

## Assignment 1 Computer Vision

### 1. WHAT IS IMAGE HISTOGRAM?

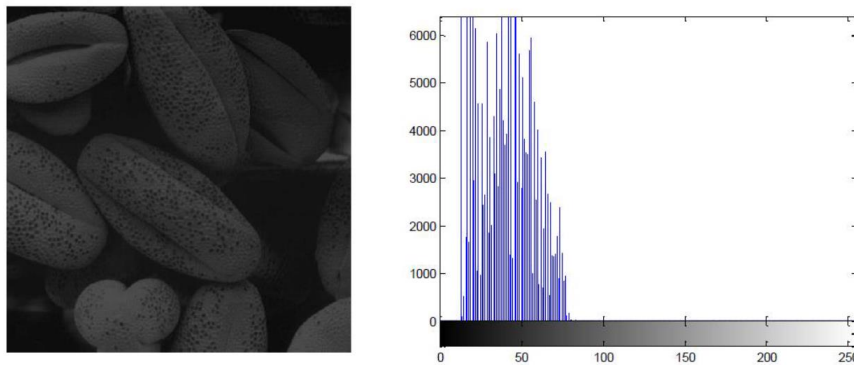
Image Histogram merupakan histogram yang menggambarkan penyebaran nilai-nilai intensitas *pixel* dari suatu gambar. Histogram ini juga dapat menunjukkan banyak hal mengenai kecerahan (*brightness*) dan kontras (*Contrast*) dari suatu gambar.

#### HOW TO TELL WHETHER AN IMAGE HAVE A GOOD CONTRAST FROM ITS HISTOGRAM?

Kontras merupakan sebaran terang (*lightness*) dan gelap (*darkness*) pada suatu gambar. Terdapat 4 tipe gambar berdasarkan kekontrasannya:

##### 1. Gambar gelap (*under exposed*)

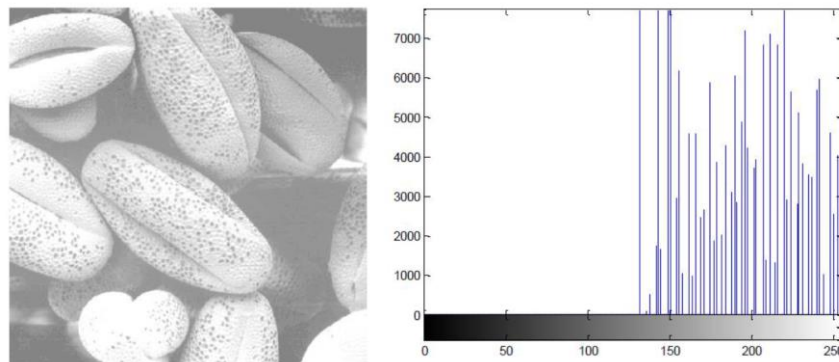
Pada gambar gelap, komponen histogramnya terkonsentrasi pada nilai-nilai *grayscale* yang rendah (menumpuk pada sisi histogram sebelah kiri)



Sumber: **Image Processing** By Dr. Jagadish Nayak ,BITS Pilani, Dubai Campus

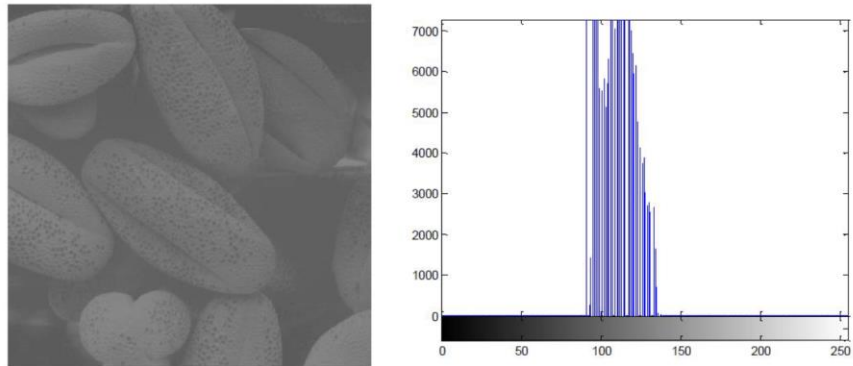
##### 2. Gambar terang (*over exposed*)

Pada gambar terang, komponen histogramnya terkonsentrasi pada nilai-nilai *grayscale* yang tinggi (menumpuk pada sisi histogram sebelah kanan).



