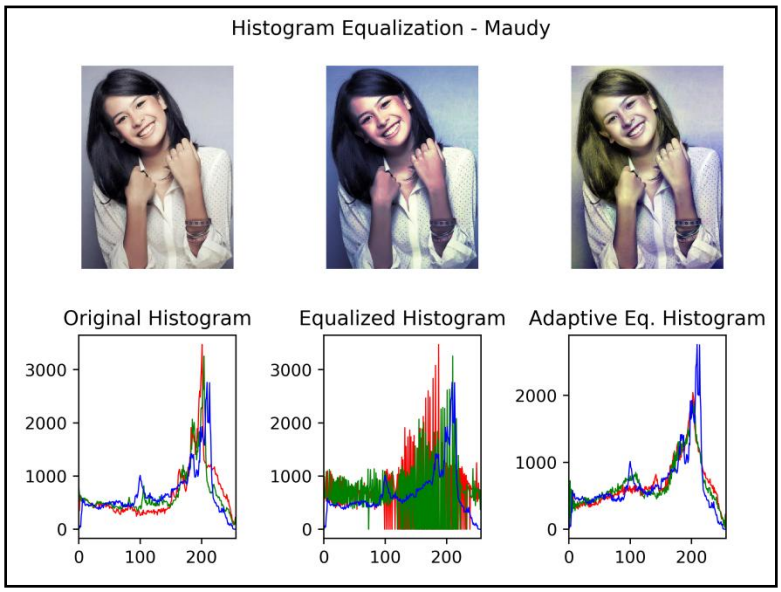
Tugas 2 - Histogram

IF5153 - Pemrosesan Pengelolaan Data Multimedia

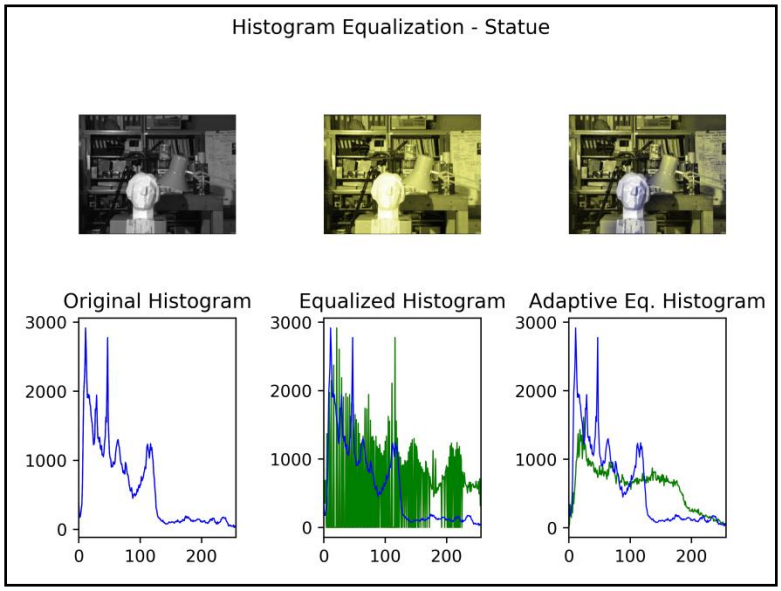
Nama : Mohamad Hanifan

NIM : 23518026

A. Histogram Equalization

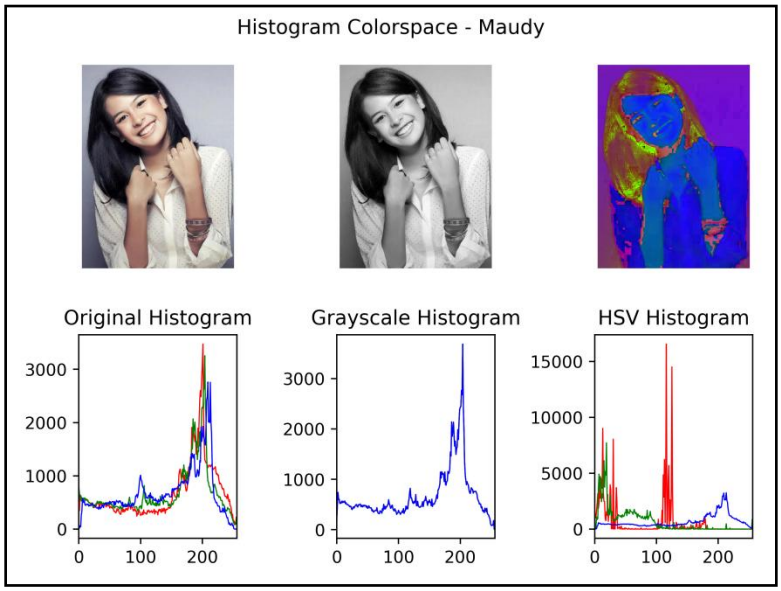


Gambar 1. Histogram Equalization Case 1

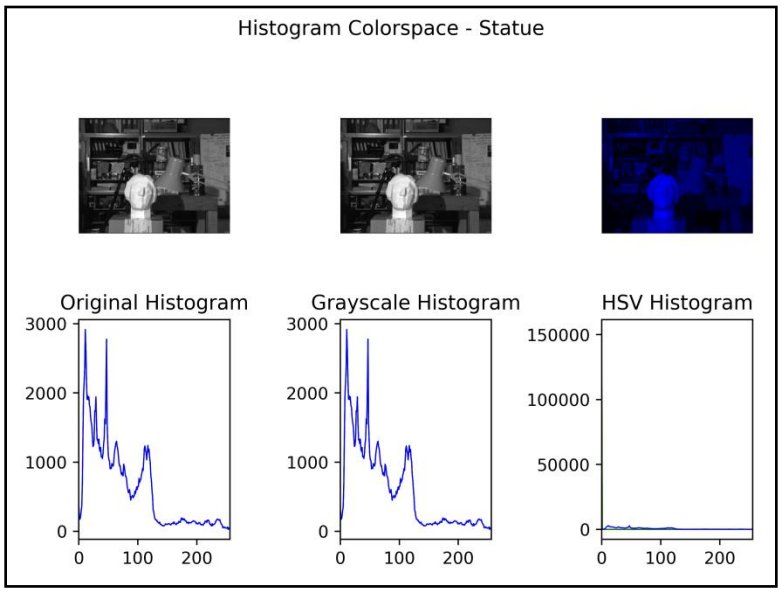


Gambar 2. Histogram Equalization Case 2

B. Comparation of Different Color Spaces Histogram



Gambar 3. Histogram of Different Color Spaces Case 1



Gambar 4. Histogram of Different Color Spaces Case 2

Pada gambar ***grayscale***, nilai R, G, dan B nya selalu sama.

Pada gambar **HSV**, garis merah merepresentasikan ***Hue***, garis hijau merepresentasikan ***Saturation***, dan garis biru merepresentasikan ***Value***.

C. Questions And Jawaban

- What is bins?

  Pada konteks histogram, *bin* (tunggal) adalah kelompok nilai. *Bin* direpresentasikan dalam range nilai. Misalnya jika *range* sebuah *bin* adalah [0 - 10], maka nilai dalam kisaran tersebut akan masuk ke *bin* ini.
  Pada histogram, umumnya digunakan 256 *bins,* artinya setiap bin memiliki *range* [x - x.99]. Sehingga setiap *bin* hanya mengandung piksel-piksel dengan nilai yang sama.

- What is the difference between histogram calculation in openCV and numpy ?

  Numpy memiliki 2 fungsi untuk menghitung histogram.
  - Pertama adalah **histogram**. Fungsi ini proses komputasinya cenderung lambat.
  - Kedua adalah **bincount**. Fungsi ini 10 kali lebih cepat dibandingkan fungsi histogram.

  Kedua fungsi pada numpy menghasilkan jumlah bins sebanyak 257.

  OpenCV memiliki fungsi **calcHist** yang 40 kali lebih cepat dibangdingkan fungsi histogram, dan menghasilkan bins sebanyak 256.

