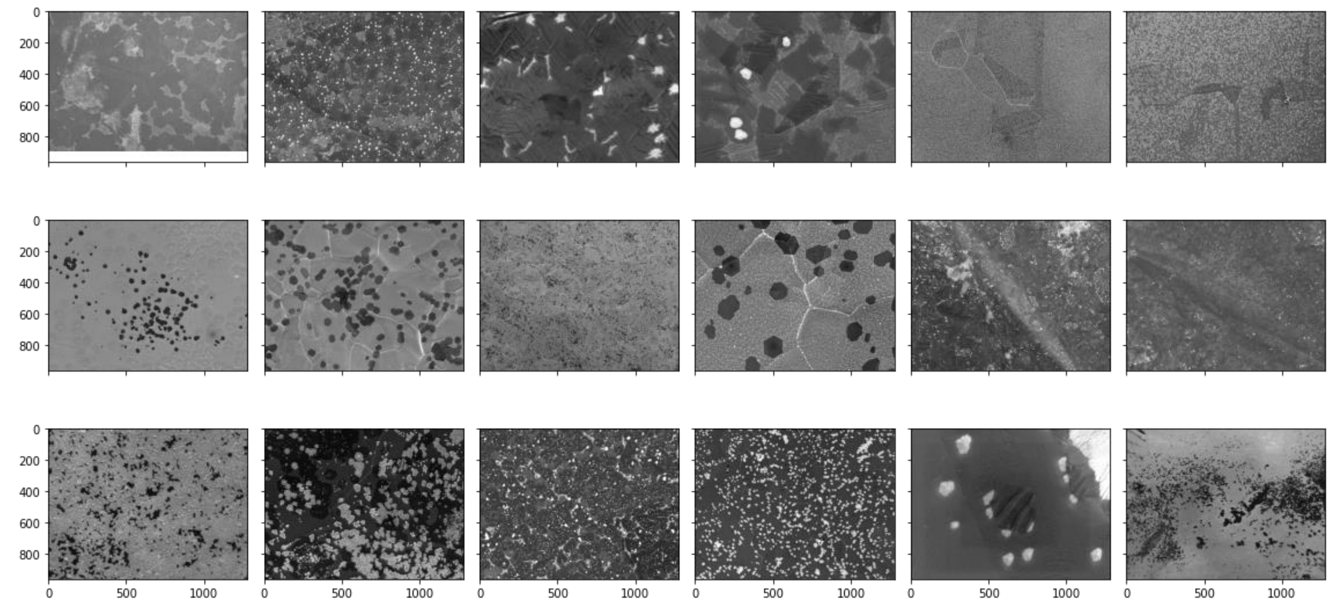


Fig 3.26 Raw SEM images of graphene over a copper layer

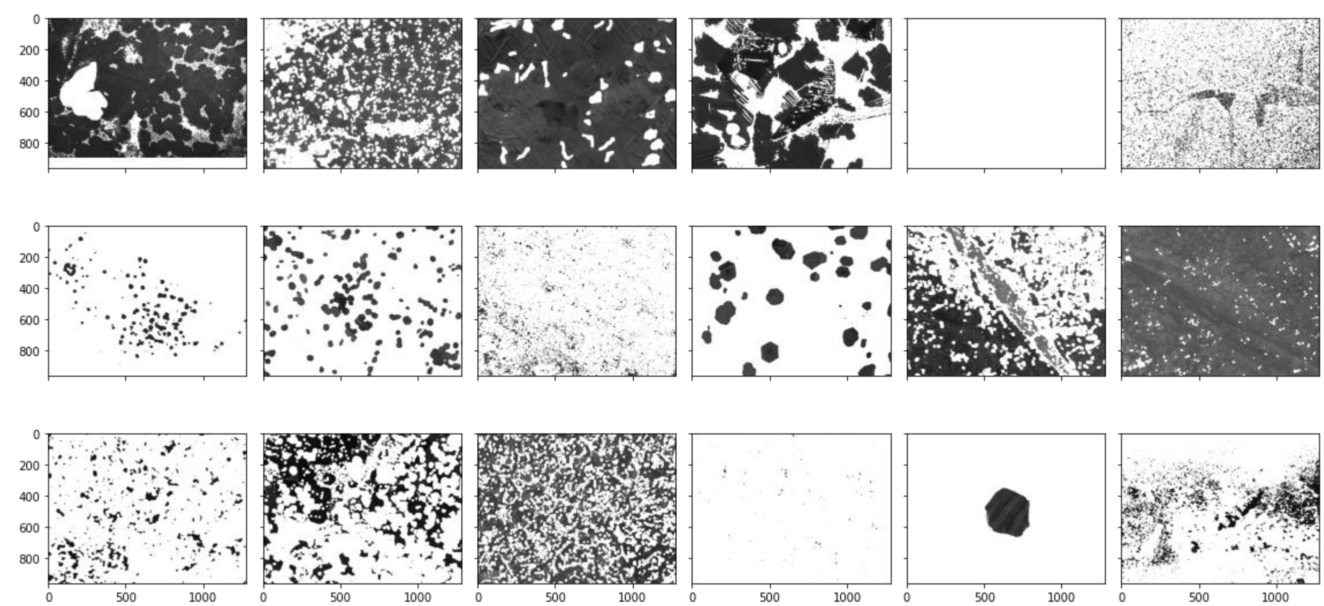


Fig 2.27 Ground truth 'masks'

