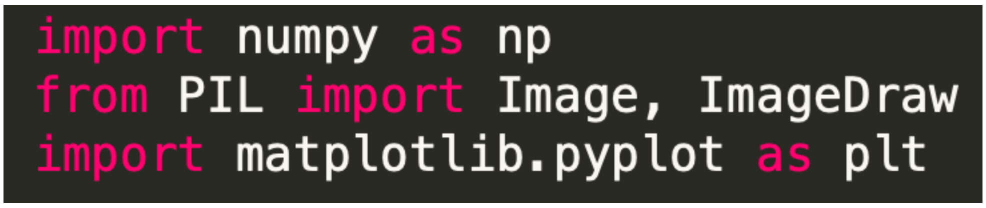
電腦視覺 Homework 3 – Histogram Equalization

資工碩一 r08922a04 林承德

Language : python 3.7

Library :



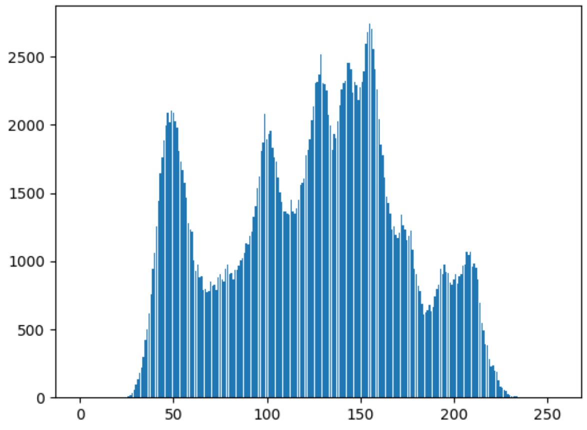
```
import numpy as np
from PIL import Image, ImageDraw
import matplotlib.pyplot as plt
```

Execution way: python3 hw3.py (please put lena.bmp at the same directory as hw3.py)

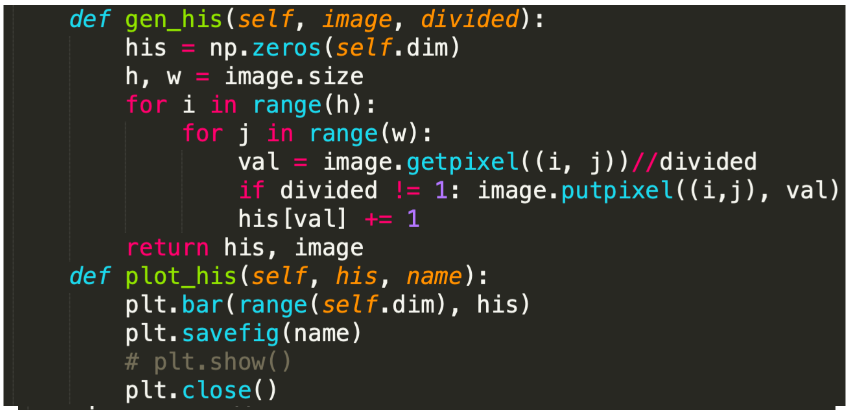
Problems:

Problem(a) generate original image and its histogram

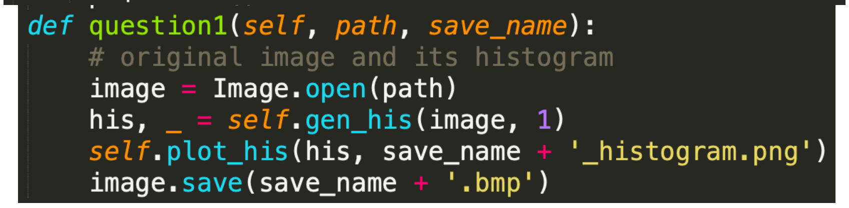
Result



Code



```
    def gen_his(self, image, divided):
        his = np.zeros(self.dim)
        h, w = image.size
        for i in range(h):
            for j in range(w):
                val = image.getpixel((i, j))//divided
                if divided != 1: image.putpixel((i,j), val)
                his[val] += 1
        return his, image
    def plot_his(self, his, name):
        plt.bar(range(self.dim), his)
        plt.savefig(name)
        # plt.show()
        plt.close()
```



