

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC & KỸ THUẬT MÁY TÍNH



BÀI TẬP LỚN 1
XỬ LÝ ẢNH SỐ VÀ THỊ GIÁC MÁY TÍNH

GRAYSCALE
AND COLOR IMAGES

GVHD: Ths. Võ Thanh Hùng
SV thực hiện: Võ Tấn Hưng – 2113623



Mục lục

1	Giới thiệu	2
1.1	Grayscale Image	2
1.2	Color Image	2
2	Hiện thực	4
2.1	Cơ sở lý thuyết	4
2.1.1	Grayscale Image	4
2.1.2	Color Image	4
2.2	Hiện thực với Python	4
2.2.1	Grayscale Image	5
2.2.2	Color Image	5
3	Kết quả	6
4	Nhận xét & Kết luận	7
4.1	Nhận xét	7
4.2	Kết luận	7

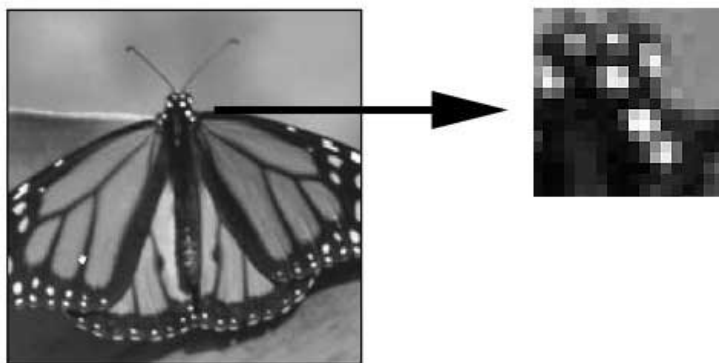
1 Giới thiệu

1.1 Grayscale Image

Hình ảnh xám là một loại hình ảnh màu đơn, trong đó mỗi pixel được biểu diễn bằng một giá trị duy nhất thể hiện độ sáng của pixel đó. Mỗi giá trị này thường là một con số nguyên trong khoảng từ 0 đến 255, với 0 thường tương ứng với màu đen và 255 tương ứng với màu trắng. Các giá trị ở giữa đại diện cho các cấp độ của sự xám từ đen đến trắng. So với hình ảnh đen trắng, mà chỉ có hai mức độ sáng: đen và trắng, hình ảnh xám có thể chứa nhiều hơn các mức độ sáng. Sự khác biệt này tạo ra cơ sở cho việc sử dụng và xử lý khác nhau giữa hai loại hình ảnh này trong các ứng dụng khác nhau, từ xử lý ảnh đến phân tích hình ảnh và nhận dạng đối tượng.

Hình ảnh xám, dù đơn giản chỉ với một kênh màu, có nhiều ứng dụng quan trọng trong nhiều lĩnh vực khác nhau:

- Trong y học, nó được sử dụng để chẩn đoán bệnh, theo dõi sự tiến triển của các tình trạng y tế, và xem xét hình ảnh từ các thiết bị như MRI và CT scan.
- Trong xử lý ảnh và phân tích hình ảnh, hình ảnh xám giúp phát hiện biên cạnh, nhận dạng đối tượng, và kiểm tra chất lượng sản phẩm trong sản xuất.
- Trong lĩnh vực an ninh và giám sát, hình ảnh xám giúp nhận dạng và theo dõi các sự kiện, như theo dõi sự di chuyển của đối tượng, nhận dạng khuôn mặt, và xác định các hành vi đáng ngờ.



Hình 1: Hình ảnh grayscale

1.2 Color Image

Ảnh màu (color image) là loại hình ảnh mà mỗi điểm ảnh được biểu diễn bằng 24 bits, trong đó mỗi kênh màu - đỏ (Red), lục (Green), và lam (Blue) - được lưu trữ trong 8 bits. Điều này cho phép mỗi điểm ảnh của hình ảnh RGB nhận giá trị từ 0 đến 16.777.216, tạo ra đến 16,7 triệu màu khác nhau. Cụ thể, mỗi kênh màu có thể có giá trị từ 0 đến 255, tạo ra các mức độ màu khác nhau. Hình ảnh màu thường được sử dụng trong ảnh chụp từ máy ảnh kỹ thuật số, trong đó chế độ hình ảnh là RGB.

