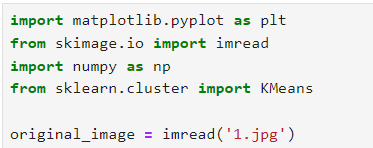
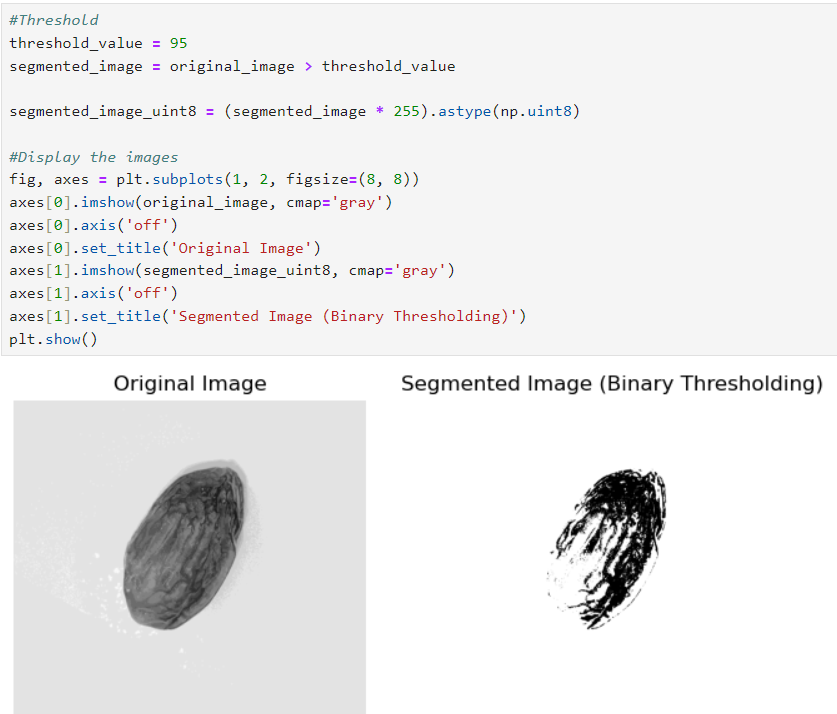
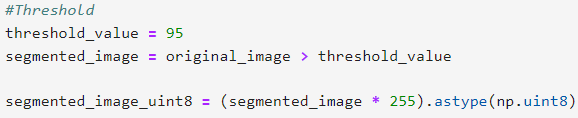
At the beginning, I’ve imported the necessary libraries for reading and showing the images, and for two segmentation methods (Binary thresholding, and K-means).



First, applying the first method (**Binary thresholding**)



Basically, the code consists of 3 lines only, the rest are for showing the images



Here I’ve chosen 95 as a value for threshold, this will classify which pixel will be a background and which as foreground.

Then, if the pixel value > 95, it will be True, (Foreground)

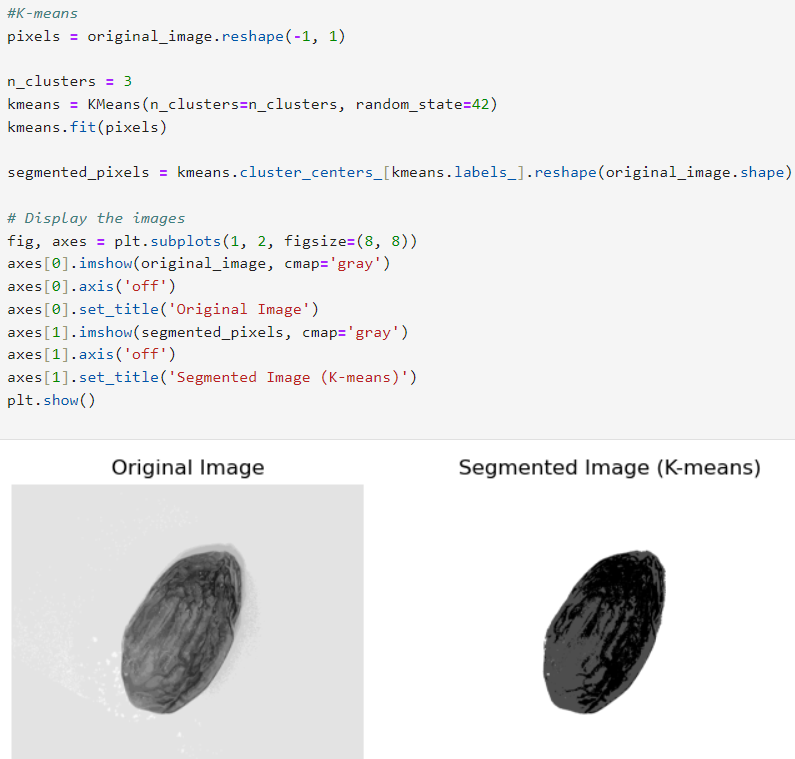
And if the pixel value <= 95, it will be False, (Background)

Then the Boolean array will become an integer array by converting:

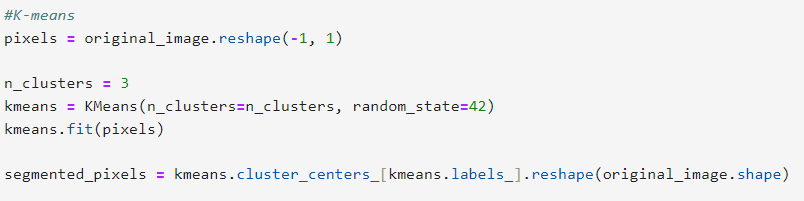
True = 255 “White”, False= 0 “Black”

Finally, the original image and the segmented image will be shown

Second, applying the first method (**K-means**)



Like the first method, the last part was for showing the images, so let’s talk about the important part in the code.



At first, I changed the image to be as a 2D array with one column, which is basically like

a 1D array. It is still technically 2D-Array, that’s good for separating the data and the K-means expecting a 2D-Array.

Then I specified that I want to segment the image into 3 distinct clusters.

I’ve searched through google and I found out that putting a random\_state is very important for controlling the randomness of clustering.

Then, each pixel is replaced by the color of the cluster it belongs to.

Finally, the original image and the segmented image will be shown