Computer Vision HW3, Histogram Equalization Report

tags: **NTU CS Computer Vision Writeup Report**

NTU CSIE, R08922024, Alfons Hwu
Prequisites and env as the following

```
Ubuntu WSL for windows with jupyter notebook
Python3.6.7
OpenCV for image IO
Matplotlib for displaying image
```

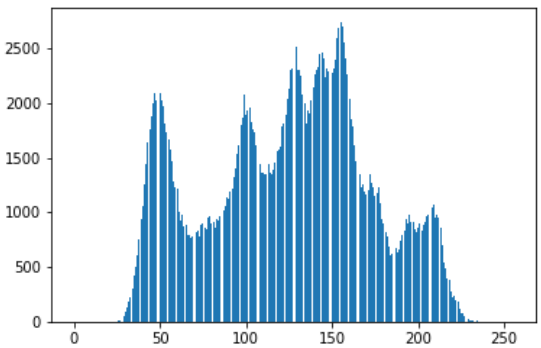
a, original image and its histogram

```
def img_hist(img_in, name):
    hist = [0 for i in range(256)]

    row, col  = img_in.shape
    for i in range(0, row):
        for j in range(0, col):
            hist[img_in[i, j]] += 1

    return hist
```

Iterate through the image pixel by pixel and calculate the statistical data.
Time complexity: $O(MN)$

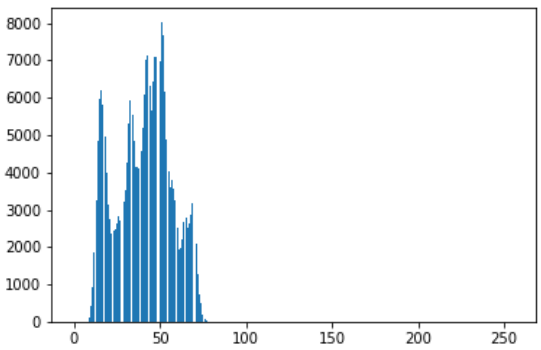


b, image with intensity divided by 3 and its histogram

Just use

```
def div3(img_in):
    return img_in // 3
```

And the histogram function mentioned above
Time complexity: $O(MN)$



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

From wiki here
(https://zh.wikipedia.org/wiki/%E7%9B%B4%E6%96%B9%E5%9B%BE%E5%9D%87%E8%A1%A1%E5%8C%96)
We can equalize the pixel as the following formula

$$h(v) = round(\frac{cdf(v) - cdf_{min}}{(M \times N) - cdf_{min}} \times (L - 1))$$

MN is the dimension of the image, cdf is the **cumulative distribution function** over v, the grayscale.

So with the following code:

```
def histogram_eq(img_in, hist):
    row, col  = img_in.shape
    cdf_list = [0 for i in range(256)]
    cdf = 0.0
    max_value = 0
    min_value = 1 << 31

    # calculating the distribution function over v
    for i in range(0, len(hist)):
        if hist[i]:
            max_value = max(max_value, i)
            min_value = min(min_value, i)
            cdf += hist[i]
            cdf_list[i] = cdf

    # transform to perform histogram equalization
    for i in range(0, row):
        for j in range(0, col):
            img_in[i, j] =  int((cdf_list[img_in[i, j]] \
            - cdf_list[min_value]) \
                           /(row * col - cdf_list[min_value]) \
                           * (0xff - 1)) # 0 - 255 for grayscale

    return img_in
```

Time complexity: $O(MN)$