COMP9517 ASSI1

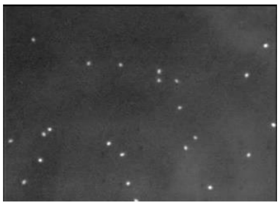
JiaYang Jiang-z5319476

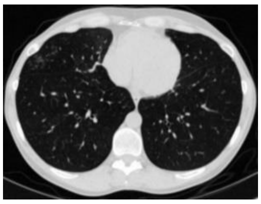
Task1 - Otsu Thresholding

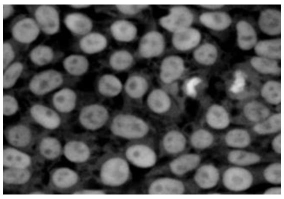
For otsu threshold, we need to find the gray-level which maximises the inter-class variance, then put values of input img below the threshold to black in the output and put values of input img equal/above the threshold to white in the output.

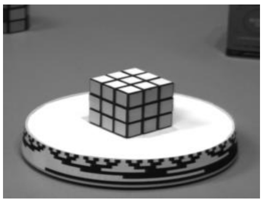
I wrote a function called otsu which accepts a input img and return its best threshold and output img. In the function, I firstly use gray_img.shape to get the height and width of matrix of img. Then I use for-loop to traverse gray-level(0-255), for every gray-level, I created two lists and use a for-loop to calculate the fraction of pixels below the threshold(class 0) and the fraction of pixels equal to or above the threshold(class1). Then for each list, I use formula sum(list)/len(list) to calculate mean (u0 and u1). Finally, find the gray-level which maximises the inter-class variance and end the loop. I finally got the output imgs:

input: (Algae.png -> CT.png -> Nuclei.png -> Rubik.png -> Satellite.png)

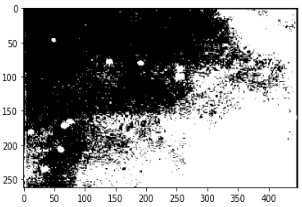


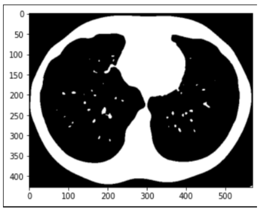


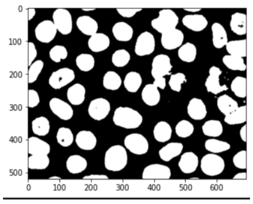


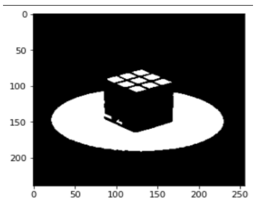


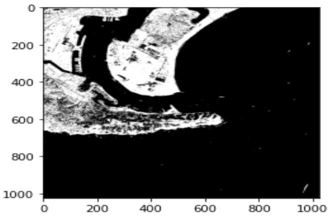
output: (Algae.png -> CT.png -> Nuclei.png -> Rubik.png -> Satellite.png)







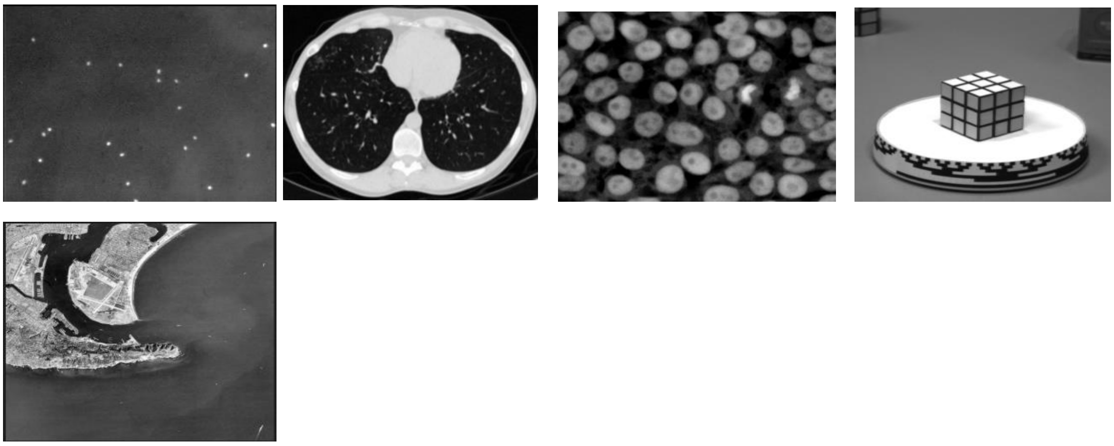




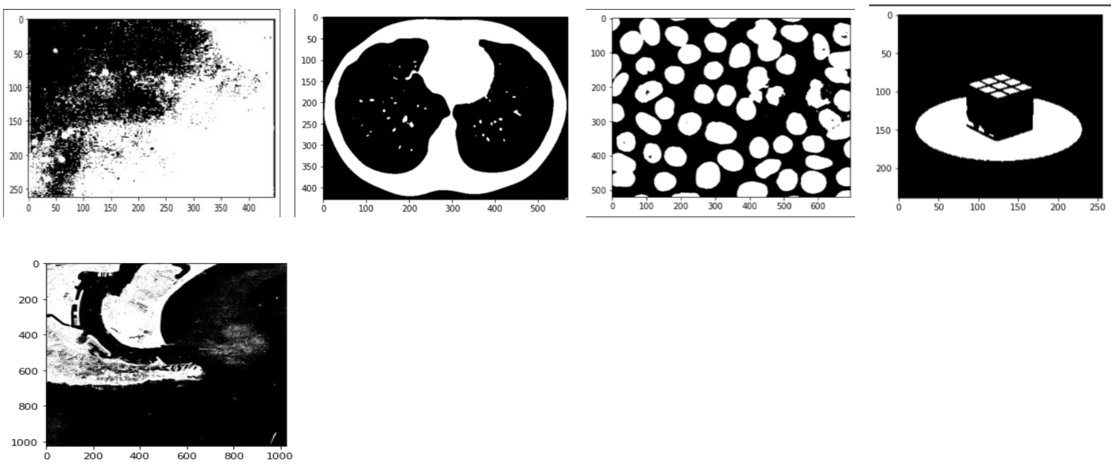
Task2 - Isodata Thresholding

For isodata thresholding, I wrote a function which accepts a input img and a initial threshold. Firstly, I created two lists and use them to calculate u0 and u1.Then I use a for loop and the formula — t = (u0+u1)/2   to calculate the suitable threshold. If the threshold is equal to the previous threshold, then end the loop and I got the result, else continue the for loop until the next threshold is equal to the previous threshold.

input: (Algae.png -> CT.png -> Nuclei.png -> Rubik.png -> Satellite.png)



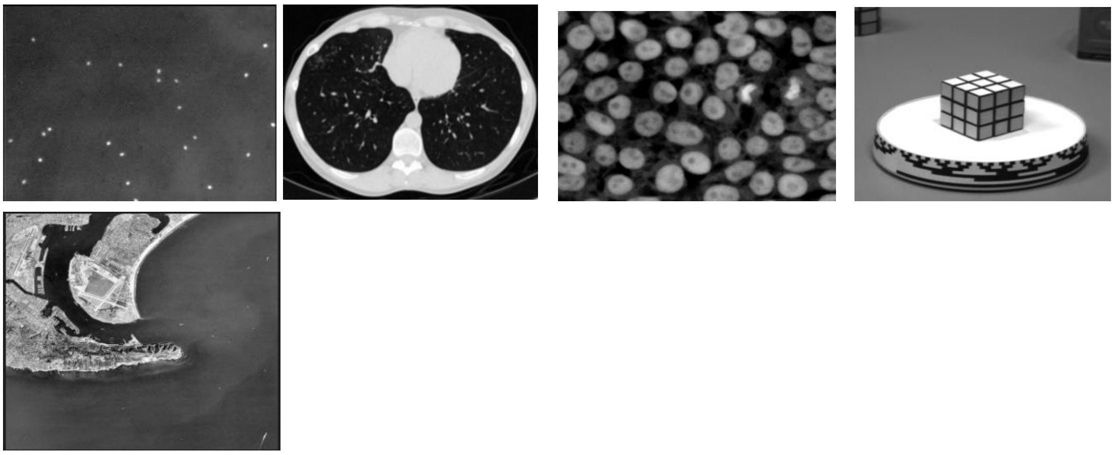
output: (Algae.png -> CT.png -> Nuclei.png -> Rubik.png -> Satellite.png)



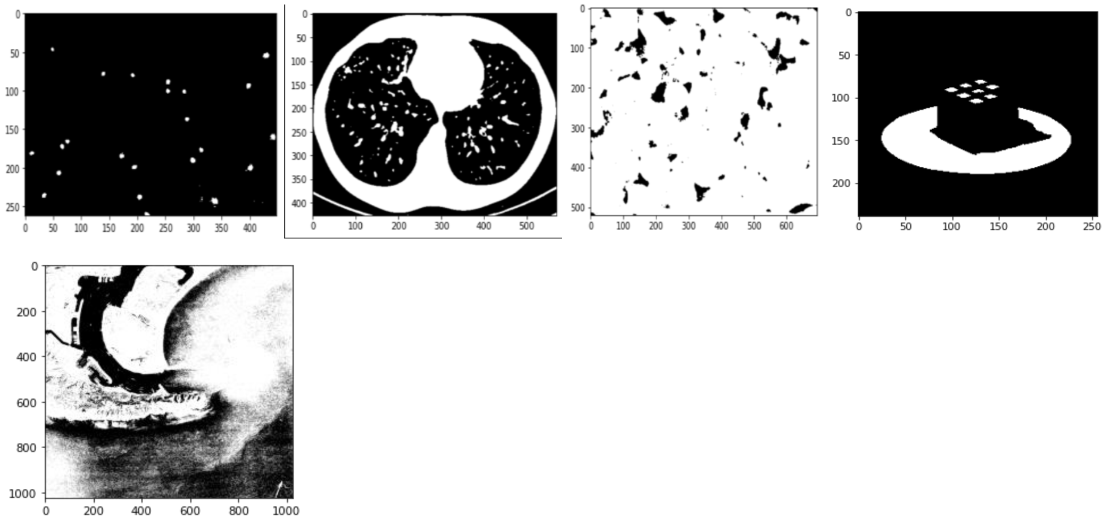
Task3 – Triangle Thresholding

For Triangle threshold, I firstly converted the img matrix into histogram list and wrote a function to calculate the threshold. In the function, I used a for-loop to calculate a straight line from the peak of the histogram to the extreme of the histogram (the highest gray level point) and find the gray level for which the histogram deviates the most from the line. There are two situations which are that the peak of the histogram corresponds to the highest gray level and the peak of the histogram corresponds to the lowest gray level. Therefore , we need to inverted the algorithm for some imgs, so I wrote a bool variable called 'invert' to complete the algorithm and calculate the threshold.

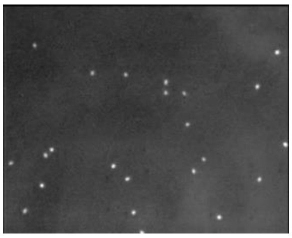
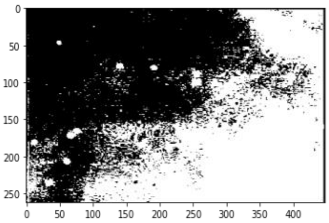
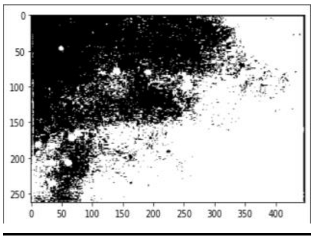
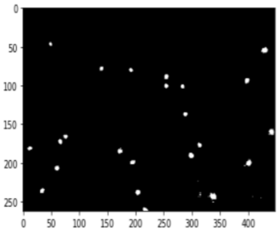
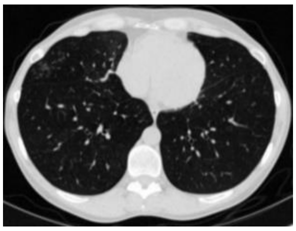
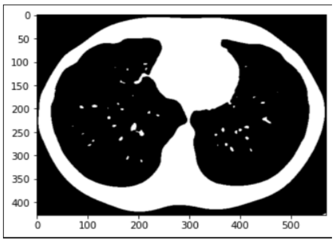
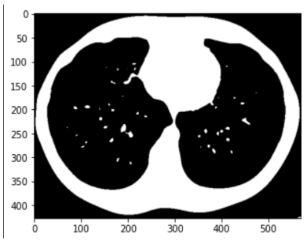
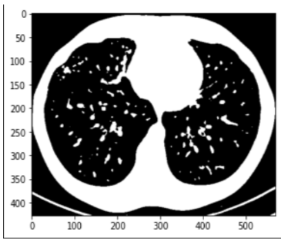
input: (Algae.png -> CT.png -> Nuclei.png -> Rubik.png -> Satellite.png)