Using RMSProp (or root mean squared propagation) optimizer to optimize the learning rate, binary cross entropy function as our loss function and accuracy as our performance metric

For initial learning rate e^{-5}, total no. of epoch = 5:

EPOCH:	**TRAINING LOSS:**	**TRAINING ACCURACY:**
0	0.7700497508049011	0.25020313262939453
1	0.6945462822914124	0.7304439544677734
2	0.6942190527915955	0.7348470687866211
3	0.6939237117767334	0.7314462661743164
4	0.6937770247459412	0.7352943420410156
TOTAL TIME: 9 MIN 35 SECONDS	**VALIDATION LOSS:** 0.6932294368743896	**VALIDATION ACCURACY:** 0.05227088928222656

Table 1: Performance Metric 1

For initial learning rate $1e^{-4}$, total no. of epoch = 5:

EPOCH:	**TRAINING LOSS:**	**TRAINING ACCURACY:**
0	0.693673849105835	0.7389278411865234
1	0.6930414438247681	0.7555446624755859
2	0.6929898858070374	0.7444896697998047
3	0.6929426789283752	0.746769905090332
4	0.69289231300354	0.7547016143798828
TOTAL TIME: 10 MIN 32 SECONDS	**VALIDATION LOSS:** 0.6936641931533813	**VALIDATION ACCURACY:** 0.05384254455566406

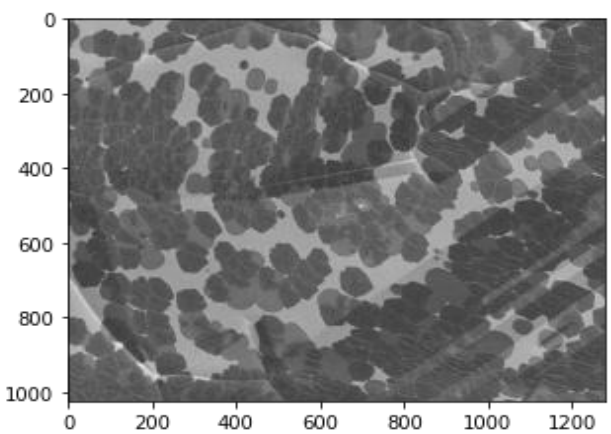
Table 2: Performance Metric 2

For initial learning rate 1e^{-4}, total no. of epoch = 100:

EPOCH:	**TRAINING LOSS:**	**TRAINING ACCURACY:**
0	0.693673849105835	0.7389278411865234
1	0.6930414438247681	0.7555446624755859
2	0.6929898858070374	0.7444896697998047
.	.	.
.	.	.
.	.	.
98	0.129426789268372	0.97467699090338
99	0.12892313070354	0.975470161437988
TOTAL TIME: 262 MIN 47 SECONDS	**VALIDATION LOSS:** 0.1931533813	**VALIDATION ACCURACY:** 0.9742436885447675

Table 3: Performance Metric 3

3.5.2: Using the trained U-net model to segment SEM images of graphene



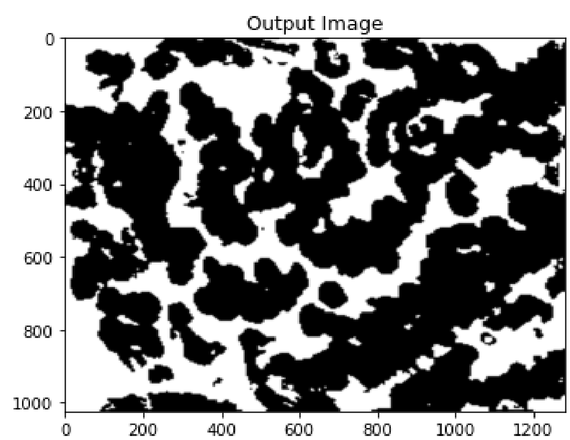


Fig 3.28 Input Image for segmentation

Fig 3.29 Output Image

Coverage: 0.6268137693405151

4. Conclusions

4.1: Conclusion

Exploring various segmentation techniques, we got a better understanding of the parameters involved and how those parameters affect the segmentation of the input image

In Template Matching, we observed that on the segmentation is prone to slight changes in the choice of template and larger the size of the template, higher the computational cost. We tried resolve this issue by down sampling the image, however this resulted in lower resolution output image, which is undesirable for the segmentation of SEM images.