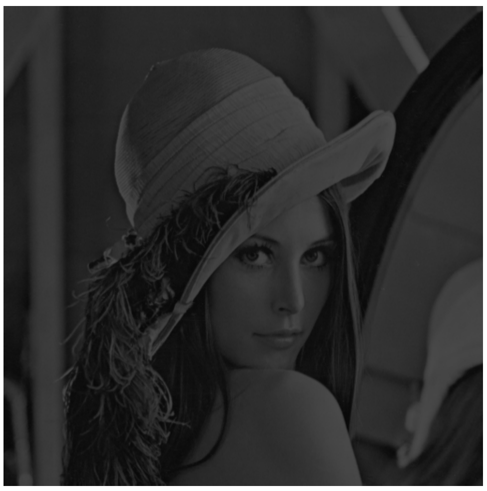
```
def question1(self, path, save_name):
    # original image and its histogram
    image = Image.open(path)
    his, _ = self.gen_his(image, 1)
    self.plot_his(his, save_name + '_histogram.png')
    image.save(save_name + '.bmp')
```

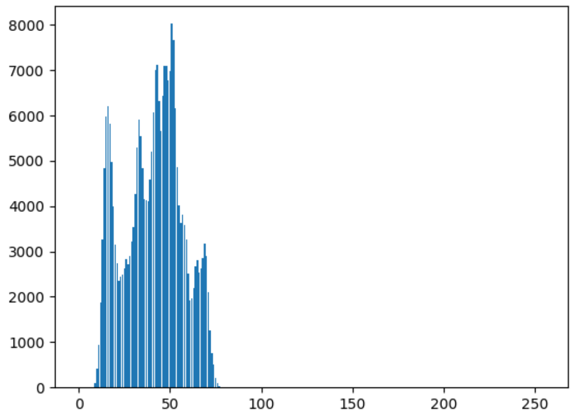
Description:

For the origin image, I can open the image easily, and go through every pixel with two level for-loop, then use an array to remember how many pixels has this intensity for every possible intensities. Finally, we could get everything I need for plot the histogram of origin image.

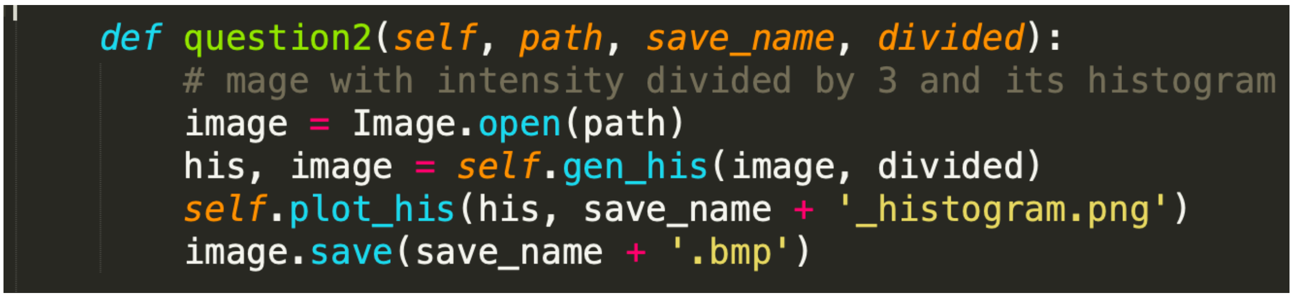
Problem(b) image with intensity divided by 3 and its histogram

Result





Code <class function I call here is same as previous page>



```
def question2(self, path, save_name, divided):
    # mage with intensity divided by 3 and its histogram
    image = Image.open(path)
    his, image = self.gen_his(image, divided)
    self.plot_his(his, save_name + '_histogram.png')
    image.save(save_name + '.bmp')
```