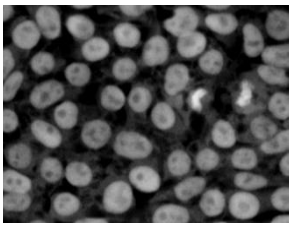
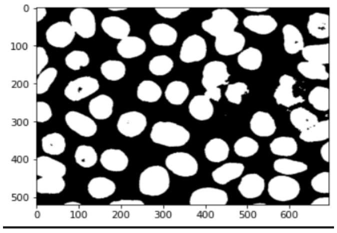
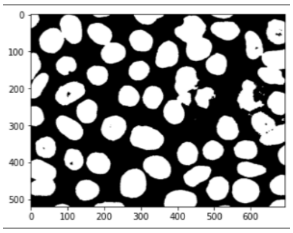
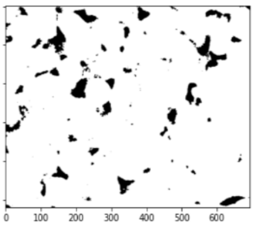
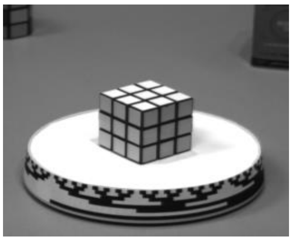
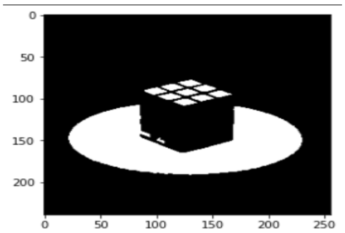
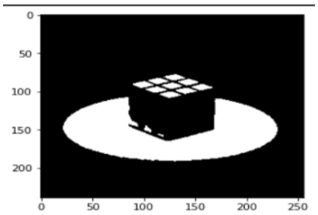
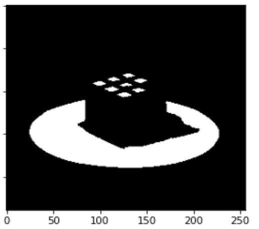
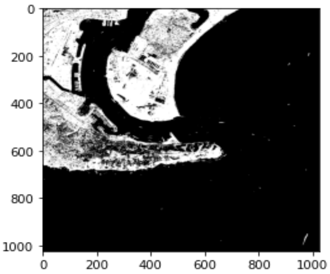
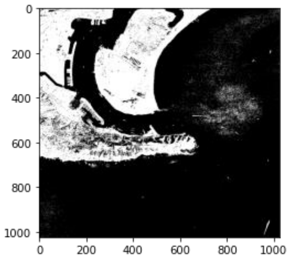
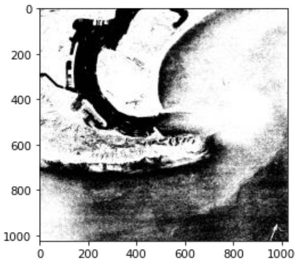


output: (Algae.png -> CT.png -> Nuclei.png -> Rubik.png -> Satellite.png)



Task4

Input img	otsu	isodata	triangle
			
			

