

영상정보처리 6주차 과제 템플릿

- 점수: 10점 만점
- 이미지 경로 잘못 사용한 경우: -3

이름: 이현정

학번: 32203660

다음의 경로는 변경하지 말것

```
image_path = '../Dongkeun-OpenCV-ImgData/lena.jpg'
```

```
from google.colab import drive
drive.mount('/gdrive')
```

```
%cd /gdrive/MyDrive/ImageProcClass/Notebook-Week6
```

```
Mounted at /gdrive
/gdrive/MyDrive/ImageProcClass/Notebook-Week6
```

문제 1: 5점

1. 주어진 영상을 읽어 들인다 (org_img)
2. 입력 영상과 같은 크기의 빈 컬러 이미지를 만든다 (new_image)
3. 공백 이미지(new_image) 를 2 x 2 로 분할하여 4개의 서브이미지 영역을 구분한다.
4. 오른쪽 위를 1 영역이라고 하고, 시계 방향으로 2, 3, 4 영역이라고 한다.
5. 1영역은 blue channel 값만을 가지고, 2영역은 green channel, 3영역은 red channel 값만 가지고, 4 영역은 3개의 channel 이 merging 된 영상을 집어 넣는다.
6. new_image 를 출력한다.

```
import cv2
import matplotlib.pyplot as plt
import numpy as np

org_img = cv2.imread(image_path, cv2.COLOR_BAYER_BG2BGR)

new_image = np.zeros(org_img.shape, dtype=org_img.dtype)

N = 2 # 분할 영역 수
w,h = new_image.shape[:2]
w = w//N
h = h//N

roi1 = new_image[w:, 0:h]
roi2 = new_image[w:, h:]
```

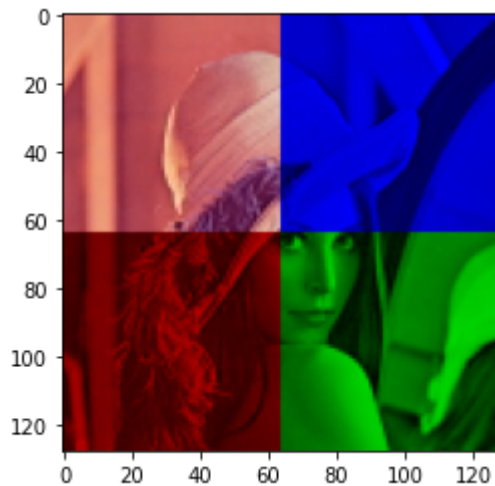
```

roi3 = new_image[0:w, h:]
roi4 = new_image[0:w, 0:h]

roi1[:, :, 0] = org_img[w:, 0:h, 0]
roi2[:, :, 1] = org_img[w:, h:, 1]
roi3[:, :, 2] = org_img[0:w, h:, 2]
(b,g,r) = cv2.split(org_img[0:w, 0:h])
roi4[:, :] = cv2.merge([r,g,b])

plt.imshow(new_image)
plt.show()

```



문제 2: 5점

강의자료 2022-1 ImgProc JB - CH05-Updated2103.pdf 에 있는 예제 중, 07.bitwise_op.py 와 08.bitwise_overlap.py 의 예제를 이 노트북에서 실행시키고, 각 라인을 분석해서 자기 노트를 만들기.

- 예제에 사용하는 이미지가 '../Dongkeun-OpenCV-ImgData' 폴더에 없는 경우, 공유한 소스코드 폴더에서 필요한 이미지를 '../Dongkeun-OpenCV-ImgData' 에 복사해서 사용할 것.

```

# 07.bitwise_op.p
import numpy as np
import cv2
import matplotlib.pyplot as plt

from google.colab.patches import cv2_imshow

image1 = np.zeros((300, 300), dtype= np.uint8) # 300*300 빈 이미지 생성
image2 = image1.copy() # 이미지 복사

h, w = image1.shape[:2] # 높이와 가로가 입력
cx, cy = w//2, h//2 # 높이, 가로의 절반 값
cv2.circle(image1,(cx,cy),100,255,-1) # (cx,cy) 원의 중심을 기준으로
cv2.rectangle(image2,(0,0,cx,h),255,-1) # (0,0)에서 시작한 사각형의 오른쪽아래가 (cx,h)가 됨

