import cv2  
import numpy as np  
from matplotlib import pyplot as plt  
  
*# Đọc ảnh từ file*img = cv2.imread('koiddd.jpg')  
print("Kích thước của ảnh gốc là:", img.shape)  
  
*# Lấy các kênh màu*R, G, B = img[:,:,2], img[:,:,1], img[:,:,0]  
  
*# Chuyển đổi sang ảnh màu xám theo công thức*img\_gray = 0.2989\* R + 0.5870 \* G + 0.1140 \* B  
  
def rgb\_to\_hsv(r, g, b):  
 *# Normalization of RGB values* r, g, b = r / 255.0, g / 255.0, b / 255.0  
  
 *# Calculate Cmax, Cmin, and Δ* cmax = max(r, g, b)  
 cmin = min(r, g, b)  
 delta = cmax - cmin  
  
 *# Hue calculation* if delta == 0:  
 h = 0  
 elif cmax == r:  
 h = 60 \* ((g - b) / delta % 6)  
 elif cmax == g:  
 h = 60 \* ((b - r) / delta + 2)  
 elif cmax == b:  
 h = 60 \* ((r - g) / delta + 4)  
  
 *# Saturation calculation* s = 0 if cmax == 0 else delta / cmax  
  
 *# Value calculation* v = cmax  
  
 return round(h % 360), round(s \* 100), round(v \* 100)  
  
*# Khởi tạo ảnh HSV với giá trị Hue, Saturation, và Value*img\_hsv = np.zeros\_like(img)  
  
*# Chuyển đổi từ không gian màu RGB sang HSV*for i in range(img.shape[0]):  
 for j in range(img.shape[1]):  
 r, g, b = img[i, j]  
 h, s, v = rgb\_to\_hsv(r, g, b)  
 img\_hsv[i, j] = [h, s, v]  
  
*# Chuyển đổi không gian màu RGB sang YCbCr*Cr = 128 + 0.438\*R - 0.336\*G + 0.071\*B  
Cb = 128 - 0.148\*R - 0.290\*G - 0.438\*B  
  
*# Hàm tăng cường độ tương phản*def contrast\_stretching(image):  
 *# Chuẩn hóa giá trị pixel về khoảng [0, 1]* normalized\_image = image / 255.0  
  
 *# Đặt mức đen và mức trắng mong muốn (vd: 5% và 95% percentile)* min\_percentile = 5  
 max\_percentile = 95  
  
 min\_value = np.percentile(normalized\_image, min\_percentile)  
 max\_value = np.percentile(normalized\_image, max\_percentile)  
  
 *# Áp dụng công thức cắt mức* contrast\_stretched = np.clip((normalized\_image - min\_value) / (max\_value - min\_value), 0, 1)  
  
 *# Chuyển đổi về giá trị [0, 255]* contrast\_stretched = (contrast\_stretched \* 255).astype(np.uint8)  
  
 return contrast\_stretched  
  
*# Tính lược đồ mức xám*histogram = cv2.calcHist([img\_gray.astype('uint8')], [0], None, [256], [0, 256])  
  
*# Hiển thị các ảnh*fig1, axs1 = plt.subplots(2, 2, figsize=(10, 8))  
  
axs1[0, 0].imshow(img[:, :, ::-1])  
axs1[0, 0].set\_title('Ảnh gốc')  
  
*# Áp dụng contrast stretching cho ảnh mức xám*contrast\_stretched\_img\_gray = contrast\_stretching(img\_gray)  
  
*# Hiển thị ảnh sau khi tăng cường độ tương phản*axs1[0, 1].imshow(img\_gray, cmap='gray')  
axs1[0, 1].set\_title('Ảnh mức xám (Tăng cường độ tương phản)')  
  
*# Hiển thị tiêu đề cho ảnh HSV*axs1[1, 0].imshow(img\_hsv)  
axs1[1, 0].set\_title('Ảnh HSV')  
  
*# Hiển thị tiêu đề cho ảnh YCBCR*axs1[1, 1].imshow(cv2.merge([img\_gray, Cr, Cb]).astype(np.uint8))  
axs1[1, 1].set\_title('Ảnh YCBCR')  
  
*# Tạo một lưới subplot với 1 hàng và 3 cột*fig, axs = plt.subplots(1, 3, figsize=(10, 8))  
  
*# Hiển thị ảnh mức xám*axs[0].imshow(img\_gray, cmap='gray')  
axs[0].set\_title('Ảnh Mức Xám')  
  
*# Hiển thị lược đồ mức xám*axs[1].plot(histogram, color='blue')  
axs[1].set\_title('Lược Đồ Mức Xám')  
axs[1].set\_xlabel('Giá Trị Pixel')  
axs[1].set\_ylabel('Số Lượng Pixel')  
  
*# Áp dụng contrast stretching cho ảnh mức xám*contrast\_stretched\_img\_gray = contrast\_stretching(img\_gray)  
axs[2].imshow(contrast\_stretched\_img\_gray, cmap='gray')  
axs[2].set\_title('Tăng Cường Độ Tương Phản')  
  
*# Tăng khoảng cách giữa các subplot*plt.tight\_layout()  
  
plt.show()



