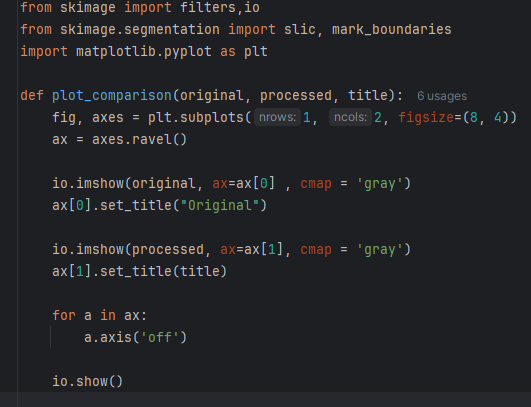
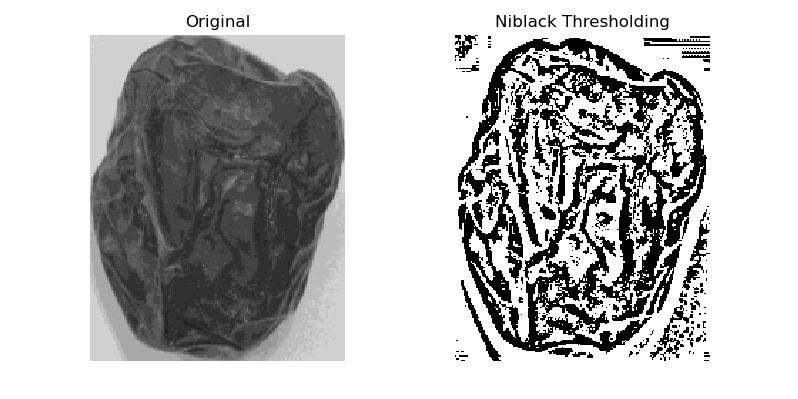
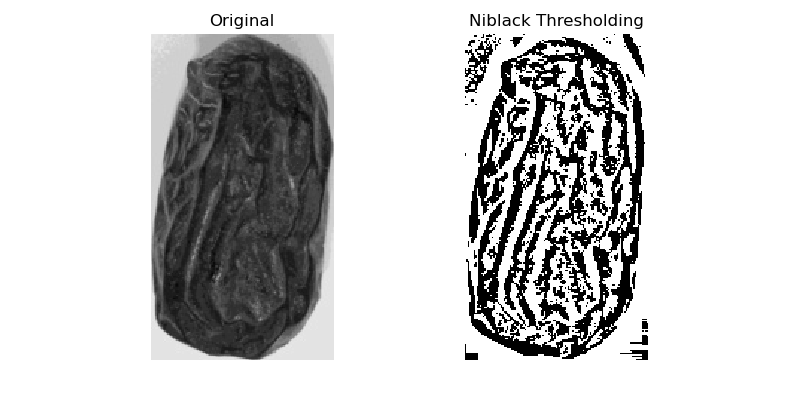
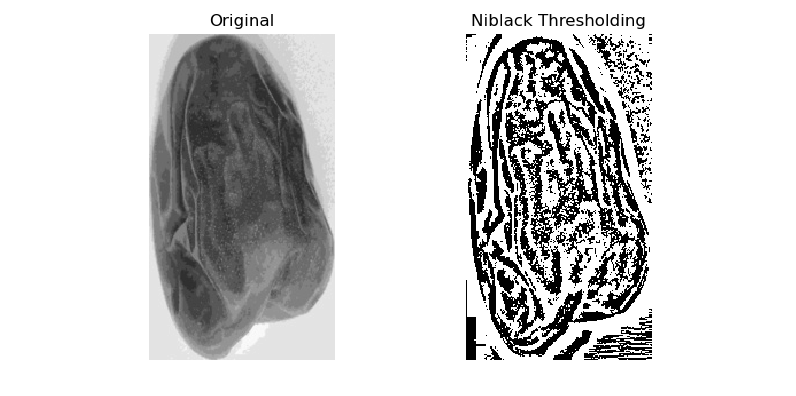
The Image below is just how I started the code of segmentation. A few import statements which will be used later and a plot comparison function which plots both the original and the altered version (taken from the previous lab)

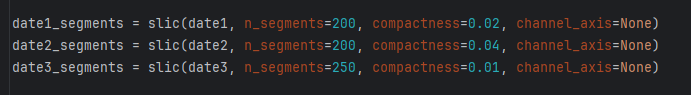


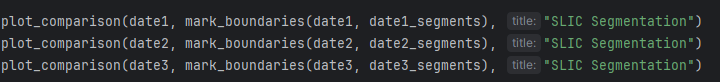
The image below consists of the following: An io.imread to read 3 different date images from the competition dataset (as\_gray to make sure they are grayscale). Then we apply the niblack threshold to each image to get an array of values. Then we turn each image into its binary representation using the threshold. Then we call the function that plots the images.







The image below shows using SLIC method to segment the images (the values below were picked for each image purely based on trial and error), The n\_segments stands for the amount of segments the image will be divided into, and compactness is related to colour and space proximity, the higher the value the more it values the space proximity so each image would be optimised by a different value. The channel axis is None to tell the function that images are grayscale, otherwise it wll assume it is rgb. The image below just shows the comparison using mark\_boundaries function to show the label image on top of the original



The results are shown below:

