

INT3404E 20 - Image Processing: Homeworks 1

Lã Thị Thanh Thúy - MSV: 21020478

1 Hàm flip_image(image)

Hàm `cv2.flip()` trong OpenCV cho phép lật hình ảnh quanh trục Ox, trục Oy hoặc cả hai trục.

- `cv2.flip(image, 1)`: lật ảnh theo chiều ngang.
- `cv2.flip(image, 0)`: lật ảnh theo chiều dọc.
- `cv2.flip(image, -1)`: lật ảnh theo cả hai chiều.

```
def flip_image(image):  
    """  
    Flip an image horizontally using OpenCV  
    """  
    if image is not None:  
        img_flipped = cv2.flip(image, 1)  
  
    return img_flipped  
pass
```

Trong ngữ cảnh của hàm `cv2.flip()` trong OpenCV, ta sử dụng lệnh `cv2.flip(image_gray, 1)` để lật hình ảnh theo chiều ngang. Với Input là **image_gray** (Figure 1)



Figure 1: image_gray.jpg.

Ta được Output là:



Figure 2: flip_image_gray.jpg.

2 Hàm rotate_image(image, angle)

Hàm rotate_image() trong OpenCV cho phép tạo ma trận xoay cho việc xoay hình ảnh bằng cách chỉ định tâm xoay và góc.

```
def rotate_image(image, angle):
    """
    Rotate an image using OpenCV. The angle is in degrees
    """
    if image is not None:
        (h, w) = image.shape[:2]
        center = (w / 2, h / 2)
        M = cv2.getRotationMatrix2D(center, angle, 1.0)
        img_rotated = cv2.warpAffine(image, M, (w, h))
        return img_rotated
    pass
```

Đoạn code trên thực hiện việc xoay image sử dụng OpenCV. cv2.getRotationMatrix2D(center, angle, scale) thực hiện tạo ma trận xoay cho việc xoay hình ảnh, trong đó:

- center là tham số chỉ định tâm xoay.
- angel là tham số chỉ định góc xoay.
- scale là tham số xác định tỷ lệ của hình ảnh sau khi xoay.

cv2.warpAffine(image, M, (w, h)) được sử dụng để áp dụng phép biến đổi affine cho hình ảnh. Trong đó:

- image là hình ảnh đầu vào cần thực hiện biến đổi.
- M là ma trận biến đổi affine.
- (w, h) là kích thước ảnh đầu ra sau khi biến đổi.

Trong ngữ cảnh của hàm `rotate_image(image, angle)`, với input là `image = "image_gray.jpg"`, `angle = "-45"` (Figure 3),



Figure 3: `image_gray.jpg`.

Ta được output là ảnh được xoay 45 độ so với ban đầu theo chiều ngược kim đồng hồ (Figure 4)



Figure 4: `rotate_image_gray.jpg`.

3 Hàm grayscale_image(image)

```
def grayscale_image(image):
    b, g, r = image[:, :, 0], image[:, :, 1], image[:, :, 2]
    gray = 0.299*r + 0.587*g + 0.114*b
    image_gray = gray.astype(np.uint8)
    return image_gray
pass
```

Với input là "uet.png", ảnh sẽ được trích xuất giá trị của các kênh màu blue, green và red $b, g, r = \text{image[:, :, 0]}, \text{image[:, :, 1]}, \text{image[:, :, 2]}$ sau đó tính giá trị của mỗi điểm ảnh trong không gian màu xám $\text{gray} = 0.299*r + 0.587*g + 0.114*b$ và trả về kết quả là một hình ảnh xám, trong đó mỗi giá trị điểm ảnh có đánh giá từ 0 đến 255 (được chuyển đổi sang kiểu dữ liệu uint8).



Figure 5: uet_image.jpg

Như vậy ta thu được output là:



Figure 6: image_gray.jpg.