The k-means was simple to perform but the issue is, k-means does not detect pixels as 'graphene' or 'not graphene', it simply bunches together similar looking pixels into one group i.e., it cannot identify individual graphene molecules and separate them at their edge. So, we cannot count the instances of graphene that occurs in a single image. These overlapping regions need to be segmented in order to measure their size and shape, without any bias. Similar issues occur with GMM

While implementing GMM, we observed one important characteristic of K-means that it is a hard clustering method, which means that it will associate each point to one and only one cluster. A limitation to this approach is that there is no uncertainty measure or probability that tells us how much a data point is associated with a specific cluster. So, what happens if we used a soft clustering method instead of a hard one, in which individual points can belong many clusters and how strongly each point is associated with a particular cluster is determined by some probability function? This is exactly what GMMs, attempt to do. GMM uses expectation-maximization algorithm for finding the right model parameters, that makes it an efficient and effective clustering technique compared to traditional clustering technique. Due to this soft clustering technique, we observed that gaussian model performed better than k-means at segmenting the same SEM image

The U-Net model we used inputs an image of size 256x256X1 pixels and outputs the same resolution the segmented image. The model performs 3X3 convolution twice (256X256x64) and then maxpooling reduces the size of the image to half i.e. 128X128X64.The process is repeated 3 more times till we reach the bottom-most layer(16X16X512) where we find still 2 convolutional layers, but with no max pooling. The image at this moment has been resized to 16x16x1024. Now let's get to the expansive path.

In the expansive path, the image is going to be upsized to its original size by using 2x2 transposed convolution, the image is upsized from 16x16x1024 → 32x32x512, and then, this image is concatenated with the corresponding image from the contracting path and together makes an image of size 32x32x1024. The reason here is to combine the information from the previous layers in order to get a more precise prediction. Same as before, this process is repeated 3 more times. Now we've reached the uppermost of the architecture, the last step is to reshape the image to satisfy our prediction requirements. The last layer is a convolution layer with 1 filter of size 1x1. And the rest left is the same for neural network training

4.2: Scope of research

The literature is replete with works focused on the segmentation and enhancements of not only the normal images but also the specialized images - especially the images of porous materials, containing voids. However, the segmentation of an SEM image is a challenging task, especially when it has overlapping objects/regions. These overlapping regions need to be segmented in order to measure their size and shape, without any bias. The watershed algorithm, for its efficiency and ease of use, may be a good choice for segmenting normal images with particular characteristics. The same may not be true in case complex and textured images, like SEM, and the use of watershed segmentation may lead to over- or under-segmentation.

We can explore more about applying a Discrete Curvelet Transform to SEM images, in order to get the fine details in the frequency domain before processing the image for segmentation. In addition, the Curvelet transform enables us to perform the analysis with various block sizes. So, the segmentation method will involve an additional preprocessing step, aimed at the noise removal, in order to avoid false edges.

In the above experiments we have mostly performed basic algorithms of Template matching, kmeans, gaussian mixture model and without much optimization of parameters.

Improvements to template matching can be made to the matching method by using more than one template (eigenspaces), these other templates can have different scales and rotations. Template matching can further be improved by using object detection techniques. Since graphene is a 2D structure, most of the rotations occur in 2D plane, therefore we can use the sample template and rotate it in different orientations to match over the entire SEM Image

Also, In U-Net for the supervised training of such a segmentation network, a set of paired images is required, i.e., an SEM image and its corresponding ground truth segmentation mask, in which each individual particle in the image is outlined. To avoid the tedious task of manually outlining tens or even hundreds of particles in dozens of SEM images, we can explore the use GANs (Generative adversarial networks) for the automated generation of the training data. In a typical GAN architecture, a generator network and a discriminator network are trained against each other in an unsupervised learning process. Thereby created fake SEM images together with their corresponding segmentation masks and increasing our training dataset.

The right choice of meaningful metrics for comparing segmentation masks for a specific task can be challenging and is still a subject of ongoing debates. And is another region to explore on.

5. References

- https://towardsdatascience.com/artificial-intelligence-vs-machine-learning-vs-deep-learning-2210ba8cc4ac by Artem Oppermann
- https://www.deeplearningbook.org by Ian Goodfellow and Yoshua Bengio and Aaron Courville
- http://neuralnetworksanddeeplearning.com By Michael Nielsen
- nanohub.org
- https://www.analyticsvidhya.com/blog/2021/04/k-means-clustering-simplified-in-python/
- https://towardsdatascience.com/machine-learning-algorithms-part-9-k-means-example-in-python-f2ad05ed5203
- https://jakevdp.github.io/PythonDataScienceHandbook/05.11-k-means.html
- https://nanonets.com/blog/how-to-do-semantic-segmentation-using-deep-learning/
- U-Net: Convolutional Networks for Biomedical Image Segmentation by Olaf Ronneberger, Philipp Fischer, and Thomas Brox
- Ciresan, D.C., Gambardella, L.M., Giusti, A., Schmidhuber, J.: Deep neural networks segment neuronal membranes in electron microscopy images. In: NIPS. pp. 2852–2860 (2012)
- International journal of computer applications by Paridhi Swaroop and Neelam Sharma
- International journal of computer science and mobile computing by Dilpreet Kaur and Yadwinder
- https://www.upgrad.com/blog/image-segmentation-techniques/ by Pavan Vadapalli
- https://www.analytixlabs.co.in/blog/what-is-image-segmentation/