

Computer Vision hw3

B05902050

黃子源

I use PIL to complete the homework, and use matplotlib to draw the histogram.
In my program, I use function `getpixel()` and `putpixel()` to get the value of every pixel, and use `pyplt.bar()` to draw the histogram.

Histogram Equalization:

First, I create a new image `img2` whose pixel value is one third of `lena.bmp`, and record the number of value in a list named `histogram`.

After that, list named `new` is used to record the new pixel value after histogram

equalization, which $new[k] = 255 * \sum_{j=0}^k \frac{n_j}{n}$, n_j : number of pixels with intensity j , n :

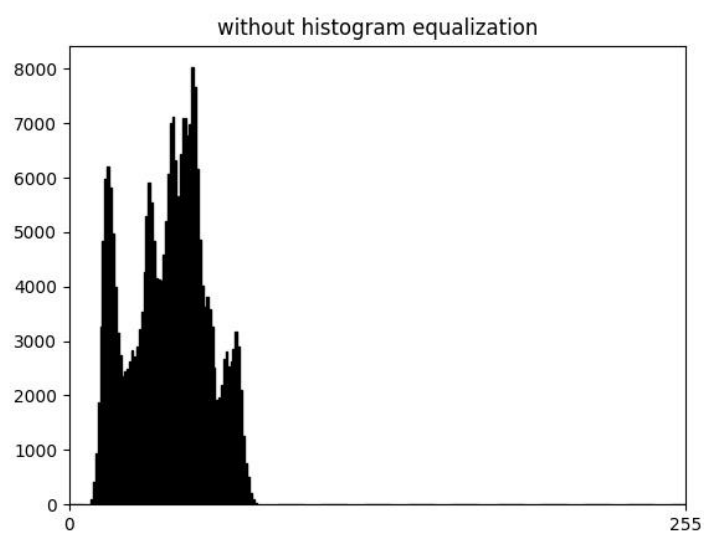
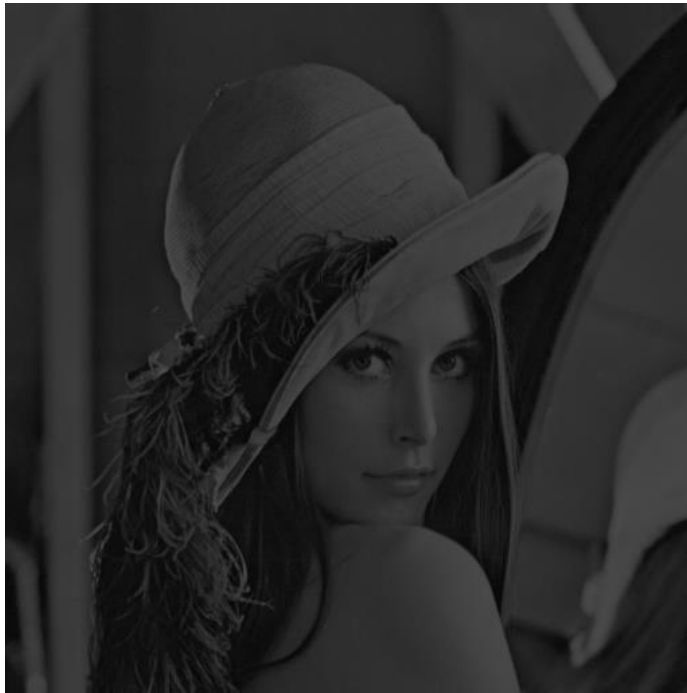
total number of pixels.

Last, we transform the intensity j in `img2` to `new[j]` then the histogram equalization is done.

Principal code fragment:

```
5  img = (Image.open("lena.bmp")).convert("L")
6  w, h = img.size
7  img2 = Image.new("L", (w,h))
8  img3 = Image.new("L", (w,h))
9  histogram = [0 for i in range(256)]
10 histogram2 = [0 for i in range(256)]
11 xx = np.arange(256)
12 xxx = np.arange(256)
13 new = [0 for i in range(256)]
14 sums = 0
15 for i in range(w):
16     for j in range(h):
17         x = int(img.getpixel((i,j))/3)
18         histogram[x] += 1
19         img2.putpixel((i, j), x)
20         sums += 1
21 now = 0
22 for i in range(256):
23     now += histogram[i]
24     #print(histogram[i])
25     new[i] = int(255*now/sums)
26 for i in range(w):
27     for j in range(h):
28         img3.putpixel((i,j), new[img2.getpixel((i,j))])
29         histogram2[new[img2.getpixel((i,j))]] += 1
30 for i in range(256):
31     print(histogram2[i])
32 img3.save("lena3.bmp")
33 img2.save("lena2.bmp")
34 plt.bar(xx,histogram,facecolor = 'black',edgecolor = 'black')
35 plt.xlim(0,255)
36 plt.xticks(np.linspace(0,255,2))
37 plt.title('without histogram equalization')
38 plt.savefig('histogram_1.jpg')
39 plt.bar(xxx,histogram2,facecolor = 'black',edgecolor = 'black')
40 plt.xlim(0,255)
41 plt.xticks(np.linspace(0,255,2))
42 plt.title('histogram equalization')
```

The image and histogram before equalization:



The image and histogram after equalization:

