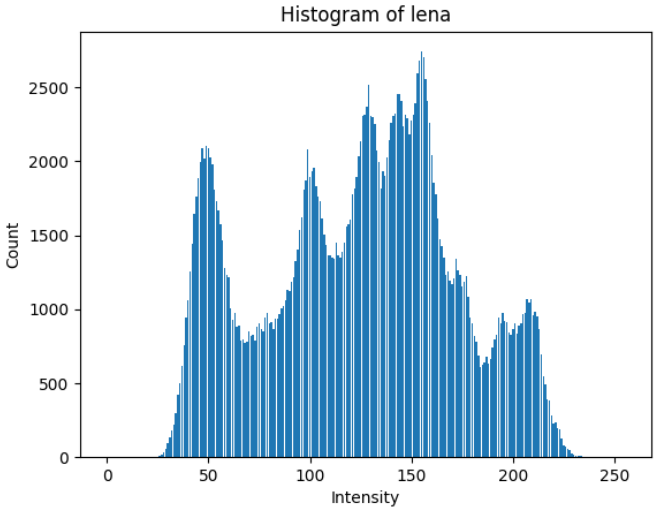
Homework 3

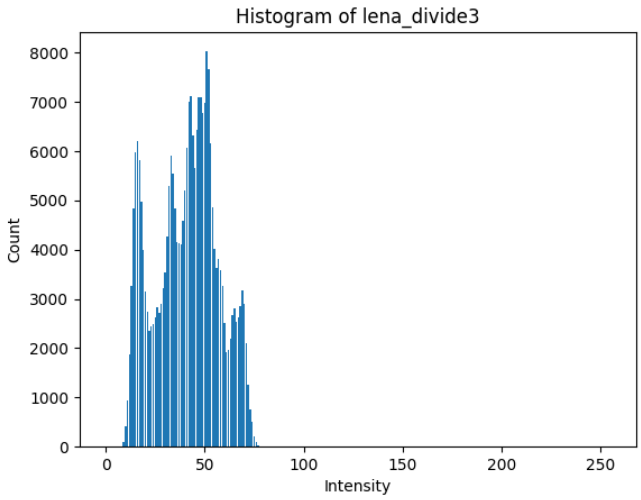
(a)



To calculate histogram, the function `calculate_hist(img)` takes the array of the image read by `cv2.imread()` as input, and create a 1-dimesional array “count”. Then run a two layer for loop, such that for each pixel (r, c) in the image, count[intensity(r, c)] += 1. After the loop finished, the histogram is recorded in the array, and the function return the array.

To draw the histogram, the function `draw_hist(name, count)` takes a string “name” used as file name when storing the picture and the array “count”, return by `calculate_hist(img)`. Then set “index” to be an array of integer from 0 to 255, and use `plt.bar(index, count)` to draw the picture.

(b)

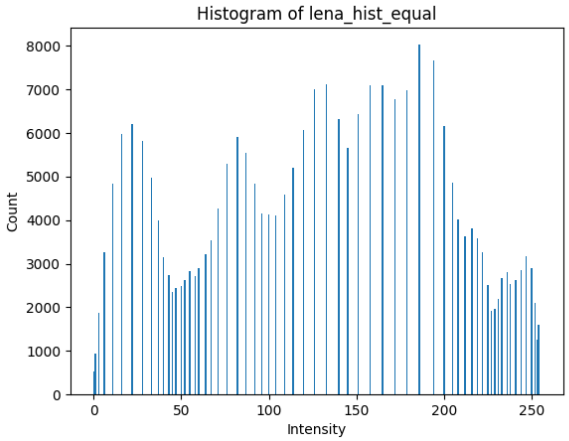



```
for r in range(h):
        for c in range(w):
            img_divide3[r, c] = img[r, c] // 3
```

The image is obtained by run a two layer for loop as above, such that for each pixel (r, c), the intensity of the new_img[r, c] is old_img[r, c] / 3.

The way of calculate and draw histogram is the same is (a).

(c)



The function `hist_equalization(img)` first calculate the histogram of the input image by calling `calculate_hist(img)` and store it in “count”. Then, run a for loop from 0 to 255 in order to calculate the cumulative function of histogram stored in “count”, and store it in an array, “cumulative”. Create an array “s”, such that for i in 0 to 255, s[i] = 255 * (cumulation[i] / n), where n is the total number of pixels. Finally, run a two layer for loop, such that for each pixel (r, c), img_new[r, c] = s[img[r,c]].

The way of calculate and draw histogram of the final image is the same is (a).