Để lưu trữ ảnh màu người ta có thể lưu trữ từng màu riêng biệt, mỗi màu lưu trữ như một ảnh đa cấp xám. Do đó, không gian nhớ dành cho một ảnh màu lớn gấp 3 lần một ảnh đa mức xám cùng kích cỡ.

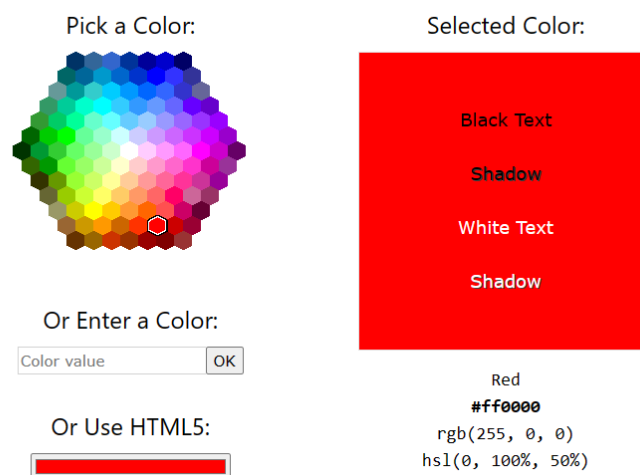
Ví dụ về các kết hợp màu RGB:

- (0, 0, 0) là màu đen.
- (255, 255, 255) là màu trắng.
- (255, 0, 0) là màu đỏ.



Hình 2: Hình ảnh Color

- (0, 255, 0) là màu xanh lục.
- (0, 0, 255) là màu xanh lam.
- (255, 255, 0) là màu vàng.
- (0, 255, 255) là màu xanh ngọc.
- (255, 0, 255) là màu hồng cánh sen.



Hình 3: Màu được chọn là $rgb(255, 0, 0)$, nghĩa là $R=255, G=0, B=0$.

2 Hiện thực

2.1 Cơ sở lý thuyết

2.1.1 Grayscale Image

Chuyển đổi từ ảnh màu sang ảnh xám là quá trình biến đổi một hình ảnh màu có ba kênh màu (RGB: Red, Green, Blue) thành một hình ảnh xám chỉ có một kênh màu. Trong mỗi pixel của ảnh màu, giá trị của mỗi kênh màu (R, G, B) thể hiện mức độ độ sáng của màu đỏ, xanh lá cây và xanh dương tương ứng.

Có một số phương pháp để chuyển đổi ảnh màu sang ảnh xám, trong đó một phương pháp phổ biến là sử dụng trọng số để tính toán giá trị của mỗi pixel trong ảnh xám từ giá trị của ba kênh màu trong ảnh màu. Một trong những cách phổ biến nhất để làm điều này là sử dụng trung bình có trọng số của các giá trị màu, với trọng số được chọn sao cho nó phản ánh độ quan trọng tương đối của mỗi kênh màu trong việc tạo ra ảnh xám.

Công thức phổ biến được sử dụng để tính toán giá trị của mỗi pixel trong ảnh xám từ các giá trị màu (R, G, B) là công thức trung bình có trọng số (weighted average formula), có thể được biểu diễn như sau:

$$\text{Giá trị pixel xám} = R \times \alpha + G \times \beta + B \times \gamma$$

Trong đó, **R**, **G**, và **B** là các giá trị màu cơ bản (Red, Green, Blue), và α , β , và γ là các trọng số tương ứng. Các trọng số này được lựa chọn dựa trên sự đóng góp tương đối của mỗi kênh màu vào độ sáng của một pixel xám.

2.1.2 Color Image

Chuyển đổi từ ảnh xám sang ảnh màu là một quá trình phức tạp nhằm tái tạo màu sắc từ thông tin độ sáng của pixel trong ảnh xám. Có nhiều phương pháp để thực hiện quá trình này, một trong số đó là sử dụng một bảng màu cố định và ánh xạ các giá trị pixel từ ảnh xám vào các màu tương ứng trong bảng màu.

