Image Manipulation Assignment Report

Manohar Kowthavarapu (11554625)

Section-1:

* Below is the my_flip1 method that takes image as an input and returns the mirror image.

```
def my_flip1(lena_image):
  print("Mirror Image")
  cv2_imshow(lena_image[:, ::-1, :])
```

* Below is the my_flip2 method that takes image as an input and returns the rotated image with grey in colour.

```
def my_flip2(lena_image):
    radius = math.radians(270)
    lena_image = lena_image[:, ::-1, :]
    image = np.uint8(np.zeros(lena_image.shape))
    image_height, image_width, image_width_x, \
    image_height_y = get_image_width_height(image)
    for i in range(image_height):
        for j in range(image_width):
            x = round((i - image_width_x) * math.cos(radius) \
            + (j - image_height_y) * math.sin(radius)) + image_width_x
            y = round(-(i- image_width_x) * math.sin(radius) \
            + (j - image_height_y) * math.cos(radius)) + image_height_y
            if ( x >= 0 and y >= 0 and x < image_height and y < image_w
idth):
                image[i,j,:] = lena_image[x,y,:]
    print("Rotated Image with Grey in color")
    cv2_imshow(image[:, :, 1])
```

* Below is the upper_quardant method that takes image as an input and return the 1/4th of the right upper quadrant image.

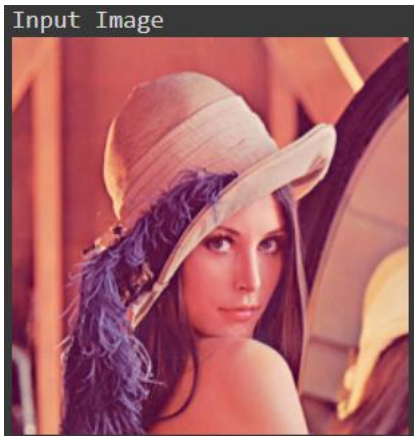
```
def upper_quadrant(lena_image):
  image_height, image_width, image_width_x, \
  image_height_y = get_image_width_height(lena_image)
  upper_quadrant_right_image = lena_image[
                                          0:image_height_y,
                                          image_width_x: image_width
                                          ]
  print("Top Right 1/4th uppe")
  cv2_imshow(upper_quadrant_right_image)
```

* Below is my main method, in which I'll call the methods I've described above.

```
def main():
  lena_image =cv2.imread('/content/lena.png')
  print("Input Image")
  cv2_imshow(lena_image)
  my_flip1(lena_image)
  my_flip2(lena_image)
  upper_quadrant(lena_image)
```

Section-2:

Output result for the given above methods.



Section-3:

- We'll get an array after reading the image, now I'll generate a mirror image using the index logic.
- There are numerous libraries that can rotate a picture by simply providing the image to the appropriate library method, but I wanted to develop that logic from scratch, so I did some research and tried a few different logics before settling on a mathematical formula.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} cos(0) & \sin(0) \\ -\sin(0) & \cos(0) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

  I've written the equation using the math library, where x' and y' are the new coordinates after rotation by angle 270.

- I utilized the first two numbers from lena image.shape as height and width, then did the floor division on both, then used those values on the lena_image to get the upper quadrant.