Threshold:

Algae:

Otsu – 68 Isodata – 64 Triangle - 97

CT:

Otsu – 116 Isodata – 114 Triangle - 47

Nuclei:

Otsu – 79 Isodata – 80 Triangle - 2

Rubik:

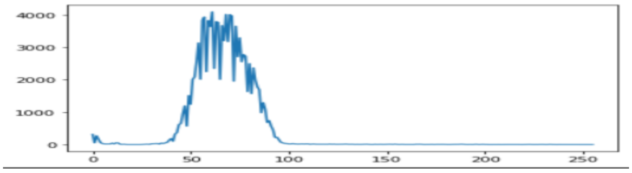
Otsu – 174 Isodata – 172 Triangle - 250

Satellite:

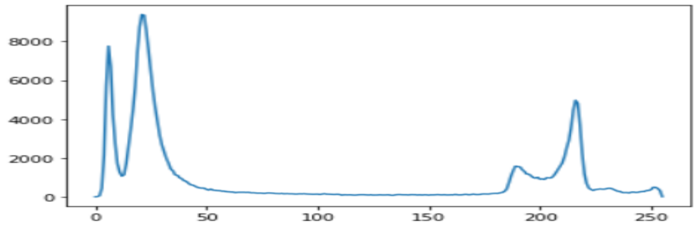
Otsu – 127 Isodata – 100 Triangle - 72

Histogram:

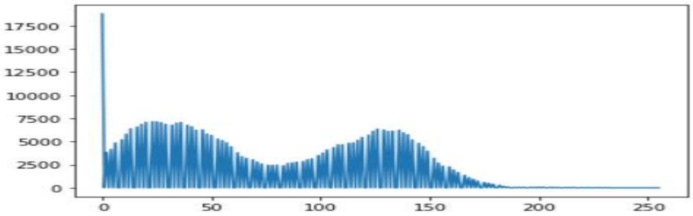
Algae:



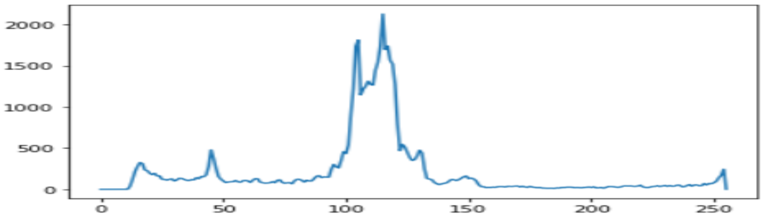
CT:



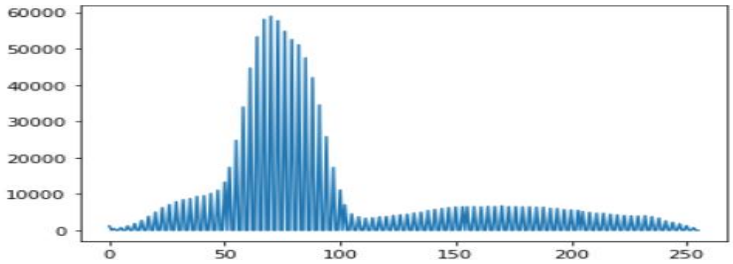
Nuclei:



Rubik:



Satellite:



From the result we can see that the output of otsu and isodata is almost same and there are some differences between the output of Triangle and (otsu and isodata). For example, for Algae.png and Rubik.png, the output(triangle) is darker than output (otsu and isodata), for CT.png, Nuclei.png and Satellite.png, there is more white part of output(triangle) than output (otsu and isdata). This may due to the differences between the histogram of them, some histograms are unimodal and some are multimodal. Also, for different initial threshold, the results are different for isodata thresholding. In conclusion, otsu threshold and isodata threshold are simple and fast in calculation, and is not affected by image brightness and contrast. The disadvantage is sensitivity to image noise; Can only be segmented for a single target; When the ratio between the target and background size is significantly different, and the inter class variance function may exhibit bimodal or multimodal characteristics, the effect is not good. For triangle thresholding, it is fast and suitable for processing images with bimodal distribution of grayscale values.