



Lab #2

Image Enhancement (Pixel intensity Adjustment)

Math Operation

Histogram based Operation

Lab#2

Topics

2.1 Image Pixel Adjustment

2.2 Image Histogram Equalization

2.3 Image Histogram Matching

Libraries

- `import cv2`
- `import numpy as np`
- `from matplotlib import pyplot as plt`

“

2.1 Image Pixel Adjustment

2.1.1 Image Pixel Adjustment

(Linear Equation)

- 2.1.1 ให้สร้างภาพผลลัพธ์จากสมการเส้นตรง $g(x, y) = af(x, y) + b$
 - กำหนดให้ปรับค่า a , b เพื่อสร้างภาพอย่างน้อย 20 ภาพ
 - โดยให้ทำการคำนวณ Image Array ด้วยตนเอง
 - ห้ามใช้ฟังก์ชันจาก Library
 - เขียนภาพทั้งหมดลงไฟล์วิดีโอ เพื่อให้เห็นการเปลี่ยนแปลงความคมชัด (contrast) และ ความสว่าง (brightness) อย่างต่อเนื่อง
 - หากค่าพิกเซลผลลัพธ์เกินช่วงค่า $\text{bit_dept} = 8$
 - ปรับค่า $g(x, y) > 255 \rightarrow g(x, y) = 255$
 - ปรับค่า $g(x, y) < 0 \rightarrow g(x, y) = 0$

2.1.2 Image Pixel Adjustment

(Gamma Equation)

- 2.1.2 ให้สร้างภาพผลลัพธ์จากสมการ gamma $g(x, y) = af(x, y)^{\gamma} + b$
 - กำหนดให้ปรับค่า $a = 1$, $b = 0$ เพื่อสร้างภาพ $0 < \gamma < 1$ อย่างน้อย 10 ภาพ และ $\gamma > 1$ อย่างน้อย 10 ภาพ รวมทั้งหมดไม่น้อยกว่า 20 ภาพ
 - เขียนภาพทั้งหมดลงไฟล์วิดีโอ เพื่อให้เห็นการเปลี่ยนแปลงความคมชัด (contrast) และ ความสว่าง (brightness) อย่างต่อเนื่อง
 - หากค่าพิกเซลผลลัพธ์เกินช่วงค่า $\text{bit_dept} = 8$
 - ให้ทำ Quantization ค่าพิกเซลให้อยู่ในช่วง $[0, 255]$

2.1 Image Pixel Adjustment



“

2.2 Image Histogram Equalization

2.2 Image Histogram Equalization

- 2.2
- ให้สร้างภาพผลลัพธ์จากกระจายเฉดสีด้วย
 - เทคนิค Image Histogram Equalization
 - Using OpenCV: `cv2.equalizeHist()` แยกสี R, G, B
- คำนวณค่า Histogram แบบแยกสี R,G,B
 - ของภาพต้นฉบับ และ ภาพหลังทำ Histogram Equalization
 - Using `cv2.calcHist()`
- แสดงภาพเปรียบเทียบ
 - รูปภาพ และ กราฟเส้นแสดง Histogram สี R,G,B
 - ของภาพต้นฉบับและภาพผลลัพธ์หลังทำ Histogram Equalization