```

```
# image1의 값 = v1, image2의 값 = v2 가정
image3 = cv2.bitwise_or(image1, image2)
# v1=1 or v2=1 : 흰색 / v1=0, v2=0 : 검정색
image4 = cv2.bitwise_and(image1, image2)
# v1=0 or v2=0 : 검정색 / v1=1, v2=1 : 흰색
image5 = cv2.bitwise_xor(image1, image2)
# v1!=v2 : 흰색 / v1==v2 : 검정색
image6 = cv2.bitwise_not(image1)
# v1=0 : 흰색 / v1=1 : 검정색
```

```
print('image1')
cv2_imshow(image1) # image1 결과 출력
print('image2')
cv2_imshow(image2) # image2 결과 출력
print(' bitwise_or')
cv2_imshow(image3) # bitwise_or 결과 출력
print(' bitwise_and')
cv2_imshow(image4) # bitwise_and 결과 출력
print(' bitwise_xor')
cv2_imshow(image5) # bitwise_xor 결과 출력
print(' bitwise_not')
cv2_imshow(image6) # bitwise_not 결과 출력
```

image1

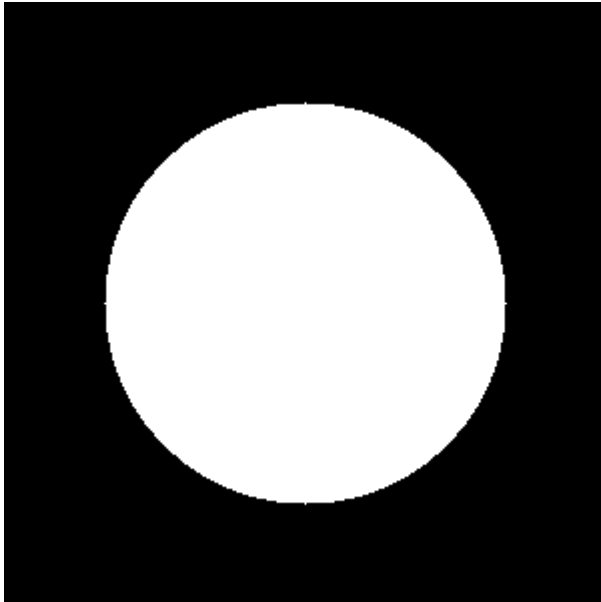
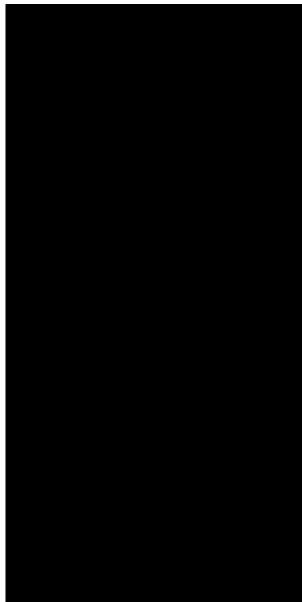


image2



bitwise_or



bitwise_and





bitwise_xor



```
# 08.bitwise_overlap.py
import numpy as np, cv2

from google.colab.patches import cv2_imshow

image = cv2.imread("../Dongkeun-OpenCV-ImgData/bit_test.jpg", cv2.IMREAD_COLOR) # 이미지 색으로 읽기
logo = cv2.imread("../Dongkeun-OpenCV-ImgData/logo.jpg", cv2.IMREAD_COLOR) # 이미지 색으로 읽기
if image is None or logo is None :
    raise Exception("영상파일 읽기 오류") # 읽지 못한 파일 처리

masks = cv2.threshold(logo, 220,255, cv2.THRESH_BINARY)[1]
# 220보다 크면 255(value)로, 작은 경우는 0으로
masks = cv2.split(masks) # 채널 분리

fg_pass_mask = cv2.bitwise_or(masks[0],masks[1])
# masks[0]값=255 or masks[1]값=255 : 255 / 그외 = 둘다 0 : 0
fg_pass_mask = cv2.bitwise_or(masks[2], fg_pass_mask)
# masks[2]값=255 or fg_pass_mask값=255 : 255 / 그외 = 둘다 0 : 0
# 전경 통과 - 3개 마스크 중 1개라도 255인 경우가 존재하는 부분만 255
bg_pass_mask = cv2.bitwise_not(fg_pass_mask)
# 배경 통과 - fg_pass_mask=255 : 0 / fg_pass_mask=0 : 255

# => 필요한 부분을 255로

(H,W), (h,w) = image.shape[:2], logo.shape[:2] # 각 이미지의 height와 width
x,y = (W-w)//2, (H-h)//2 # 시작하는 이미지 좌표
roi = image[y:y+h, x:x+w] # 큰 이미지의 중앙에 작은 이미지가 들어갈 부분

foreground = cv2.bitwise_and(logo, logo, mask=fg_pass_mask)
# fg_pass_mask가 0이 아닌 부분만 logo 값들이 합쳐짐 => 전경
background = cv2.bitwise_and(roi, roi, mask=bg_pass_mask)
# bg_pass_mask가 0이 아닌 부분만 roi 값들이 합쳐짐 => 배경

dst = cv2.add(background, foreground) # 2개의 값이 결합, 255이상의 결과는 255로 처리
image[y:y+h, x:x+w] = dst # image 이미지의 중앙에 dst 배치

print('background')
```

```
cv2_imshow(background)
print('foreground')
cv2_imshow(foreground)
print('dst')
cv2_imshow(dst)
print('image')
cv2_imshow(image)
```