- Is there any correlation between contrast and the result of histogram equalization?

  Ada. Gambar yang gelap cenderung memiliki representasi histogram yang bernilai kecil (mendekati 0). Sedangkan gambar yang terang cenderung memiliki representasi histogram yang bernilai besar (mendekati 255).

- What is the difference between histogram equalization and adaptive histogram equalization?

  Pada ***histogram equalization*****,** setiap piksel di normalisasi terhadap seluruh piksel. Sehingga dapat menghilangkan detail dari gambar. Hal ini dapat dilihat pada Gambar 2. Wajah dari patung terlalu terang sehingga lekukan-lekukannya kurang jelas.

  Pada ***adaptive histogram equalization***, piksel-piksel yang berdekatan dikelompokkan terlebih dahulu. Kemudian setiap kelompok tersebut di normalisasi. Hal ini dilakukan untuk mencegah detail gambar menjadi hilang.

<u>Referensi</u>

1. https://docs.opencv.org/4.0.1/d1/db7/tutorial_py_histogram_begins.html
2. https://docs.opencv.org/4.0.1/d5/daf/tutorial_py_histogram_equalization.html

D. Source Code

```
import cv2
from matplotlib import pyplot as plt

# Declare Equalize Function
def equalizeHist(img):
    res = img.copy()
    for c in range(0, 2):
       res[:,:,c] = cv2.equalizeHist(res[:,:,c])
    return res

def adaptiveEqualizeHist(img):
    res = img.copy()
    clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8,8))
    for c in range(0, 2):
        res[:,:,c] = clahe.apply(res[:,:,c])
    return res

# Declare Histogram Function
def plotHist(img):
    color = ('r','g','b')
    for i,col in enumerate(color):
        histr = cv2.calcHist([img],[i],None,[256],[0,256])
        plt.plot(histr, color = col, linewidth=0.7)
        plt.xlim([0,256])

# Load Image
mainTitle = 'Maudy'
img = plt.imread('./maudy.jpg') # image loaded using RGB color scheme

# Histogram Equalization
equ = equalizeHist(img)
ahe = adaptiveEqualizeHist(img)

plt.figure(1)
plt.subplot(231), plt.imshow(img), plt.axis('off'),
plt.subplot(232), plt.imshow(equ), plt.axis('off'),
plt.subplot(233), plt.imshow(ahe), plt.axis('off'),
plt.subplot(234), plotHist(img), plt.gca().set_title('Original Histogram')
plt.subplot(235), plotHist(equ), plt.gca().set_title('Equalized Histogram')
plt.subplot(236), plotHist(ahe), plt.gca().set_title('Adaptive Eq. Histogram')
plt.tight_layout()
plt.subplots_adjust(top=0.90)
plt.suptitle('Histogram Equalization - ' + mainTitle)
plt.savefig('Histogram Equalization - ' + mainTitle, dpi=500)
```

```
# Histogram In Multiple Colorspace
img_gray = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)
img_gray = cv2.cvtColor(img_gray, cv2.COLOR_GRAY2RGB) # convert back to rgb for
histogram analysis
img_hsv = cv2.cvtColor(img, cv2.COLOR_RGB2HSV)
plt.figure(2)
plt.subplot(231), plt.imshow(img), plt.axis('off'),
plt.subplot(232), plt.imshow(img_gray), plt.axis('off'),
plt.subplot(233), plt.imshow(img_hsv), plt.axis('off'),
plt.subplot(234), plotHist(img), plt.gca().set_title('Original Histogram')
plt.subplot(235), plotHist(img_gray), plt.gca().set_title('Grayscale Histogram')
plt.subplot(236), plotHist(img_hsv), plt.gca().set_title('HSV Histogram')
plt.tight_layout()
plt.subplots_adjust(top=0.90)
plt.suptitle('Histogram Colorspace - ' + mainTitle)
plt.savefig('Histogram Colorspace - ' + mainTitle, dpi=500)

plt.show()
```