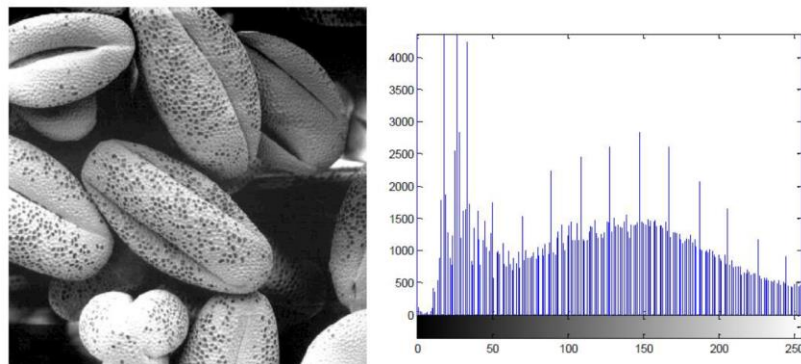
3. Gambar kontras rendah (*low contrast*)

Pada gambar *low contrast*, komponen histogramnya sempit dan terpusat di pertengahan nilai-nilai grayscale.

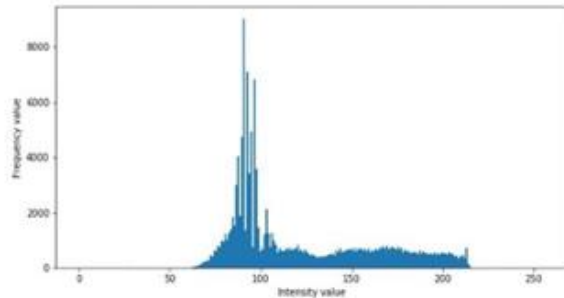


4. Gambar kontras tinggi (*high contrast*)

Pada gambar dengan *high contrast* komponen histogramnya tersebar merata di seluruh nilai-nilai grayscale dan distribusi pixel-pixel tidak terlalu jauh dari uniform, dan garis-garis vertikalnya tidak banyak yang lebih tinggi dari garis yang lain.



2. A. LET SAY YOU HAVE A HISTOGRAM OF A LOW CONTRAST IMAGE AS IT IS SHOWN IN THE FOLLOWING FIGURE, CAN YOU SKETCH A TRANSFORMATION THAT WILL LIKELY IMPROVE THE IMAGE CONTRAST.



Berdasarkan penjelasan saya diatas, *low contrast image* memiliki ciri-ciri komponen histogramnya terpusat pada suatu titik intensitas dimana pada gambar kita bisa lihat bahwa pusatnya berada pada kisaran intensitas 100. Untuk meningkatkan kontras gambar tersebut, kita perlu untuk melakukan yang namanya **histogram equalization**. Pada *histogram equalization* terjadi perataan histogram yaitu dengan mengubah derajat keabuan suatu pixel ( $r$ ) dengan derajat keabuan yang baru ( $s$ ) dengan suatu fungsi transformasi  $T$ , yang dalam hal ini  $s = T(r)$ . Dua sifat yang dipertahankan pada transformasi ini:

1. Nilai  $s$  merupakan pemetaan 1 ke 1 dari  $r$ . Ini berarti  $r$  dapat diperoleh kembali dari  $s$  dengan transformasi invers:  $r = T^{-1}(s)$ ,  $0 < s < 1$
2. Untuk  $0 < r_i < 1$ , maka  $0 < T(r) < 1$ .

Rumus yang digunakan untuk menghitung *histogram equalization*:

$$P_r(r_k) = \frac{n_k}{n} \text{ dalam hal ini } r_k = \frac{k}{L-1}, 0 \leq k \leq L-1$$

Dimana:

$n_k$  = nilai piksel pada derajat keabuan  $k$

$n$  = jumlah seluruh piksel pada gambar

Dari rumus ini kita dapat melihat bahwa derajat keabuan ( $k$ ) dinormalkan terhadap derajat keabuan ( $L-1$ ). Nilai  $r_k = 0$  menyatakan hitam, dan  $r_k = 1$  menyatakan putih dalam skala keabuan yang didefinisikan.

