

Computer Vision HW3

R09922093 楊子萱

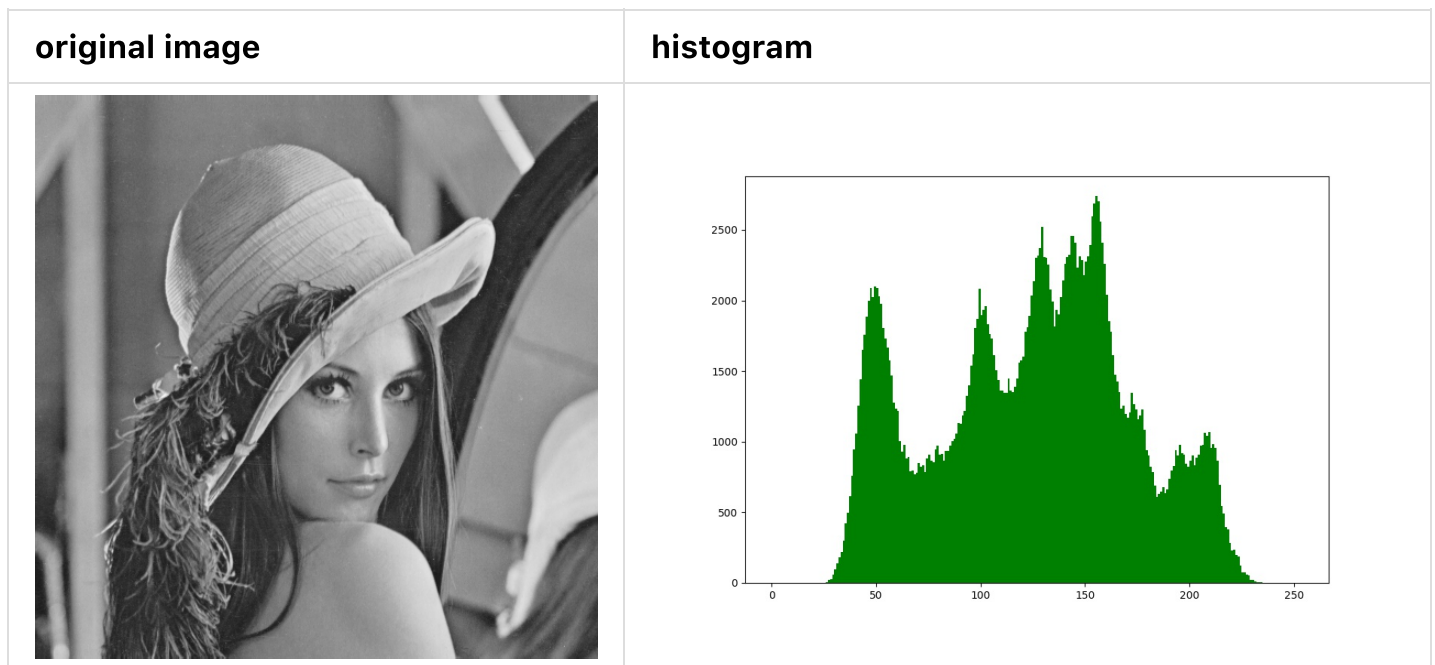
In this homework, I use python3 and import cv2 to read and write images. To run my code, use command line and enter python3 hw2.py (<http://hw2.py>) .

For drawing the histogram, I import matplotlib.pyplot. Read the pixel value for each image pixel and store them in a list then use plot histogram to draw the list out with range from 0 to 255.

```
def plothis(image, color, filename):  
    resultlist= []  
    for i in range (image.shape[0]):  
        for j in range(image.shape[1]):  
            resultlist.append(image[i, j])  
    plt.figure(figsize=(10,7))  
    plt.hist(resultlist, range(0,255), color = color)  
    # plt.show()  
    plt.savefig(filename)
```