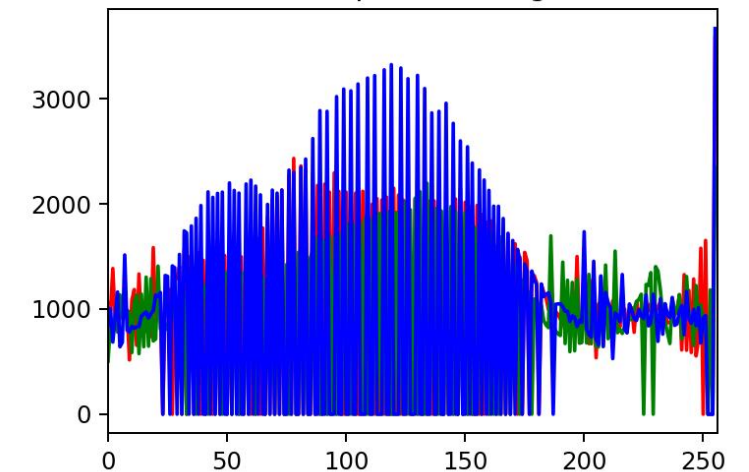
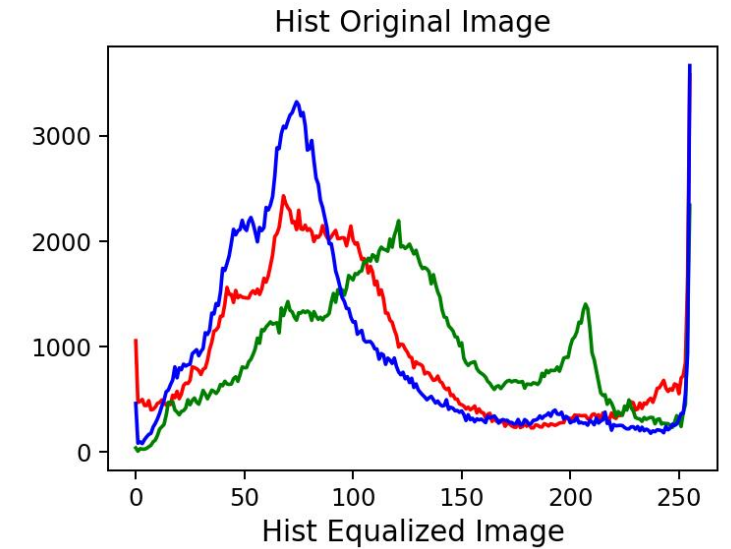
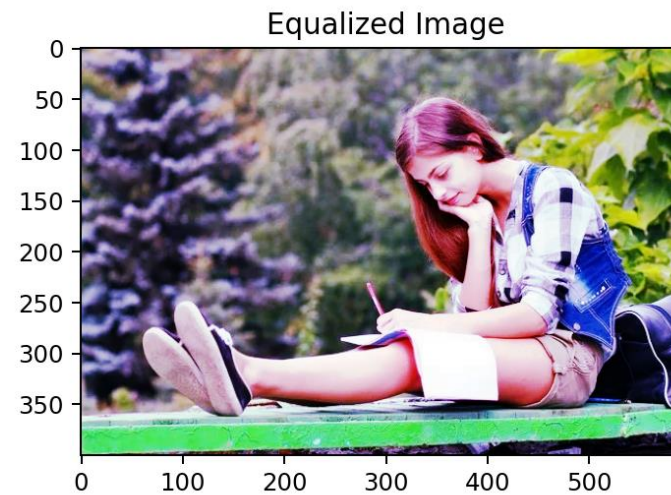
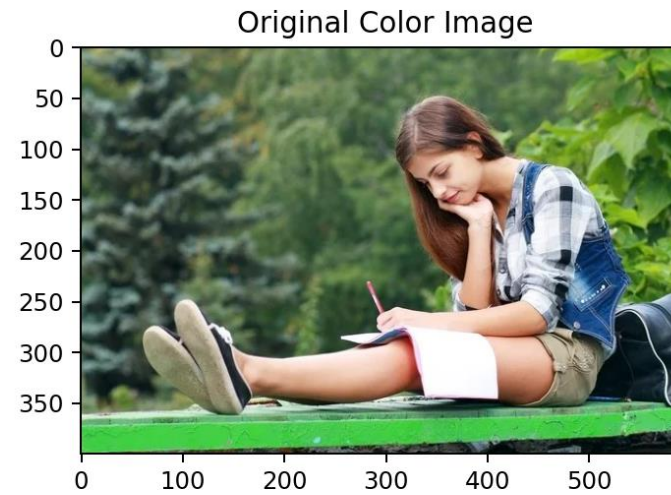
2.2 Image Histogram Equalization

- ☑ Histogram Equalization
Using openCV

`cv2.equalizeHist()`

R, G, B each channel

- ☑ Visualize Histogram of R,G,B
`cv2.calcHist()`



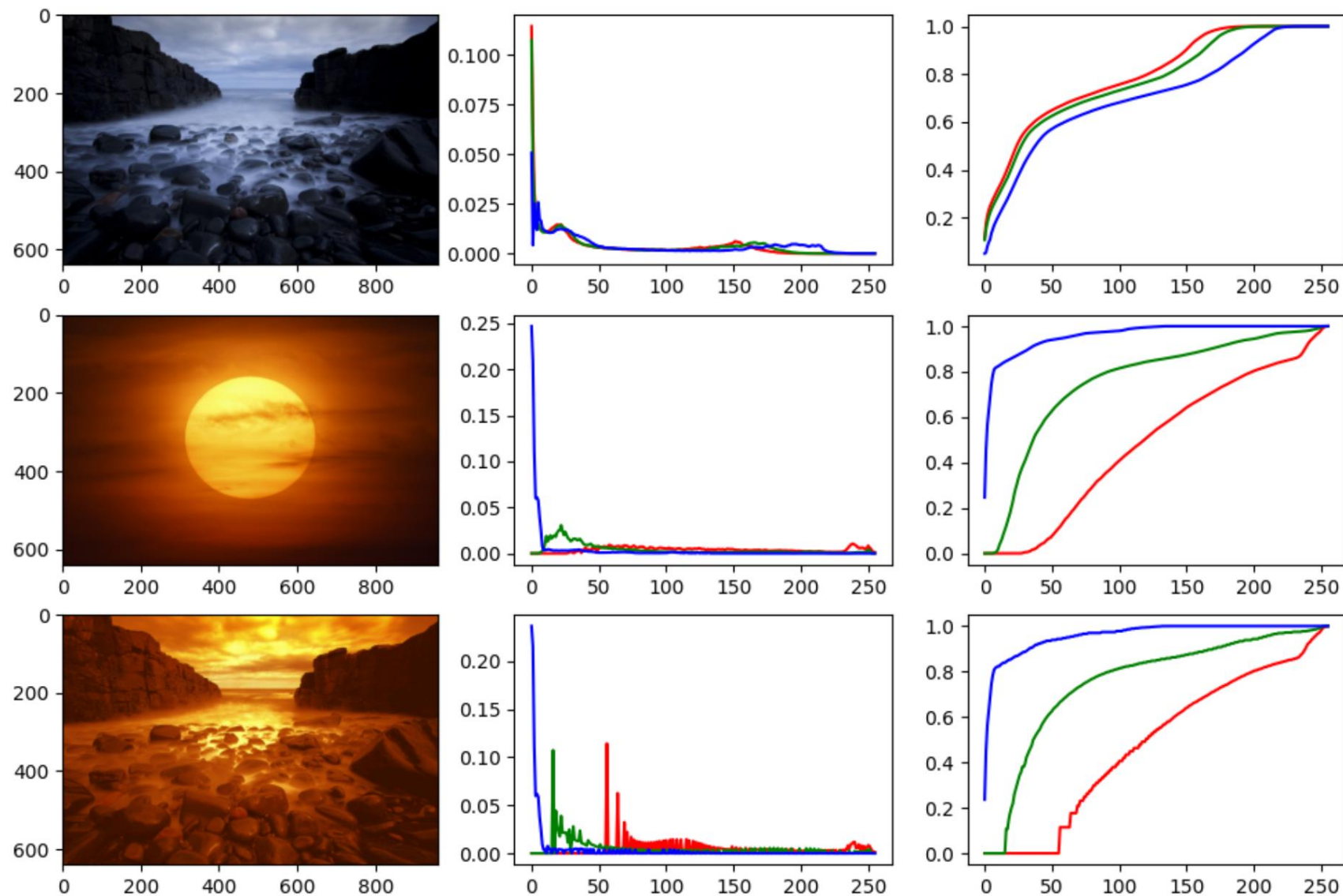
“

2.3 Image Histogram Matching

2.3 Image Histogram Matching

- 2.3
- อ่านไฟล์ภาพ
 - ภาพต้นฉบับ และภาพ template
- คำนวณค่า Normalized Histogram (pdf) แบบแยกสี R,G,B
 - ของภาพต้นฉบับ และภาพ template
 - โดยทำ Normalized array ผลลัพธ์จากฟังก์ชัน `cv2.calcHist()`
 - จำนวน nbins = 256 เกรดสี
- คำนวณค่า Cumulative Histogram (cdf)
 - จาก Normalized Histogram (pdf) ที่คำนวณได้ก่อนหน้านี้
 - ของภาพต้นฉบับ และภาพ template
 - ตรวจสอบ cdf bin ใน array ตำแหน่งสุดท้าย ต้องมีค่าใกล้เคียงหรือเท่ากับ 1.0
- เขียนฟังก์ชันเพื่อทำ Histogram Matching
 - โดยสร้างตารางการแปลงค่าสี ที่เกิดจาก
 - การเทียบแต่ละค่าสีที่มีค่า cdf ของภาพ template ที่ใกล้เคียงกับ cdf ของภาพต้นฉบับ
 - สร้างภาพผลลัพธ์ที่แปลงค่าเกรดสี R, G, B ของภาพต้นฉบับ ตามตารางแปลงค่าสีที่สร้างได้ก่อนหน้านี้
- แสดงภาพเปรียบเทียบ ภาพต้นฉบับ ภาพ template และกราฟ pdf, cdf ของภาพทั้งสอง
 - โดยใช้ฟังก์ชัน `subplot()`
- เขียนโปรแกรม save รูปภาพผลลัพธ์
 - โดยใช้ฟังก์ชัน `imwrite()`
 - ตรวจสอบ data type ให้ถูกต้องก่อนเขียนลงไฟล์

2.3 Image Histogram Matching



2.3 Image Histogram Matching

