INT3404E 20 - Image Processing: Homework 1

Vu Thanh Long

1 Gray-scale Image:

```
def grayscale_image(image):
    img_gray = image.copy()
    R = np.array(image[:, :, 0])
    G = np.array(image[:, :, 1])
    B = np.array(image[:, :, 2])
    avg = R * .299 + G * .587 + B * .114
    for i in range (0, 3):
        img_gray[:, :, i] = avg
    return img_gray
```

The above function, converts a color image into gray-scale, following these steps:
1. Firstly, it creates an array with the same values as the input image.
2. Then, it separates the **Red**, **Green**, and **Blue** channels of the image into 3 separate arrays **(R, G, B)**.
3. It calculates the average intensity of the color using the formula below:

$$\text{avg} = \text{R} * .299 + \text{G} * .587 + \text{B} * .114$$

The above formula is a standard method for converting an image to gray-scale.
4. Then, it replaces each color channel in the copied image with this average intensity, result in a gray-scale version of the original, and returns the converted image.
The final result can be seen in Figure 1 below.

Figure 1: Converted to Gray-scale Image

2 Flip Image:

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

The above function flip the image horizontally using **cv2.flip** function. **cv2.flip(src, flipCode)** requires 2 parameters:
- *src* is the source image.
- *flipCode* is a flag which is used to identify the axis of rotation, i.e., 0 is x-axis, 1 is y-axis, -1 is all axes.
The final result is shown in Figure 2 below.

Figure 2: Flipped Image

3 Rotate Image:

```python
def rotate_image(image, angle):
    """
    Rotate an image using OpenCV. The angle is in degrees
    """
    height, width = image.shape[:2]
    rotateMatrix = cv2.getRotationMatrix2D(
        center = (width / 2, height / 2),
        angle = angle,
        scale = 1
    )
    res = cv2.warpAffine(
        src = image,
        M = rotateMatrix,
        dsize = (width, height)
    )
    return res
```

The above function rotates an image by a specified angle (in degrees) using a 2D rotation matrix, following these steps:
1. It calculates the height and width of the input image.
2. Then, a **2D Rotation Matrix** is created by using the **cv2.getRotationMatrix2D** function:
- *center* is the center of rotation, which in this case is the center of the image.
- *angle* is the input angle of rotation.
- *scale* is the scaling factor which scales the image.
3. The rotation is applied to the image using **cv2.warpAffine** function.
- *src* is the source image.
- *M* is the calculated rotation matrix.
- *dsize* is the size of the output image, which is the same as the input image in this case.
The final result is presented in Figure 3 below.

Figure 3: Rotated Image