

	영상정보처리 LAB 02
학번	1814965
이름	김현주

## # 실습 문제 1. Color 영상 RGB 값만 획득

# 1 소스코드(원본 서식 유지로 복사 or 코드 화면 캡처)

```

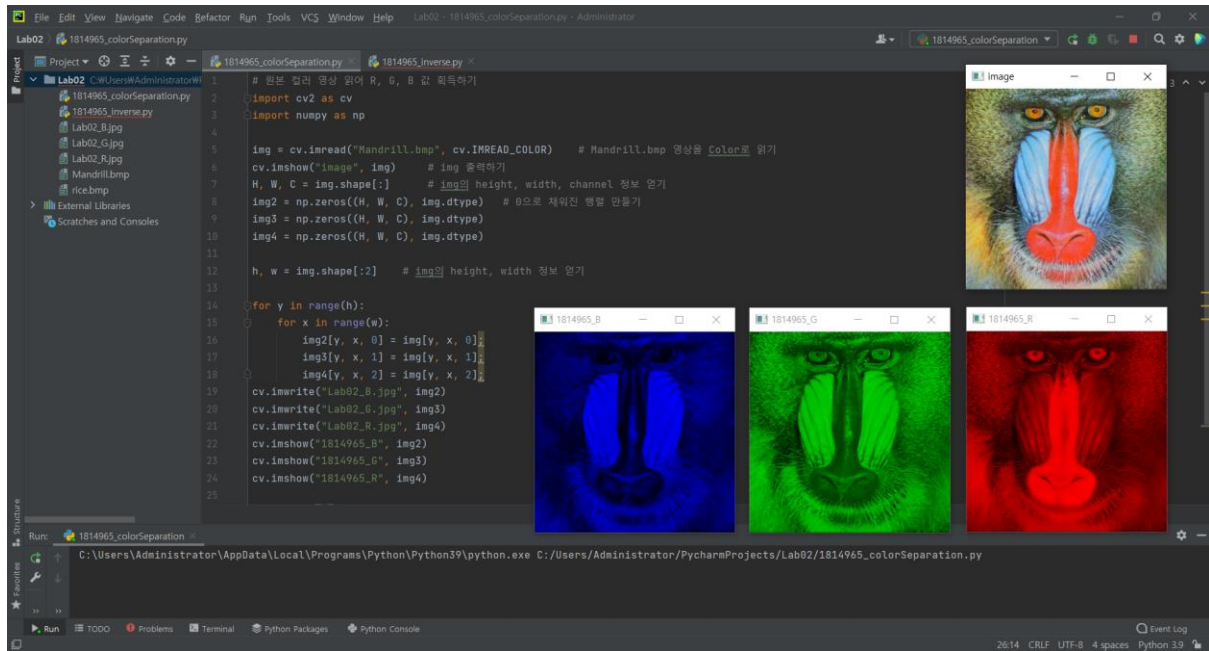
1  # 원본 컬러 영상 읽어 R, G, B 값 획득하기
2  import cv2 as cv
3  import numpy as np
4
5  img = cv.imread("Mandrill.bmp", cv.IMREAD_COLOR) # Mandrill.bmp 영상을 Color로 읽기
6  cv.imshow("image", img) # img 출력하기
7  H, W, C = img.shape[:] # img의 height, width, channel 정보 얻기
8  img2 = np.zeros((H, W, C), img.dtype) # img와 동일한 크기의 0으로 채워진 행렬 만들기
9  img3 = np.zeros((H, W, C), img.dtype)
10 img4 = np.zeros((H, W, C), img.dtype)
11
12 h, w = img.shape[:2] # img의 height, width 정보 얻기
13
14 for y in range(h):
15     for x in range(w):
16         img2[y, x, 0] = img[y, x, 0] # img의 B 값만 가져와 img2의 B에 저장
17         img3[y, x, 1] = img[y, x, 1] # img의 G 값만 가져와 img3의 G에 저장
18         img4[y, x, 2] = img[y, x, 2] # img의 R 값만 가져와 img4의 R에 저장
19     cv.imwrite("Lab02_B.jpg", img2) # img2를 Lab02_B.jpg로 저장
20     cv.imwrite("Lab02_G.jpg", img3)
21     cv.imwrite("Lab02_R.jpg", img4)
22     cv.imshow("1814965_B", img2) # img2 출력하기
23     cv.imshow("1814965_G", img3)
24     cv.imshow("1814965_R", img4)
25
26 cv.waitKey(0)

```

: 원본 영상과 동일한 크기를 가진 배열을 만들고 0으로 초기화한 후, 원본 영상의 특정 색상만 지정

- 1) 영상의 height, width, shape 정보(배열 크기) 얻기: `H, W, C = img.shape`
- 2) 원본 영상 크기의 배열 생성 후 0으로 초기화: `numpy.zeros((H, W, C), img.dtype)`
- 3) 각 픽셀에 원본 영상의 특정 색상 값 저장: `img2[y, x, 0] = img[y, x, 0]`
- 4) 영상 저장: `cv.imwrite(파일 이름, 저장할 영상)`

# 2 실행 화면



## # 실습 문제 2. 흑백 영상 역상 이미지 만들기

# 1 소스코드(원본 서식 유지로 복사 or 코드 화면 캡처)

```

1  # 흑백 영상 읽어 역상 영상 획득하기
2  import cv2 as cv
3  import numpy as np
4
5  img = cv.imread("rice.bmp", cv.IMREAD_GRAYSCALE)      # 흑백 영상 읽기
6  H, W = img.shape[:]      # height, width 값 얻기(흑백 - 2차원)
7  img_inverse = np.zeros((H, W), img.dtype)      # 원본 영상과 동일한 크기의 0으로 채워진 배열 생성
8  for y in range(H):
9      for x in range(W):
10         img_inverse[y, x] = 255 - img[y, x]      # 역상 색상 저장
11
12     cv.imwrite('Lab02_inverse.jpg', img_inverse)
13
14     cv.imshow("image", img)
15     cv.imshow("image_inverse", img_inverse)
16     cv.waitKey(0)

```

: 흑백 영상은 0~255의 값만 가짐. 역상 영상은 255에서 원본의 픽셀 값을 뺀

- 1) 영상의 height, width 정보(배열 크기) 얻기: `H, W = img.shape`
- 2) 원본 영상 크기의 배열 생성 후 0으로 초기화: `numpy.zeros((H, W), img.dtype)`
- 3) 각 픽셀에 255에서 원본 픽셀 값 뺀 값 저장: `img_inverse[y, x] = 255 - img[y, x]`
- 4) 영상 저장: `cv.imwrite(파일 이름, 저장할 영상)`

# 2 실행 화면

