

HW 1: Basic Image Manipulation

Source Code

All questions are written in Python code, please refer to the file “main.py”.

All images will be stored in the folder “res” (automatically create a new folder).

In accordance with the **FAQ** of course website:

- Part 1 does not use any library function, except Numpy (for organizing purposes).
- Part 2 uses other library functions as **there are no restrictions**.

Part 1

1. Upside Down

Algorithm:

- 1) Read from the bottom-right corner of the original image
- 2) Store in a list
- 3) Convert list to numpy array
- 4) Reshape to original size

```
udimg = []
for i in reversed(range(height)):
    for j in range(width):
        udimg.append(img[i][j])
udimg = np.asarray(udimg).reshape(height, width)
cv2.imwrite("res/upside_down.bmp", udimg)
```



2. Right-side Left

Algorithm:

- 1) Read original image from the right
- 2) Store in a list
- 3) Convert list to numpy array
- 4) Reshape to original size

```
lrmg = []
for i in range(height):
    for j in reversed(range(width)):
        lrmg.append(img[i][j])
lrmg = np.asarray(lrmg).reshape(height, width)
cv2.imwrite("res/righside_left.bmp", lrmg)
```



3. Diagonally Flip

Algorithm:

- 1) Read original image
- 2) Swap the values of left half of the image and right half of the image

```
dfimg = np.copy(img)
for i in range(height):
    for j in range(width):
        if i == j:
            break
        else:
            temp = dfimg[i][j]
            dfimg[i][j] = dfimg[j][i]
            dfimg[j][i] = temp
cv2.imwrite("res/diagonal_flip.bmp", dfimg)
```



Part 2

4. Rotate 45 degrees clockwise

Algorithm:

- 1) Use `scipy.ndimage` to rotate image (`ndimage.rotate(img, -45)`)
- 2) Change values of 0 with 255 (black -> white)

```
rotateimg = ndimage.rotate(img, -45)
rheight, rwidth = rotateimg.shape
for i in range(rheight):
    for j in range(rwidth):
        if rotateimg[i][j] == 0:
            rotateimg[i][j] = 255
cv2.imwrite("res/rotated.bmp", rotateimg)
```



5. Shrink image in half

Algorithm:

- 1) Use cv2.resize to resize image
- 2) Insert size of half-sized image into the function

```
shrink = cv2.resize(img, (int(height/2), int(width/2)),  
cv2.INTER_AREA)  
cv2.imwrite("res/shrunked.bmp", shrink)
```

6. Binarize image

Algorithm: Use cv2.threshold to binarize image

```
_, binarize = cv2.threshold(img, 128, 255, cv2.THRESH_BINARY)  
cv2.imwrite("res/binarize.bmp", binarize)
```