Cách tiếp cận phổ biến nhất là chia phạm vi giá trị của ảnh xám thành các khoảng và ánh xạ mỗi khoảng này vào một màu cụ thể trong bảng màu. Quá trình này có thể được biểu diễn bằng công thức toán học như sau:

$$\text{Giá trị pixel màu} = f(\text{Giá trị theo bảng màu})$$

Trong đó, f là một hàm ánh xạ mỗi giá trị pixel xám vào một màu cụ thể trong bảng màu. Cách tiếp cận này tạo ra một ảnh màu có các pixel được gán màu dựa trên giá trị của pixel tương ứng trong ảnh xám.

Cụ thể các bước như sau:

1. Tạo bảng màu cố định: Bắt đầu bằng việc định nghĩa một bảng màu cố định. Mỗi màu trong bảng được biểu diễn bằng một mảng ba giá trị RGB.
2. Chia phạm vi giá trị của ảnh xám: Tính toán phạm vi giá trị của ảnh xám bằng cách tìm giá trị lớn nhất trong ảnh xám. Sau đó, nó chia phạm vi này thành các khoảng tương ứng với số lượng màu trong bảng màu.
3. Tạo ảnh màu từ ảnh xám: Duyệt qua từng pixel trong ảnh xám và tìm chỉ số màu tương ứng trong bảng màu dựa trên giá trị pixel. Sau đó, nó gán màu này cho pixel tương ứng trong ảnh màu đầu ra.

2.2 Hiện thực với Python

Có thể xem source tại đây: <https://github.com/hungvo2003vn/Computer-Vision/tree/Ass1>

Xem bài làm trên nhánh: **Ass1**

2.2.1 Grayscale Image

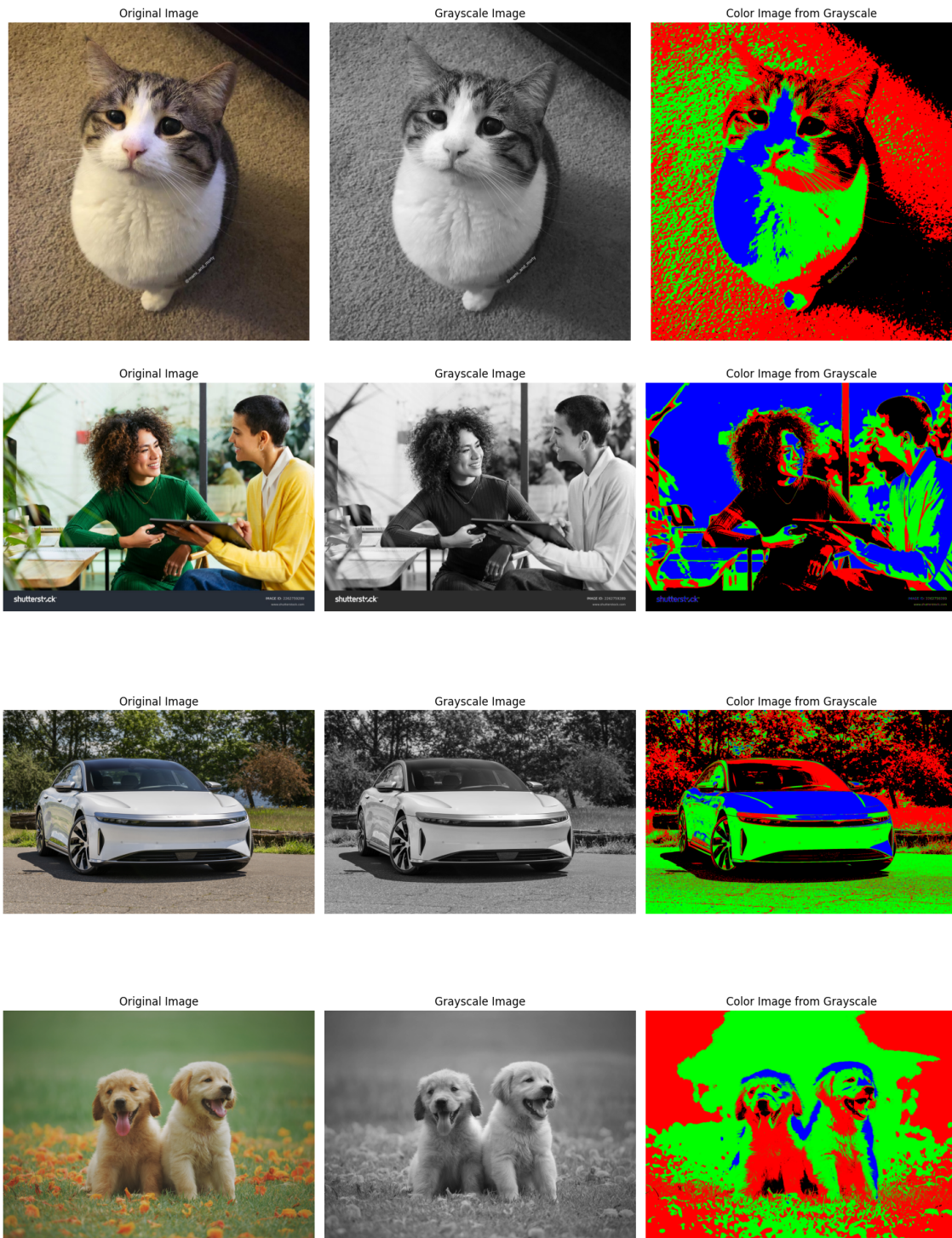
```
def rgb_to_gray(imgs):  
    return np.dot(imgs [..., :3], [0.2989, 0.5870, 0.1140])
```

