

Computer Vision

Homework 3 Report

林義聖
B03902048

October 11, 2016

1 Introduction

I use *Python* as my programming language and *Pillow* as my Image Library. Also, I use *matplotlib* to draw the histogram.

2 Histogram Equalization

I use the following steps to equalize *lena.bmp*.

1. Read *lena.bmp* in as Python *list*.
2. Loop through every pixels, accumulate its intensity.
3. Calculate the CDF of its intensity.
4. Equalize the image using CDF.

Listing 1: Equalize the image

```
1 class HistEqualization:
2     def __init__(self, data, img_size):
3         self.data = data
4         self.img_size = img_size
5         self.hist_lst = self.calc_histogram()
6         self.cdf_lst = self.calc_cdf()
7
8     # accumulate intensity to get histogram
9     def calc_histogram(self):
10        lst = [0] * 256
11        for x in range(self.img_size):
12            lst[ self.data[x] ] += 1
13        return lst
14
15    # from hist_lst, calculate CDF
16    def calc_cdf(self):
17        lst = [0] * 256
18        for x in range(1,256):
```

```

19         lst[x] = lst[x-1] + self.hist_lst[x]
20     return lst
21
22     # equalize the data
23     def equalized(self):
24         cdf_min = 0
25         lst = [0] * self.img_size
26         intensity = [0] * 256
27
28         for x in range(256):
29             if self.cdf_lst[x] > 0:
30                 cdf_min = self.cdf_lst[x]
31                 break
32
33         for x in range(256):
34             if self.cdf_lst[x] < cdf_min:
35                 intensity[x] = 0
36             else:
37                 intensity[x] = int(round( (float(self.cdf_lst[x] - cdf_min) / float(self.
38                                         img_size - cdf_min)) * 255.0 ))
39
40         for x in range(self.img_size):
41             lst[x] = intensity[ self.data[x] ]
42
43     return lst

```

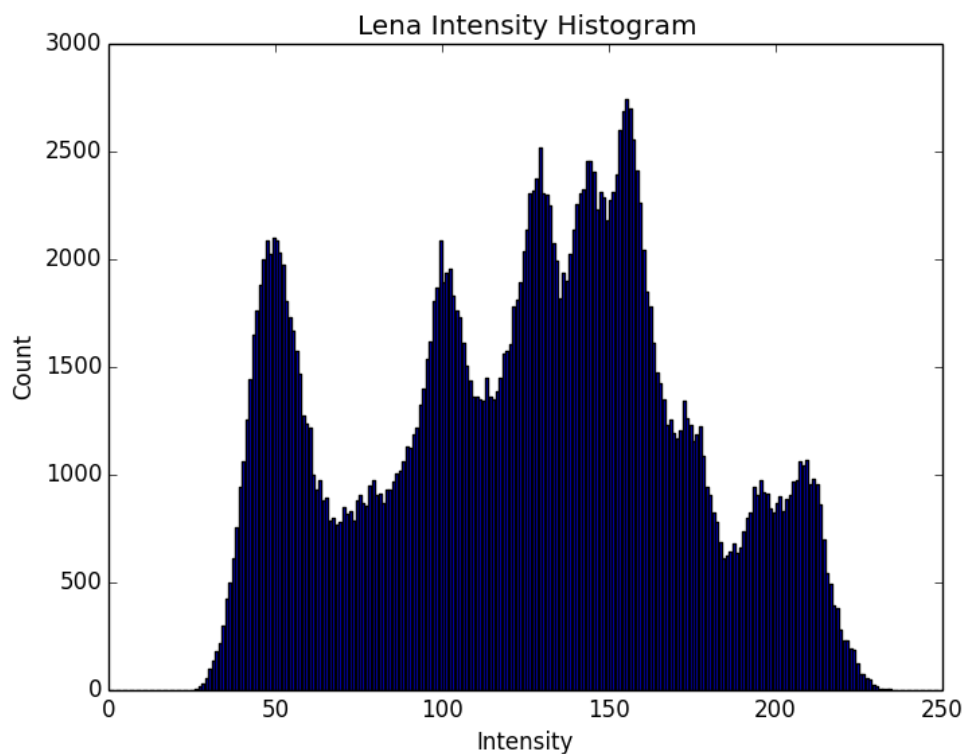


Figure 1: original lena histogram

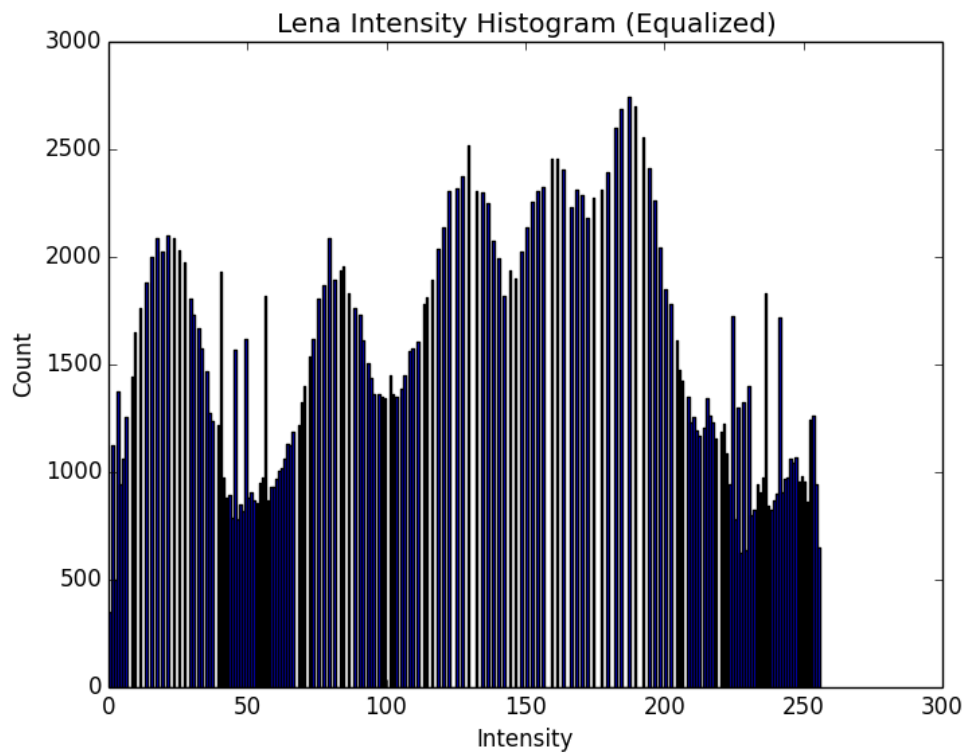


Figure 2: equalized histogram



(a) original picture



(b) equalized picture

Figure 3: Comparison between two pictures

3 HSV Equalization (additional)

Additionally, I write the other program to equalize the image. I change the image data from *RGB* mode to *HSV* mode and use the same algorithm but apply on *S* - *saturation* and *V* - *lightness*. The program is longer than the previous one. So I won't paste it here, just show the result.



(a) original picture



(b) equalized picture

Figure 4: Comparison between two colorful pictures

4 How to Use

There are 2 programs,

1. *histogram-equalization.py*
2. *hsv-equalization.py* (additional)

You just need to enter command in this format: "*program [input image name] [output image name]*" to use it. For example, `./histogram-equalization.py lena.bmp output.bmp`.