(a) original image and its histogram

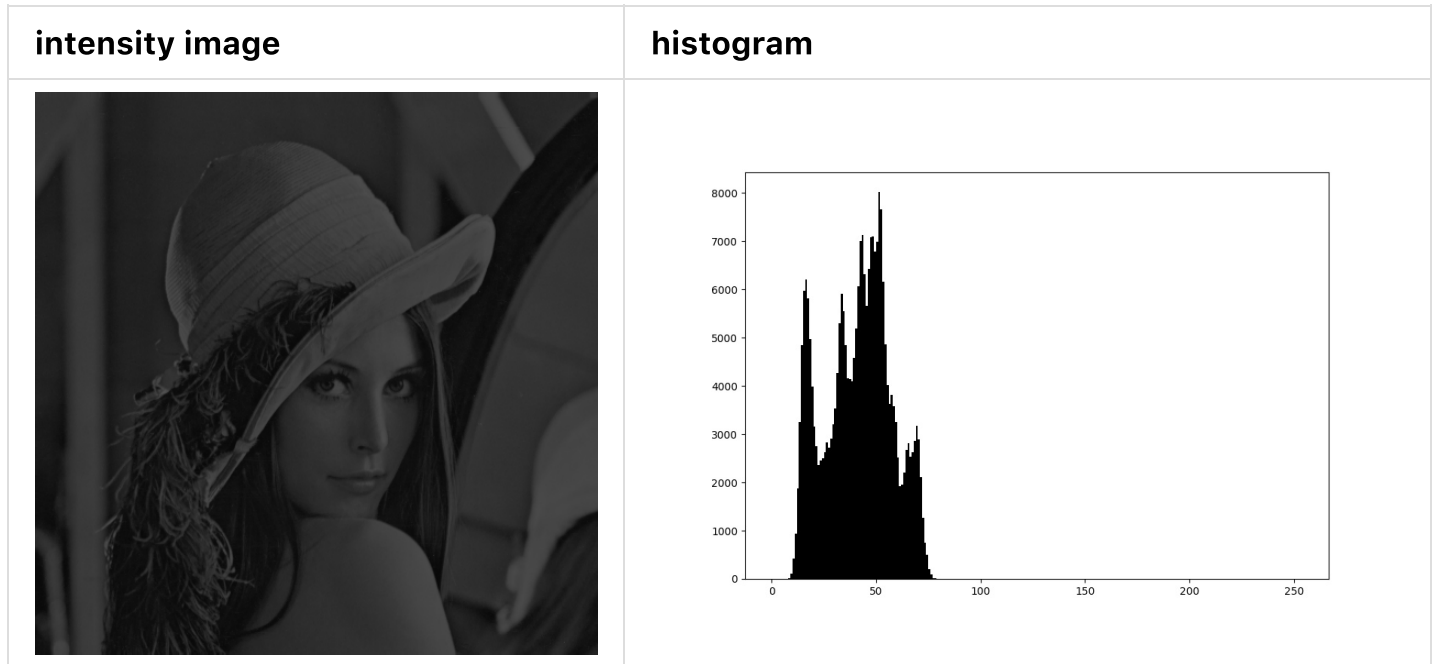
I use cv2.imread to read the image with gray scale and cv2.imwrite to save the image.



(b) image with intensity divided by 3 and its histogram

I use the image read from part(a) and divide each image pixel value with 3 to get the result.

```
def intensity(image):
    image1 = image.copy()
    # new = lena.copy()
    for i in range (image1.shape[0]):
        for j in range(image1.shape[1]):
            image1[i, j] = image1[i, j] // 3
    cv2.imwrite("intensity.bmp",image1)
    return image1
```



(c) image after applying histogram equalization to (b) and its histogram

I use the algorithm mentioned in the class ppt as following to get the result image from part (b)

- histogram equalization histogram linearization

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

- $k = 0, 1, \dots, 255$, n_j : number of pixels with intensity j

- n : total number of pixels

- for every pixel if $I(im, i, j) = k$ then $I(imhe, i, j) = s_k$

```

def equalization(image):
    n = [0] * 256
    s = [0] * 256
    image2 = image.copy()
    for i in range(image2.shape[0]):
        for j in range(image2.shape[1]):
            n[image2[i, j]]+=1
    for i in range(256):
        for j in range(i+1):
            s[i] += n[j]/(image2.shape[0] * image2.shape[1])
        s[i] *= 255
    equal_im = np.empty([image2.shape[0],image2.shape[1]])
    for i in range(image2.shape[0]):
        for j in range(image2.shape[1]):
            equal_im[i, j] = s[image2[i, j]]
    cv2.imwrite("equalization.bmp", equal_im)
    return equal_im

```

equalization image



histogram

