

# DIP homework2

---

16327109 谢昆成

## 1 Exercises

---

### 1.1

A second pass of histogram equalization will produce exactly the same result as the first pass.

Proof

The transform function is

$$s = T(r) = (L - 1) \int_0^r p_r(w)dw$$

and

$$\frac{dr}{ds} = \frac{dT(r)}{dr} = (L - 1)p_r(r)$$

After the first pass,

$$p_s(s) = p_r(r) \left| \frac{dr}{ds} \right| = p_r(r) \cdot \frac{1}{(L - 1)p_r(r)} = \frac{1}{L - 1}$$

Then we do the second pass,

$$p_t(t) = p_s(s) \left| \frac{ds}{dt} \right| = p_s(s) \cdot \frac{1}{(L - 1)p_s(s)} = \frac{1}{L - 1} = p_s(s)$$

finished

### 1.2

(1)

after filtering

$$\begin{bmatrix} -178 & -418 & -263 & -255 \\ -80 & -223 & -233 & -93 \\ 132 & 332 & 119 & 145 \\ 170 & 333 & 353 & 203 \end{bmatrix}$$

(2)

It means the extent how gray value changes in the y direction. The bigger the value after filtering, the more rapidly the gray value change, and vice versa.

(3)

It is the difference filter in y direction. It can be used to detect the edges in x direction.

(4)

solving the equation set

$$\begin{cases} c + d + e = -2 \\ b + c + d + e = -1 \\ b + c + e = -2 \\ a + b + c + d + e = 0 \\ a + c + d + e = -1 \\ a + b + c + e = -1 \\ a + b + c + d = -1 \end{cases}$$

filter

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

## 2 Programming Tasks

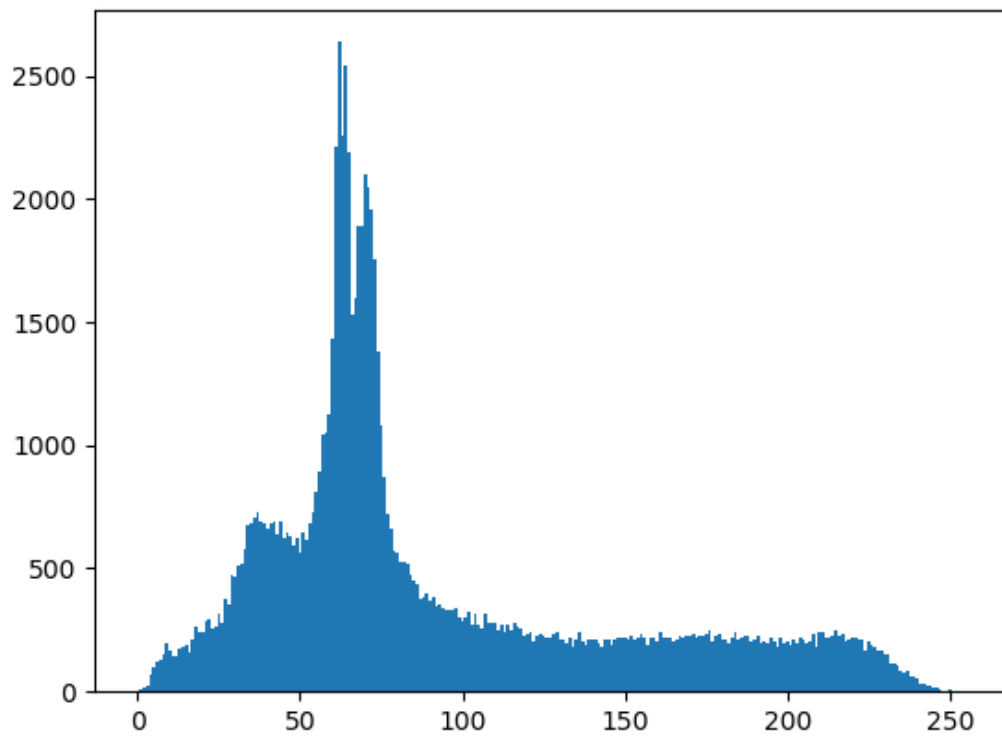
---

My student ID is 16327109, so I choose "09.png" as my input.

