CS 484 Spring 2022
Homework Assignment 1

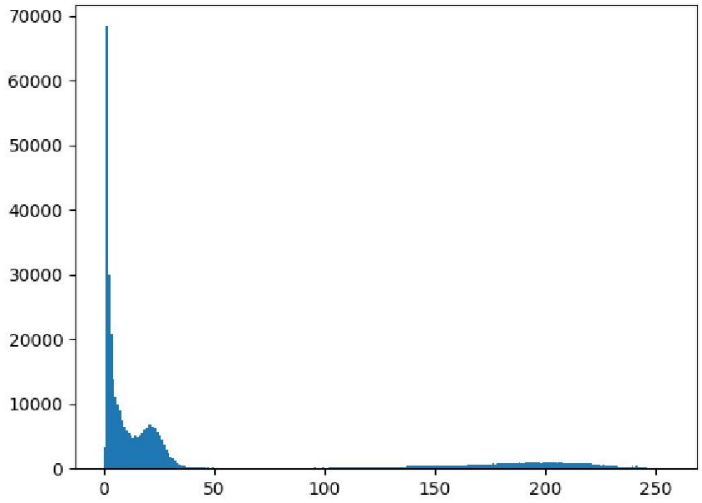
Melike Demirci – 21702346

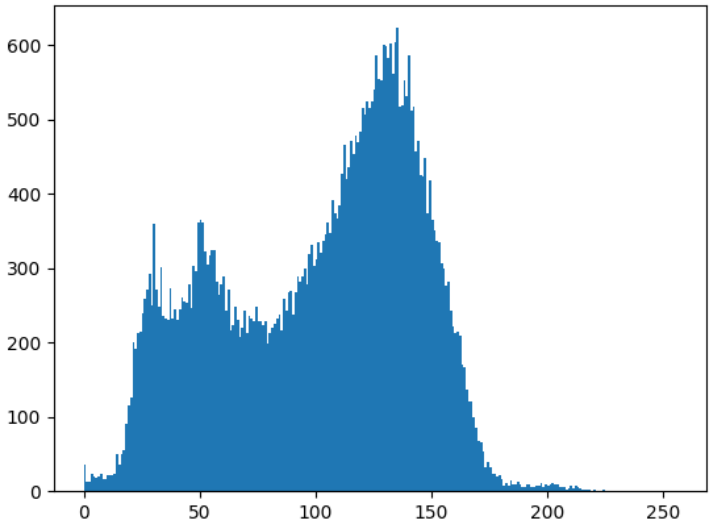
Q1)

Both dilation and erosion functions were implemented as it is described in the lecture slides. To make characters more readable first erosion is implemented in order to connect dark areas while eroding the white areas. However, erosion makes dark noises bigger when it is applied alone. Due to this reason dilation is applied to the eroded image.

Q2)

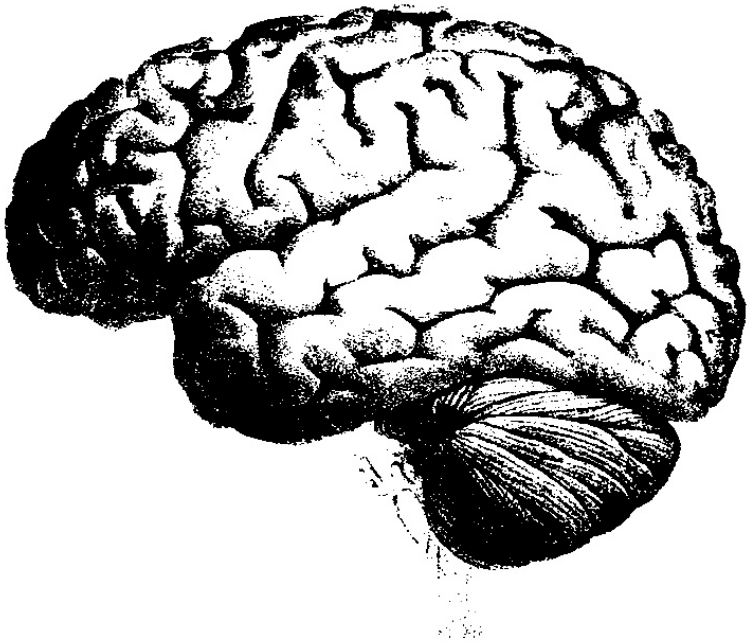
Histogram of the two images were created and compared with the histograms obtained with MATLAB. Highly similar histograms were acquired. By interpreting the plots, the accuracy of the implementation can be verified. In the flower image, there are a lot of dark area which is shown in the histogram with higher number of elements near to the value 0. In the second image there is lower contrast and pixel values more close to each other.



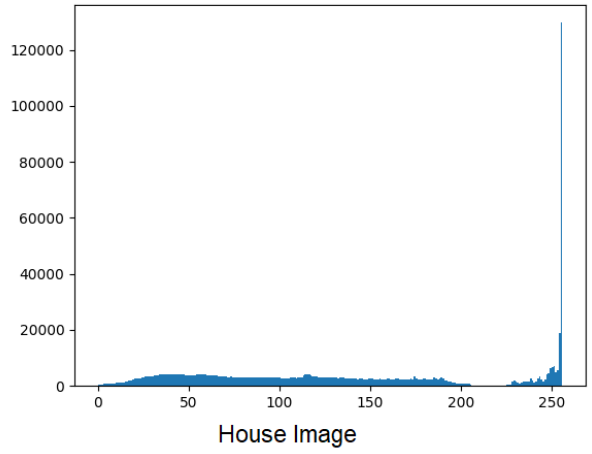


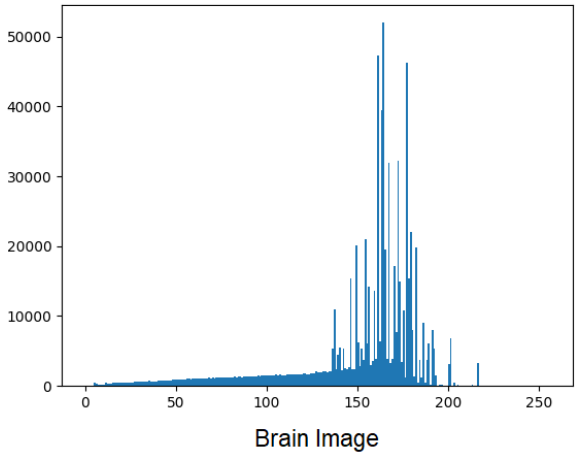
Q3)

Otsu's method was implemented. During the implementation, maximizing the between class variance were preferred due to the computational efficiency [1]. Two images were used to produce binary images. Otsu's method performed better with the brain image compared to the other image which contains a house.



The reason of this performance difference can be understood if we look at histogram of the two images.





In the house image the grayscale values are spread over the entire histogram while in the brain image there is one area which is most of the higher values populated. For the brain image, between class variance can be maximized more compared to the house image.

Q4)

2D convolution were implemented. Boundaries were handled as follows; there is no padding outside the input image, the kernel were started to convolved from half kernel size inside the image. Values for outer pixels left as zero (black).

Sobel filter for both vertical and horizontal edges convolved with the input image. After convolution results were normalized and combined as follows in order to find gradient [2].

$$G = \sqrt{G_x^2 + G_y^2}$$

Same operation conducted with the Prewitt filters. Resulting images are like following.

Prewitt

Sobel

References

[1] http://www.labbookpages.co.uk/software/imgProc/otsuThreshold.html

[2] https://docs.opencv.org/3.4/d2/d2c/tutorial_sobel_derivatives.html