

# Computer Vision 1 HW#2

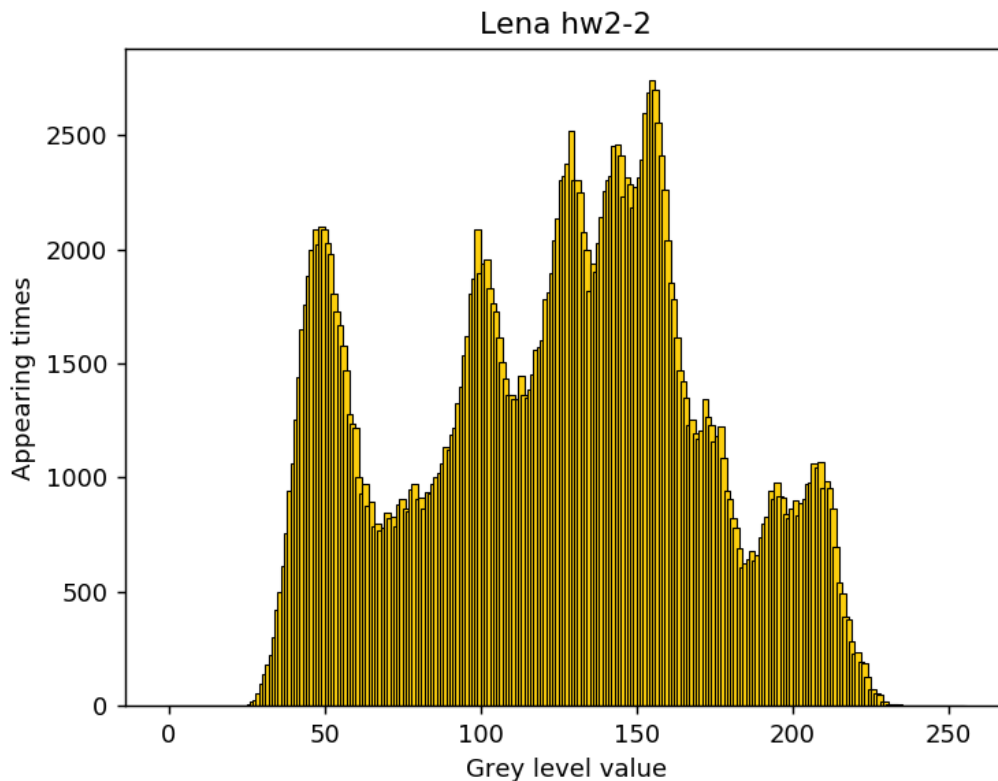
## The first part: Binarizing the image

I used the python module `skimage` to read `lena.bmp` as a two-dimensional array, and then I changed its grey level value according to the threshold, generating the below picture `binarized.png`.



## The second part: Calculating the histogram

Similarly, I used the python module `skimage` to access `lena.bmp` as a 2-D array. Then I traversed the whole array to calculate the numbers of each grey level value appearance, recording it in a list. After that, I plotted the following bar chart(`bar.png`) with `matplotlib`.



## The third part: Finding the connected components

At this part, I was using 4-connected neighborhood detection.

However, I found the connected components by a BFS algorithm, which is adapted from the `recursive algorithm` but using breadth-first-search instead of depth-first-search. In other words, I didn't really do the recursion; I traversed the whole binarized photo and determined the connected pixels with same color by BFS its all neighborhood, and writing down the area, centroids, and bounds into a list at the same time.

After that, I labelled each component with different grey level values and get the below image `connected_component.png`. (The black areas of the photo is the omitted components which are smaller than 500 px)



Then, I drew the bounding boxes(the black rectangles) and centroids(the black dots) due to the previous generated list, generating the below image `bounding_box_with_centroids.png`.