Rumus *histogram equalization* pada gambar dengan skala keabuan k bit:

$$K_o = \text{round} \left( \frac{C_i \cdot (2^k - 1)}{w \cdot h} \right)$$

Dimana

$C_i$  = Distribusi kumulatif dari nilai skala keabuan ke  $i$  dari citra asli

$\text{round}$  = Fungsi pembulatan ke bilangan yang terdekat

$K_o$  = Nilai keabuan hasil histogram equalization

$w$  = Lebar citra

$h$  = Tinggi citra

B. WRITE A PROGRAM (IN PYTHON NOTEBOOK) THAT PERFORMS HISTOGRAM EQUALIZATION ON A GRAYSCALE IMAGE. YOUR PROGRAM SHOULD : (1) COMPUTE THE HISTOGRAM OF THE INPUT IMAGE; (2) COMPUTE THE HISTOGRAM OF THE EQUALIZED IMAGE; (4) DISPLAY (AND PRINT) THE ORIGINAL AND EQUALIZED IMAGES AS WELL AS THEIR CORRESPONDING HISTOGRAMS, ALL IN ONE FIGURE. YOU MAY USE OPENCV TO QUICKLY IMPLEMENT THE PROGRAM.

Link file ipynb: [https://binusianorg-my.sharepoint.com/personal/i\\_saputra\\_binus\\_ac\\_id/\\_layouts/15/guestaccess.aspx?folderid=05053c2fe19d548048592dd2357f7fa5b&authkey=AeNp0hLNoFadNUKjckObmSc&e=HbRfMI](https://binusianorg-my.sharepoint.com/personal/i_saputra_binus_ac_id/_layouts/15/guestaccess.aspx?folderid=05053c2fe19d548048592dd2357f7fa5b&authkey=AeNp0hLNoFadNUKjckObmSc&e=HbRfMI)

Berikut adalah screenshot dari program saya:

### Assignment 1 Computer Vision

I Putu Krisna Dharma Saputra/2301924353

```
In [1]: # Import Libraries
import numpy as np
import cv2
from matplotlib import pyplot as plt

In [2]: # read image
original_image = cv2.imread("man.png")

In [3]: # change the image into an 8-bit grayscale image
grayimg = cv2.cvtColor(original_image, cv2.COLOR_BGR2GRAY)

In [4]: # height and width initialization
height = grayimg.shape[0]
width = grayimg.shape[1]

In [5]: # counter initialization
image_counter = np.zeros(256, dtype=int)

In [6]: # Looping to compute the histogram of the input image
for i in range(height):
    for j in range(width):
        image_counter[grayimg[i][j]] += 1

In [7]: # Perform histogram equalization
equ_image = cv2.equalizeHist(grayimg)

In [8]: # equalization counter initialization
equ_counter = np.zeros(256, dtype=int)

In [9]: # Looping to compute the histogram of the equalized image
for i in range(height):
    for j in range(width):
        equ_counter[equ_image[i][j]] += 1
```

```
In [10]: # Result initialization
result = np.hstack([grayimg, equ_image])
```

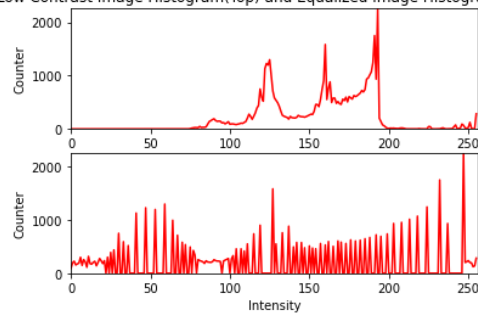
```
In [24]: # DISPLAY (AND PRINT) THE ORIGINAL AND EQUALIZED IMAGES AS WELL AS THEIR CORRESPONDING HISTOGRAMS, ALL IN ONE FIGURE.
```

```
plt.subplot(2, 1, 1)
plt.plot(image_counter, 'r', label="Histogram")
plt.title("Low Contrast Image Histogram(Top) and Equalized Image Histogram(Down)")
plt.xlabel("Intensity")
plt.ylabel("Counter")
plt.axis([0, 256, 0, image_counter.max()])

plt.subplot(2, 1, 2)
plt.plot(equ_counter, 'r', label="Histogram")
plt.xlabel("Intensity")
plt.ylabel("Counter")
plt.axis([0, 256, 0, equ_counter.max()])
plt.show()

plt.imshow(result, cmap='Greys_r')
plt.title("Before and After Equalization")
plt.show()
```

Low Contrast Image Histogram(Top) and Equalized Image Histogram(Down)



Before and After Equalization

