Computer Vision Homework 3

B03902042 宋子維

Description

In this homework, we are asked to implement the programs do histogram equalization. The transformation method is

$$s_k = 255 \sum_{j=0}^{k} \frac{n_j}{n}, \text{ where } k = 0, 1, \cdots, 255$$

n_j : number of pixels with intensity j

n : total number of pixels

$I(imhe, i, j) = s_k$ for each pixel (i, j), where $I(im, i, j) = k$

Programming

I use python to implement the algorithm. There is one python program, namely, histogram-equalization.py, where I use **pillow** to process basic image I/O. In the program, there are some I/O functions:

1. `PIL.Image.open(img)`: load the image `img` and return a pillow **Image** object.
2. `pix = Image.load()`: return the image access object of **Image** object to `pix`, which offers us to use `pix[x, y]` to access the pixel value at position `(x, y)`.
3. `PIL.Image.new(mode, size)`: create a new image with given mode and size, and return a piilow **Image** object.
4. `Image.size`: pair (width, height) of **Image** object.

Write a program to do histogram equalization

Run `python3.5 histogram-equalization.py $IMG_IN $IMG_OUT $HISTO`, and the program will do histogram-equalization

on **IMG_IN**, save the new image as **IMG_OUT** with **BMP** format and generate the histogram of **IMG_OUT**, called **HISTO**.

Figure 1-1: Do histogram-equalization on lena.bmp

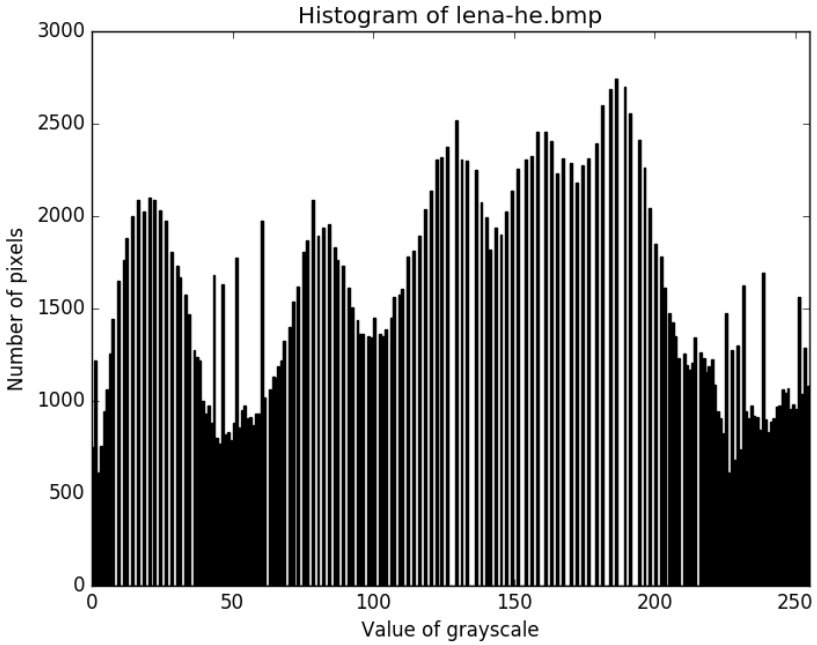

Figure 1-2: The histogram after doing histogram equalization

Algorithm

First, calculate the histogram of **IMG_IN**. To do this, iterate all pixels (x, y) and increase the histogram value at i by 1, where i is the intensity of (x, y).

```
N = 256         # grayscale value from 0 to 255
sum = [0] * N   # histogram of IMG_IN
for x in range(width):
    for y in range(height):
        sum[pix[x, y]] += 1
```

where `pix` is the image access object of **IMG_IN**. and `width` and `height` are weight and height of **IMG_IN**.

Second, calculate the cumulative histogram. To do this, simply iterate i from 1 to 255, and the cumulative histogram value at i is the sum of the cumulative histogram value at $i-1$ and the histogram value at i and the cumulative histogram value at 0 is equal to the histogram value at 0.

```
s = [0] * N     # cumulative histogram of IMG_IN
s[0] = sum[0]
for i in range(1, N):
    s[i] = s[i-1] + sum[i]
```

Finally, assign the intensity after doing histogram equalization to the new pixel, which follows the rules in the description and calculate the histogram of new image.

```
histogram = [0] * N     # histogram of IMG_OUT
for x in range(width):
    for y in range(height):
        pix_he[x, y] = int(255 * s[pix[x, y]] / size)
        histogram[pix_he[x, y]] += 1
```

where `pix_he` is the image access object of the new **IMG_OUT**, `size` is the total number of pixels and `histogram` is the histogram of new image.

As for histogram plotting, `matplotlib.pyplot.bar(left, height, color)` can help us. `left` is sequence of scalars, which are the x coordinates , `height` is also the sequence of scalars, which are the heights of the bars, and `color` is the color of the bars.

```
ind = range(N)
plt.bat(ind, histogram, color='black')
```

where `ind` is the list from 0 to 255, `histogram` is the list with the statistic data mentioned above.