```python
# Let's try to work on the 3 channels.
# Using a weighted grayscale, getting y as intensity, and then apply equalization.

# split the color image into 3 channels
# Note: after splitting, the output will be in the reverse order of the input,
# i.e., input RGB -> output BRG

(b, g, r) = cv.split(image)
y = 0.3*r + 0.59*g + 0.11*b  # Weighted grayscale, getting y as intensity
y = y.astype(np.uint8)  # change float to 0-255 integer values
cv2_imshow(y)
print("y (intensity); y = 0.3*r + 0.59*g + 0.11*b \n")

y_equalized = cv.equalizeHist(y)
cv2_imshow(y_equalized)
print("y (intensity) after HE\n")
```

```
y (intensity); y = 0.3*r + 0.59*g + 0.11*b
```

```
y (intensity) after HE
```

โค้ดชุดนี้ทำการแปลงภาพสีเป็นภาพระดับสีเทา (grayscale) และการปรับฮิสโตแกรมให้เท่ากัน (Histogram Equalization) ซึ่งสามารถอธิบายเป็นขั้นตอนได้ดังนี้

(b, g, r) = cv.split(image): ทำการแยกภาพสี image โดยใช้ฟังก์ชัน cv.split() ออกเป็นสามช่องสี ได้แก่ สีน้ำเงิน, สีเขียว และสีแดง เก็บไว้ในตัวแปร b, g, และ r

y = 0.3*r + 0.59*g + 0.11*b: บรรทัดนี้สร้างภาพระดับสีเทา y จากสีแดง เขียว และน้ำเงินที่แยกออกมา โดยให้ความสำคัญกับสีเขียวมากที่สุด รองลงมาคือสีแดง และสีน้ำเงินตามลำดับ ค่าที่ใช้มาจากมาตรฐานที่ใช้กันทั่วไปในการแปลง RGB เป็น grayscale แบบ Luminance

y = y.astype(np.uint8): บรรทัดนี้จะแปลงค่า y ให้เป็นชนิดข้อมูล np.uint8 ซึ่งเป็นจำนวนเต็ม 8 บิต มีค่าตั้งแต่ 0 ถึง 255 เพื่อให้สอดคล้องกับรูปแบบข้อมูลของภาพ เพื่อใช้กับฟังก์ชัน cv.equalizeHist() ที่จะใช้ในขั้นตอนถัดไป

cv2_imshow(y): แสดงภาพระดับสีเทา y ที่ได้จากการแปลงค่าในขั้นตอนก่อนหน้านี้

print("y (intensity); y = 0.3*r + 0.59*g + 0.11*b \n"): พิมพ์ข้อความอธิบายภาพที่แสดง

y_equalized = cv.equalizeHist(y): บรรทัดนี้ทำการปรับฮิสโตแกรมของภาพระดับสีเทา y ให้เท่ากัน โดยใช้ฟังก์ชัน cv.equalizeHist() การปรับฮิสโตแกรมให้เท่ากัน เป็นการกระจายความเข้มของสีในภาพให้สม่ำเสมอมากขึ้น ซึ่งมักจะช่วยเพิ่มคอนทราสต์และความคมชัดของภาพ

cv2_imshow(y_equalized): แสดงภาพ y_equalized ซึ่งเป็นภาพที่ผ่านการปรับฮิสโตแกรมแล้ว

print("y (intensity) after HE\n"): พิมพ์ข้อความอธิบายภาพที่แสดง

สรุป โค้ดชุดนี้จะแปลงภาพสีเป็นภาพระดับสีเทา แล้วทำการปรับฮิสโตแกรมของภาพสีเทา เพื่อปรับปรุงคุณภาพของภาพให้ดีขึ้น โดยเฉพาะในด้านคอนทราสต์และความคมชัด

ความแตกต่างกับโค้ดชุดอื่นที่ทำ Histogram Equalizied

โค้ดหลักใช้วิธีการแปลงเป็น Grayscale ก่อนปรับฮิสโตแกรม

ในขณะที่โค้ดชุดอื่นพยายามปรับฮิสโตแกรมของภาพสีโดยตรง แต่ใช้วิธีการที่แตกต่างกัน โดยใช้ระบบสีที่ต่างกัน หรือปรับแต่ละช่องสีแยกกัน

ส่วนโค้ดชุดสุดท้าย เป็นตัวอย่างที่ไม่ถูกต้องของการปรับฮิสโตแกรมของภาพสี เพราะการปรับแต่ละช่องสี RGB แยกกัน อาจทำให้สีสันของภาพผิดเพี้ยนไป