### 2.1

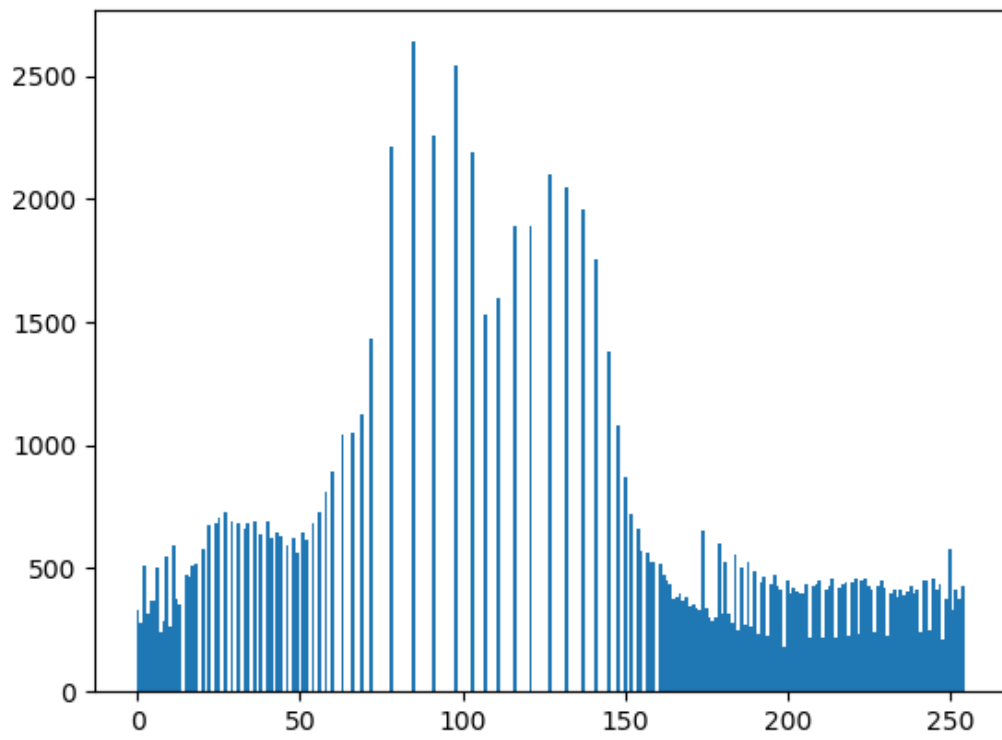
(1) Compute and display its histogram





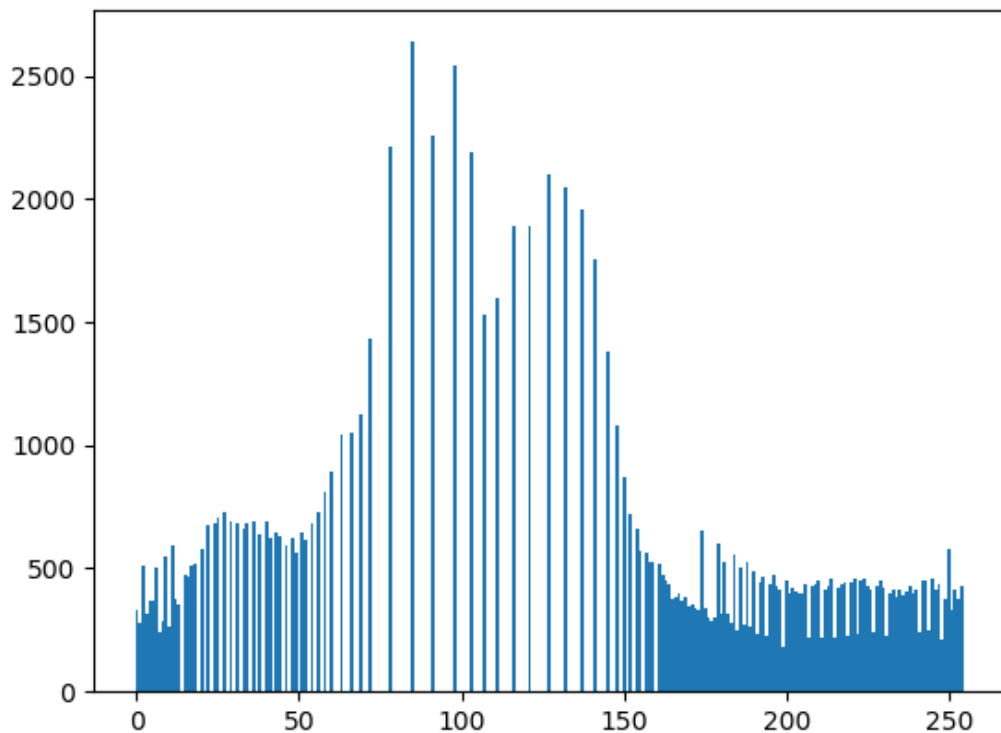
(2) Equalize the histogram





(3) Equalize the histogram again





The result is the same as (2)

As proofed in [1.1](#), after equalized, the distribution is approximately uniform distribution. Then applied equalization again, it is still approximately uniform distribution, remained unchanged.

(4) discuss how you implement the histogram equalization operation in details

The steps are as follows

1. travel the image and count the number of pixels in different value (PDF)
2. compute the CDF function of the image
3. display the histogram and save them
4. use the transform function
 
$$T(r_k) = (L - 1) \sum_{j=0}^k \frac{n_j}{MN} = \frac{L-1}{MN} \sum_{j=0}^k n_j, k = 0, 1, \dots, L - 1$$
5. display the histogram and the image after equalization and save them
6. equalize the histogram again, and display the histogram and the image, save them

## 2.2

(1) Smooth your input image with 3\*3, 5\*5 and 7\*7 averaging filters respectively.

3\*3



5\*5



7\*7



(2) Sharpen your input image with a 3\*3 Laplacian filter. In addition, briefly discuss why Laplacian filter can be used for sharpening.



Laplacian filter is like

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

it is somehow perform difference in the image in both x direction and y direction, so it can detect the extent how gray value changes in the x and y direction and used for sharpening.

(3) Perform high-boost filtering (i.e.,  $g(x, y) = f(x, y) + k * g_{mask}(x, y)$ ). Choose a  $k$  (the weight in Eq. (3.6-9)) as you see fit. Write down the selected  $k$  and paste your result on the report.

original image



$k=1$



$k=2$



$k=3$



$k=2$  and  $k=1$  are suitable for sharpening.

(4) Detailedly discuss how you implement the spatial filtering operation

The steps are as follows

1. compute the kernel according the size and mode(smooth or sharpen or high-boost)



2. transfer the parameters to filter2d
3. we compute the locations of a pixel to be filtered and multiply them by corresponding elements of filter
  1. if the position to be multiplied is out of index, just take it as zero
  2. else multiply the value and the corresponding element of filter
4. the sum of multiplication is the result of filtering of the pixel
5. do step 3,4 for all pixels