

Computer Vision I

Homework 3 - Histogram Equalization

B04902090 資工四 施長元

Usage of the full code:

```
python main.py [Image_path]
```

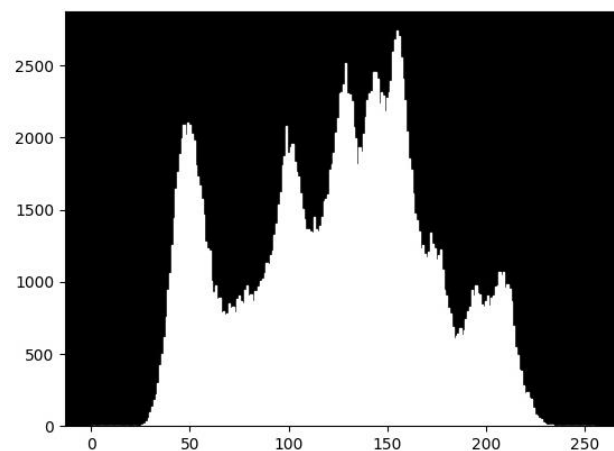
After the code exit, the output file will be in the same directory with the main.py

Environment: Python3.6 on Windows Linux Subsystem (Ubuntu 16.04)

Contents:

Write a program to do histogram equalization

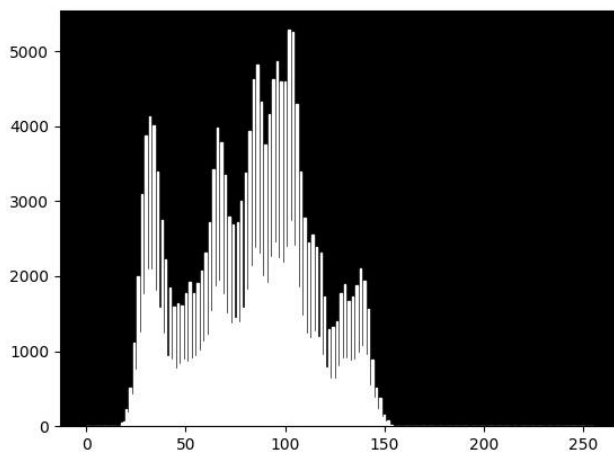
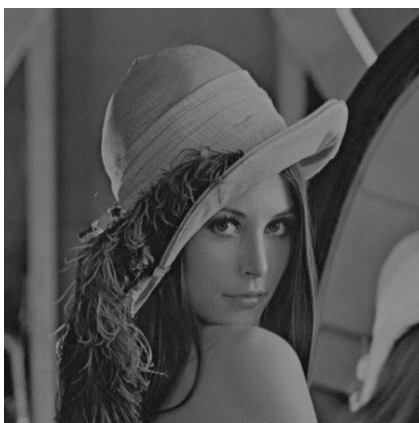
- Original Image



- Generate a target Darken Image

```
for i in range(img.shape[0]):  
    for j in range(img.shape[1]):  
        # Darken img by 2/3  
        dark_img[i, j] = 2 * (img[i, j] / 3)  
        his[0][img[i, j]] += 1  
        his[1][dark_img[i, j]] += 1
```

將每個 Pixel 乘上(2/3), 並產生 histograms.



- Result Image

```
count, min_n = 0, 0
for i in range(256):
    if min_n == 0 and his[1][i] != 0: min_n = i
    count += his[1][i]
    cdf[i] = count

result = img.copy()
n_pixels = float((img.shape[0] * img.shape[1]) - cdf[min_n])
for i in range(img.shape[1]):
    for j in range(img.shape[0]):
        y = round(255.0 * float(cdf[dark_img[i, j]] - cdf[min_n]) / n_pixels)
        result[i, j] = y
        his[2][result[i, j]] += 1
```

首先計算 CDF (Cumulative Distribution Function) ，並標註 cdf 中最小的值，用以計算下式↓

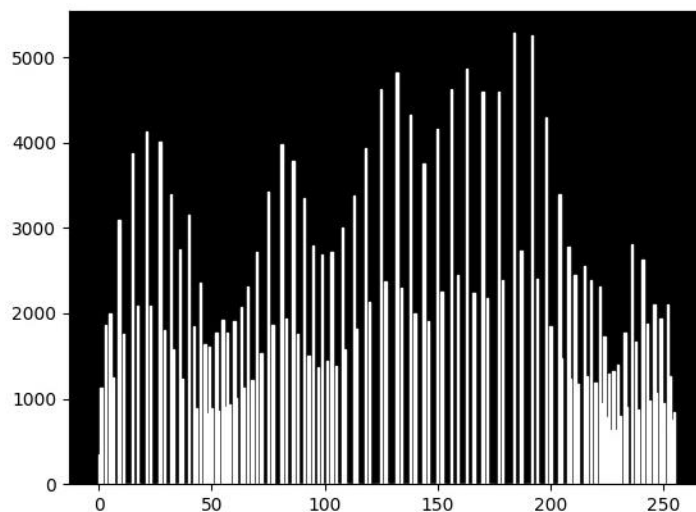
$$s_k = 255 \sum_{j=0}^k \frac{n_j}{n}$$

的分子。

接著套用公式，生成結果的點，並標在一個 numpy 矩陣上，同時生成 Histogram 。最終的結果如左圖，Histogram 如下圖。



← Result



← Histogram of the result