2.2.2 Color Image

```
def gray_to_color(img_gray):  
  
    # Create fixed color palette  
  
    color_palette = [  
        [0, 0, 0],      # Black  
        [255, 0, 0],    # Red  
        [0, 255, 0],    # Green  
        [0, 0, 255],    # Blue  
        [255, 255, 255] # White  
        # Other colors  
    ]  
  
    # Partition the grayscale image value range into intervals  
    # corresponding to the number of colors in the color palette.  
  
    max_gray_value = np.max(img_gray)  
    num_colors = len(color_palette)  
    color_step = max_gray_value / (num_colors - 1)  
  
    # Create Color image from grayscale image  
  
    height, width = img_gray.shape  
    this_image_color = np.zeros((height, width, 3), dtype=np.uint8)  
    for i in range(height):  
        for j in range(width):  
            gray_value = img_gray[i, j]  
            color_index = int(gray_value / color_step)  
            this_image_color[i, j] = color_palette[color_index]  
  
    return this_image_color
```

3 Kết quả

Mỗi hàng sẽ có các cột tương ứng từ trái sang phải là hình ảnh màu gốc, hình ảnh xám tách từ ảnh gốc, hình ảnh màu phục hồi từ ảnh xám.



Hình 4: Kết quả sau khi chạy các phép toán

4 Nhận xét & Kết luận

4.1 Nhận xét

Chuyển đổi từ ảnh màu sang ảnh xám thường mang lại kết quả tốt vì quá trình này không mất thông tin màu sắc mà chỉ giữ lại thông tin về độ sáng. Tuy nhiên, chuyển đổi từ ảnh xám sang ảnh màu thường không được chính xác, đặc biệt là khi sử dụng các phương pháp đơn giản như ánh xạ giá trị pixel sang màu trong một bảng màu cố định.

4.2 Kết luận

Để cải thiện chất lượng của quá trình chuyển đổi từ ảnh xám sang ảnh màu, phương pháp sử dụng deep learning đã được chứng minh là hiệu quả. Bằng cách sử dụng mạng neural networks sâu, các mô hình có thể học được các mối quan hệ phức tạp giữa các đặc trưng của ảnh xám và ảnh màu tương ứng. Điều này giúp cải thiện đáng kể khả năng tái tạo màu sắc và tạo ra những ảnh màu chân thực hơn và chính xác hơn.

Tóm lại, sử dụng các phương pháp deep learning là một hướng tiếp cận có tiềm năng để cải thiện quá trình chuyển đổi từ ảnh xám sang ảnh màu, giúp tạo ra những kết quả chất lượng cao hơn và gần giống với thực tế hơn.

Tài liệu

- [1] Wikipedia, *Grayscale*, <https://en.wikipedia.org/wiki/Grayscale>.
- [2] Nttuan8, *Giới thiệu về xử lý ảnh*, <https://nttuan8.com/bai-5-gioi-thieu-ve-xu-ly-anh/>.
- [3] OpenCV, *Does anybody know where I can find the source code for `cvtColor()` function in OpenCV ?*, [forum of OpenCV](#).