

INT3404E 20 - Image Processing: Homeworks 1

Lương Thị Linh - 22028202

March 2024

1 Bài toán



Figure 1: The image coordinate system

Một bức ảnh có thể được biểu diễn dưới dạng một mảng NumPy của các "pixel", với kích thước $H \times W \times C$, trong đó H là chiều cao, W là chiều rộng, và C là số kênh màu. Hình 1 minh họa hệ tọa độ. Gốc tọa độ nằm ở góc trên bên trái và chiều thứ nhất chỉ ra hướng Y (hàng), trong khi chiều thứ hai chỉ ra chiều X (cột). Thông thường, chúng ta sẽ sử dụng một bức ảnh với các kênh màu đưa ra "mức độ" Red, Green, và Blue của mỗi pixel, được gọi tắt là RGB. Giá trị cho mỗi kênh dao động từ 0 (tối nhất) đến 255 (sáng nhất). Tuy nhiên, khi tải một bức ảnh thông qua Matplotlib, phạm vi này sẽ được tỷ lệ từ 0 (tối nhất) đến 1 (sáng nhất) thay vì và sẽ là một số thực, chứ không phải là số nguyên.

Bạn sẽ viết mã Python để tải một bức ảnh, thực hiện một số thao tác trên ảnh và minh họa hiệu ứng của chúng. Bạn sẽ cần lấy file uet.png từ cùng một nơi bạn tải xuống bài tập này.

2 Các hàm xử lý ảnh

2.1 load_image(image_path)

- Nhận đầu vào là đường dẫn của ảnh.
- Sử dụng OpenCV để tải ảnh từ tệp tin.
- Chuyển đổi không gian màu từ BGR sang RGB.
- Trả về ảnh RGB đã tải.

```
def load_image(image_path):  
    """  
    Load an image from file, using OpenCV  
    """  
    #load image with opencv  
    img = cv2.imread(image_path)  
    # convert BGR to RGB  
    img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)  
    return img_rgb
```

2.2 display_image(image, title="Image")

- Nhận đầu vào là ảnh và tiêu đề (mặc định là "Image").
- Sử dụng matplotlib để hiển thị ảnh.
- Tiêu đề ảnh được hiển thị.
- Trục không được hiển thị.
- Gọi plt.show() để hiển thị hình ảnh.

```
def display_image(image, title="Image"):  
    """  
    Display an image using matplotlib. Remember to use plt.show() to display the image  
    """  
    plt.imshow(image, cmap = 'gray')  
    plt.title(title)  
    plt.axis('off')  
    plt.show()
```

2.3 grayscale_image(image)

- Nhận đầu vào là ảnh màu RGB.
- Chuyển đổi ảnh màu sang ảnh xám sử dụng công thức chuyển đổi:

$$p = 0.299R + 0.587G + 0.114B$$

Trong đó R , G , B là giá trị của từng kênh tương ứng.

- Trả về ảnh xám (Kết quả như Figure 3)

```
def grayscale_image(image):  
    img_gray = 0.299 * image[:, :, 0] + 0.587 * image[:, :, 1] + 0.114 * image[:, :, 2]  
    return img_gray.astype(np.uint8)
```

2.4 `save_image(image, output_path)`

- Nhận đầu vào là ảnh và đường dẫn đến vị trí lưu.
- Sử dụng OpenCV để lưu ảnh vào tệp tin theo đường dẫn đã cung cấp.

```
def save_image(image, output_path):  
    """  
    Save an image to file using OpenCV  
    """  
    cv2.imwrite(output_path, image)
```

2.5 `flip_image(image)`

- Nhận đầu vào là ảnh.
- Sử dụng OpenCV để lật ảnh theo chiều ngang.
- Trả về ảnh đã lật (Kết quả như Figure 4).

```
def flip_image(image):  
    """  
    Flip an image horizontally using OpenCV  
    """  
    return cv2.flip(image, 1)
```

2.6 `rotate_image(image, angle)`

- Nhận đầu vào là ảnh và góc quay.
- Sử dụng OpenCV để xoay ảnh theo góc đã cho.
- Trả về ảnh đã xoay (Kết quả như Figure 5).

```
def rotate_image(image, angle):  
    """  
    Rotate an image using OpenCV. The angle is in degrees  
    """  
    rows, cols = image.shape[:2]  
    M = cv2.getRotationMatrix2D((cols/2, rows/2), angle, 1)  
    return cv2.warpAffine(image, M, (cols, rows))
```

3 Kết Quả

Tiến hành áp dụng các hàm này vào bức ảnh 'uet.png' được cung cấp và theo dõi các kết quả thu được.



Figure 2: Ảnh gốc.



Figure 3: Ảnh chuyển sang màu xám.



Figure 4: Ảnh lật theo chiều ngang.



Figure 5: Ảnh xoay 45°.

4 Kết luận

Bài tập này hướng dẫn thực hiện các thao tác cơ bản trên hình ảnh với Python:

- Hiểu cách biểu diễn hình ảnh bằng mảng NumPy.
- Thực hiện các thao tác cơ bản: lật, xoay, chuyển đổi ảnh màu sang ảnh xám.
- Báo cáo kết quả bằng LaTeX.

Lợi ích:

- Nâng cao kỹ năng lập trình Python, tư duy logic và giải quyết vấn đề.
- Rèn luyện kỹ năng viết báo cáo khoa học.

5 Tham khảo

Chi tiết về mã nguồn có thể được tìm thấy tại:

Link Github: [INT3404E_HW1](#)