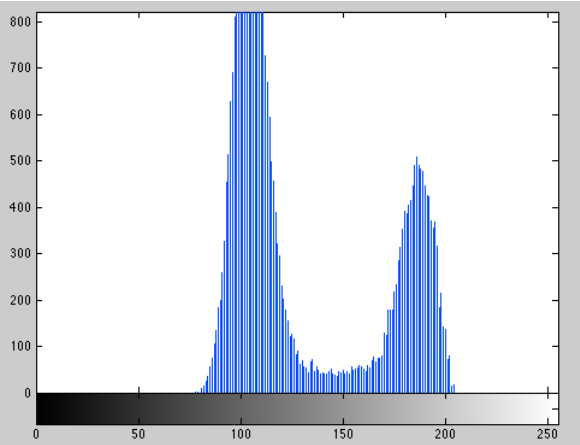
COC202 Computer Vision
Lab 8 – Image segmentation

In this lab, you will use two basic image segmentation techniques: thresholding and clustering.

If you have not yet finished the exercises from the previous lab, do them first.

1. Load the *rice.png* image which you can download from learn. Now perform thresholding on the image, so that the rice grains in the image are assigned to the foreground (i.e. white) and everything else to the background (i.e. black).

   Obviously, the tricky part here is to select an appropriate threshold. Using the `imhist()` function and inspecting the histogram should help you here.



2. Find appropriate functions from Matlab's Image Processing Toolbox to perform thresholding including automatic threshold selection and run it on the image.

   Which algorithm is implemented here and how does it select the threshold?

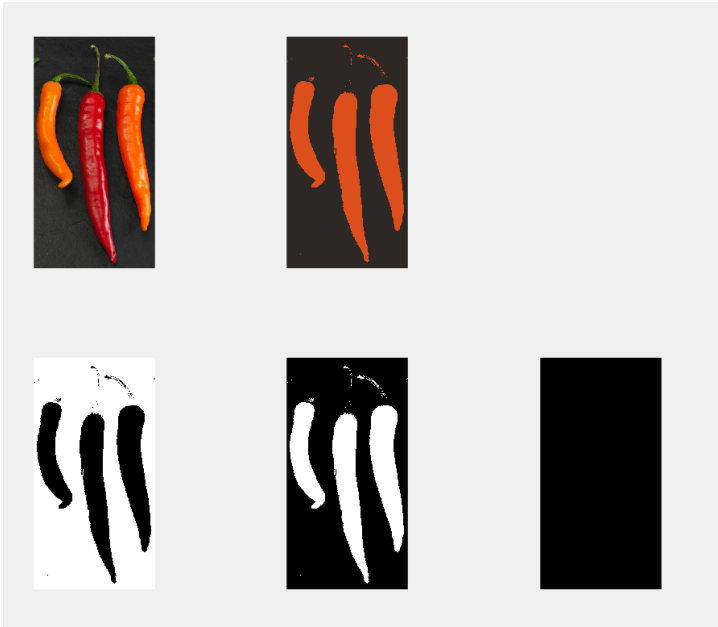
3. Write a function that implements the *k*-means clustering algorithm in Matlab. The function should be passed a data matrix that contains the samples and *k,* and return the calculated cluster centres and the assignment of samples to the clusters.

   Hint: You might find it useful to implement a function that performs a nearest-neighbour search, i.e. a function that would be passed a set of cluster centres and the data matrix, and returns, for each sample, the closest cluster.

   Hint 2: When performing the nearest neighbour search, it is sufficient to perform it based on squared Euclidean distances (which will be faster than calculating Euclidean distances).

   Hint 3: For testing if the clustering has converged sufficiently, you can test whether the change in terms of mapping error that would be introduced through the clustering (i.e., the differences between the original and clustered data) from one iteration to the next falls below a pre-defined (small) threshold.

4. Now, download the *peppers.jpg* image from learn and use the *k*-means clustering algorithm to segment the image. Experiment with various values of *k* to see their effect on the resulting segmentation. For evaluation/display you can show the resulting image so that each pixel is replaced by the colour of the corresponding cluster (centre), and/or individual images that show each segment as a binary image.



Once you have finished all exercises you may leave the lab.

Additional exercises for further study:

5. When looking at the results you obtain from multiple runs, you will probably notice that with a simple random initialisation (i.e., assigning random pixel values as initial cluster centres) it can, depending on the dataset/image, occur that empty clusters are created.

   Modify your code so that the initial cluster centres are chosen as (unique) random pixel values that actually occur in the image (which in turn will assure that at least a single pixel will be assigned to each cluster).