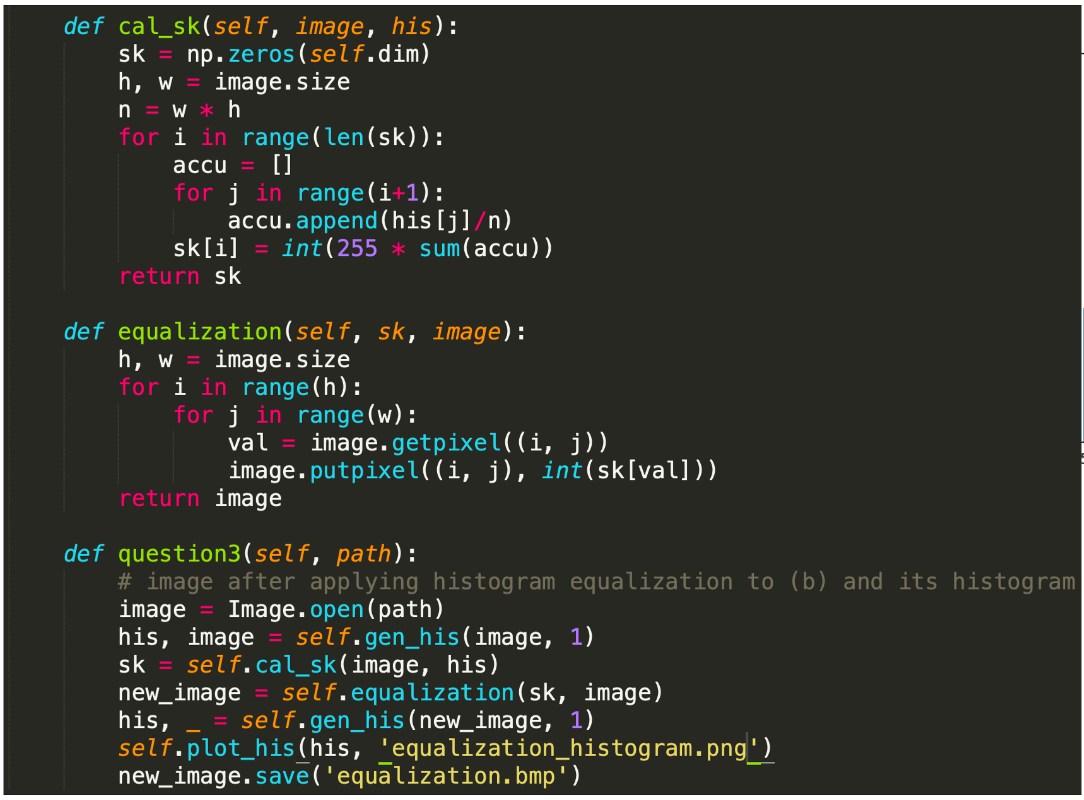
Description

For this case, argument divided is equal to 3, and I will divide the intensity by 3 before I calculate the histogram information, then use this value to replace the origin pixel. Finally, I save the histogram and image under this condition.

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

Result

Code



```python
    def cal_sk(self, image, his):
        sk = np.zeros(self.dim)
        h, w = image.size
        n = w * h
        for i in range(len(sk)):
            accu = []
            for j in range(i+1):
                accu.append(his[j]/n)
            sk[i] = int(255 * sum(accu))
        return sk

    def equalization(self, sk, image):
        h, w = image.size
        for i in range(h):
            for j in range(w):
                val = image.getpixel((i, j))
                image.putpixel((i, j), int(sk[val]))
        return image

    def question3(self, path):
        # image after applying histogram equalization to (b) and its histogram
        image = Image.open(path)
        his, image = self.gen_his(image, 1)
        sk = self.cal_sk(image, his)
        new_image = self.equalization(sk, image)
        his, _ = self.gen_his(new_image, 1)
        self.plot_his(his, 'equalization_histogram.png')
        new_image.save('equalization.bmp')
```

Description

First of all, given $s_k = 255\sum_{j=0}^{k}\frac{n_j}{n}$, I will calculate the histogram information by the image I want to do histogram equalization. And use such information to find n_j, then calculate s_k for every possible k, then as the algorithm tell us: for every pixel if $I(im, i, j) = k$ then replace this pixel's value with s_k. Finally, I use this new image to calculate the histogram information and plot it like what I did before.