

Environment: Python 3.6 with cv2, numpy

* The cv2 module is used only for image I/O.

Benchmark: lena.bmp

Usage: `$> python3 hw3.py [image_path]`

Task: Histogram equalization

Method description:

The program performs histogram equalization on input image. It first produces the image histogram, and then computes the cumulative sum of each intensity value.

The new value s_k of each intensity value k is then given by the following:

$$s_k = 255 \sum_{j=0}^k \frac{n_j}{n},$$

where k ranges from 0 to 255, n_j is the number of pixels with intensity j and n is the total number of pixels. For every pixel in the image, we reassign the pixel with intensity k with the new value s_k .

As in HW2, the image histogram is visualized using the Microsoft Excel.

Principal code fragment:

```
def equalization(img):
    new_img = np.zeros(img.shape, np.uint8)
    pixels = img.shape[0] * img.shape[1]

    count = count_bins(img) # compute image histogram (as in hw2)
    cumulative_count = count.copy()
    for i in range(len(count)):
        cumulative_count[(i+1):] += count[i]
    new_intensity = np.uint8(255. * cumulative_count / pixels)

    for i in range(img.shape[0]):
        for j in range(img.shape[1]):
            new_img[i, j] = new_intensity[img[i, j]]
    return new_img
```

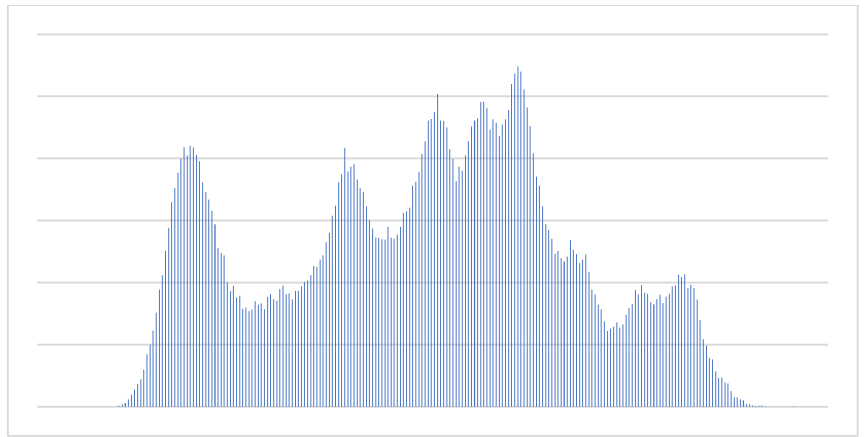
Results:

Besides the original benchmark image, a darkened version (intensity scaled by 1/3) is also used to test the program functionality.

The resulting images are shown below, as well as their corresponding histograms.



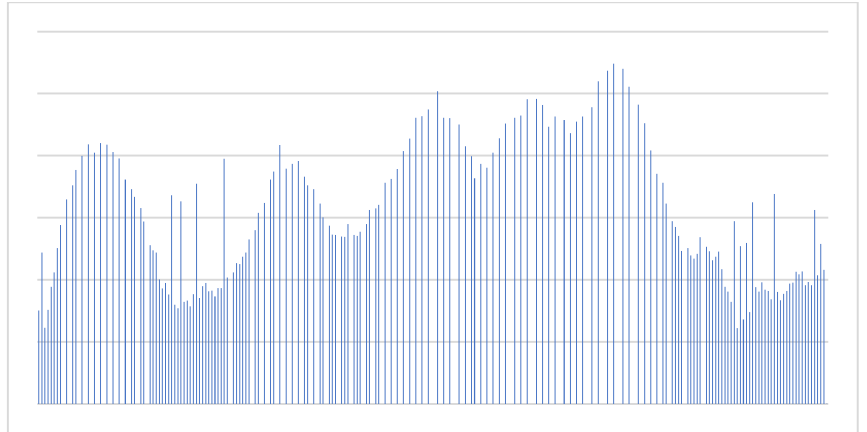
(a) Original lena.bmp



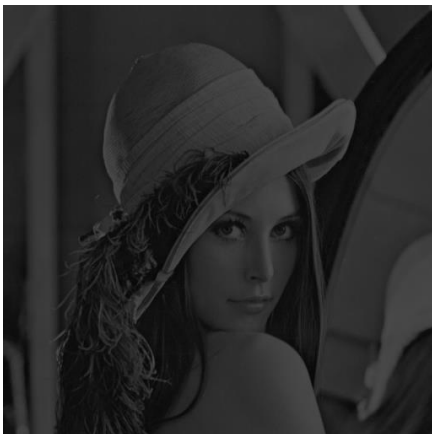
Histogram of (a)



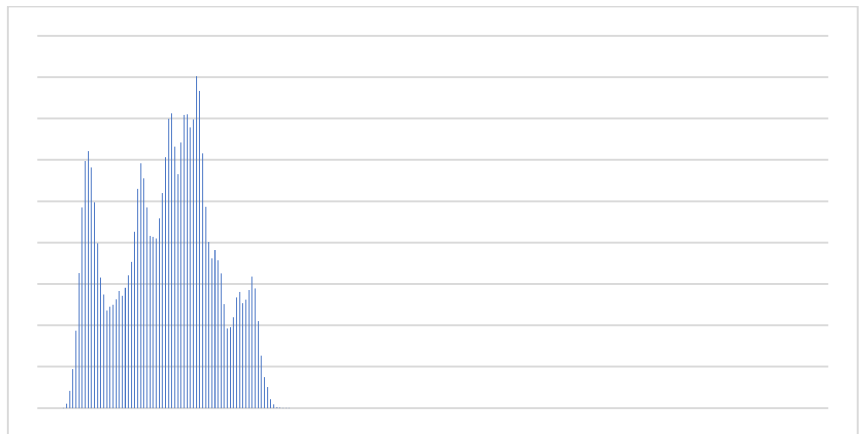
(b) Equalized



Histogram of (b)



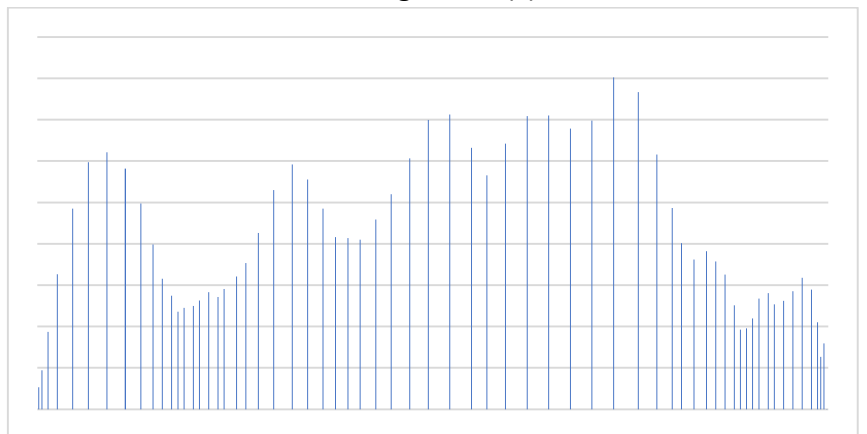
(c) Darkened



Histogram of (c)



(d) Equalized after darkened



Histogram of (d)