INT3404E 20 - Image Processing: Homeworks 2

Phạm Vũ Duy

1 The result of the function grayscale_image

1.1 Source code

Listing 1: Code of grayscale_image() function

```python
def grayscale_image(image):
    height, width = image.shape[:2]
    img_gray = np.zeros((height, width), dtype=np.uint8)
    for i in range(height):
        for j in range(width):

            B = image[i, j, 0]
            G = image[i, j, 1]
            R = image[i, j, 2]

            gray_value = 0.299 * R + 0.587 * G + 0.114 * B

            img_gray[i, j] = gray_value

    return img_gray
```

1.2 Input

- Input:
  - `image`: a Numpy array containing the value of each point of the image in different color spaces.
- Algorithm:
  - Get the size of the image.
  - Create a Numpy array with the same size as the image and data type used to store the gray image.
  - Convert each pixel of an original image to a grayscale image using the following formula for each pixel:

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

1.3 Output

- Return the converted image

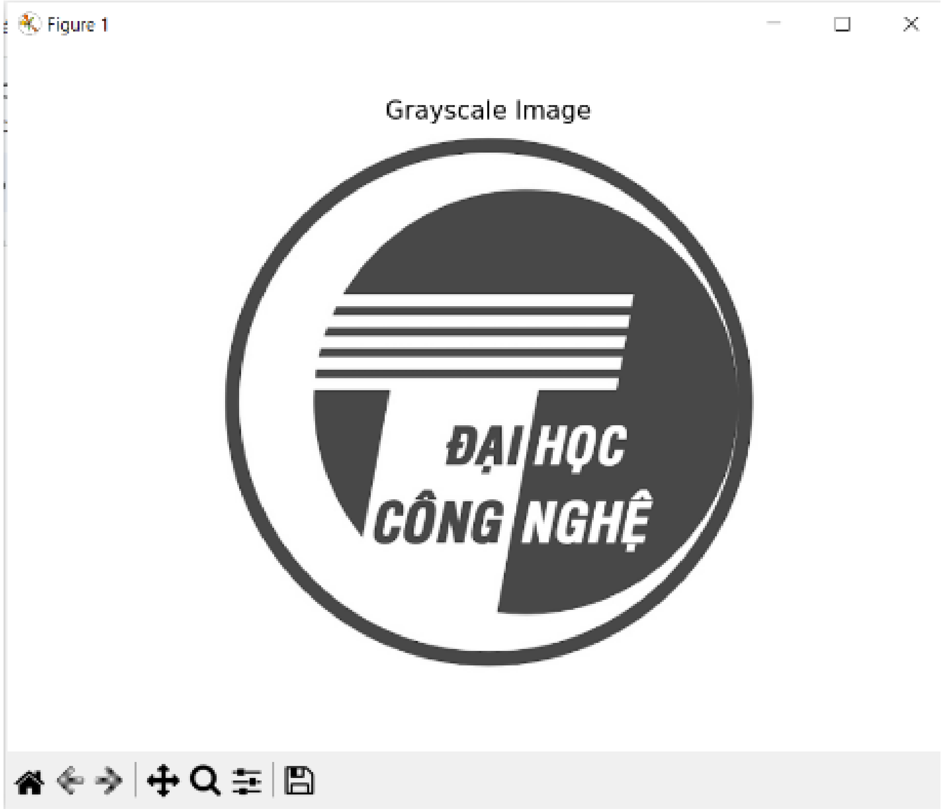

Figure 1: Grayscale Image

2 The result of the function flip_image

2.1 Source code

Listing 2: Code of flip_image() function

```
def flip_image(image):
    flip_image = cv2.flip(image, 1)
    return flip_image
```

2.2 Input

- Input:
  - `image`: A Numpy array containing the color value of each pixel in different color spaces.
- Algorithm:
  - Use function : `cv2.flip(image, 1)` with some parameters
    - `array: image` Save the color value of each pixel,
    - `flipCode = 1` Specify that the image will be flipped along the y-axis

2.3 Output

- Return the flipped image:

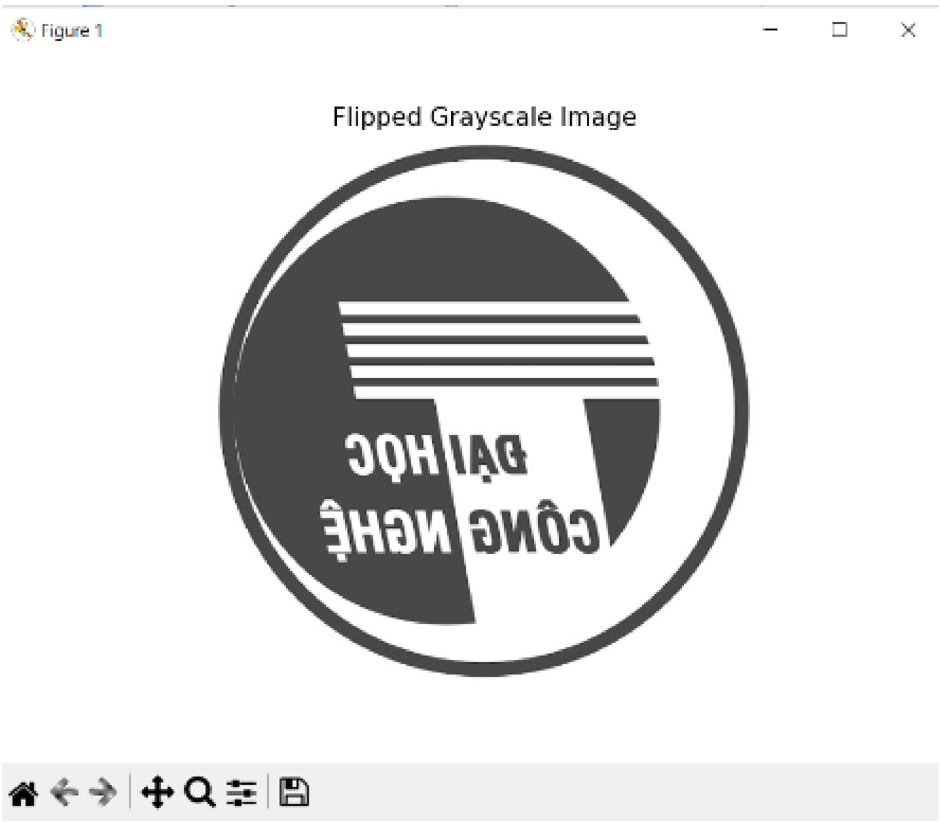

Figure 2: Flipped Grayscale Image

3 The result of the function rotate_image

3.1 Source code

Listing 3: Code of rotate_image() function

```python
def rotate_image(image, angle):
    height, width = image.shape[:2]
    center = (width / 2, height / 2)
    rotation_matrix = cv2.getRotationMatrix2D(center, angle, 1.0)
    rotated_image = cv2.warpAffine(image, rotation_matrix, (width, height)
    return rotated_image
```

3.2 Input

- Input:
  - `image`: A Numpy array containing the color value of each pixel in different color spaces.
  - `angle`: The rotation angle with a numeric value.
- Algorithm:
  - Extract the height and width of image from shape attribute.
  - Calculate the coordinate of the image's center point. The center of the image is the point halfway between the width and height.
  -

- Create a rotation matrix using the function `cv2.getRotationMatrix2D()`. This matrix is created to perform image rotation around the center point with a rotation angle and scaling ratio of 1.0.
- Apply the rotation defined by the rotation matrix `rotation_matrix` to the original image using the function `cv2.warpAffine()`. The argument `(width, height)` specifies the size of the output image.

3.3 Output

- Return the rotated image:

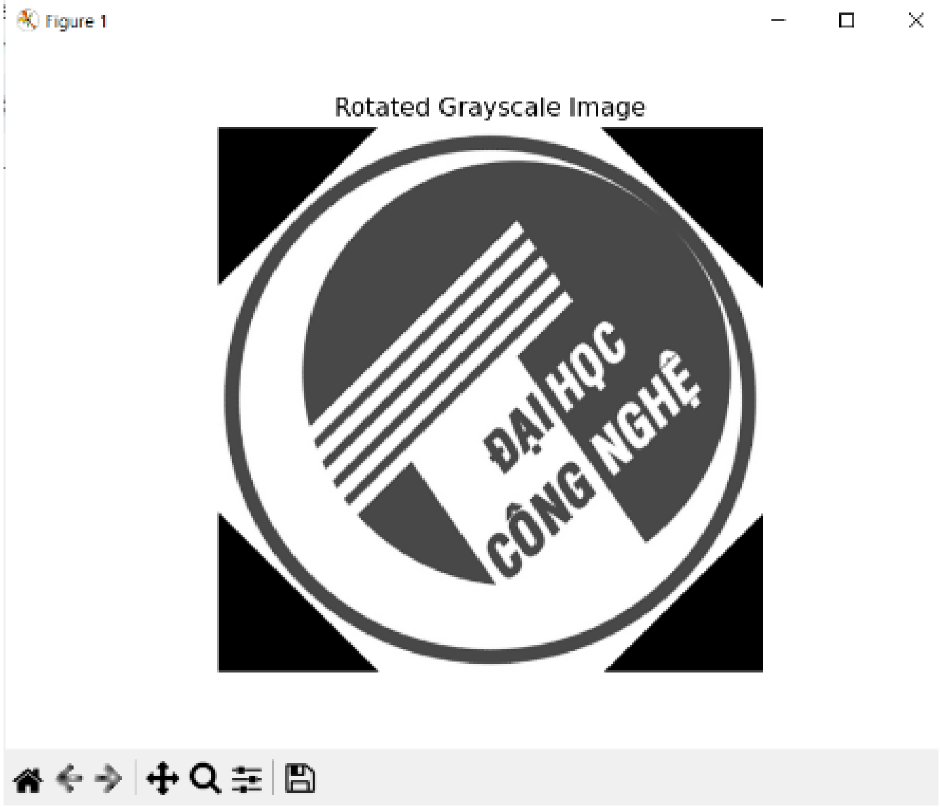

Figure 3: Rotated Grayscale Image