

▼ 2022년 1학기 영상정보처리 중간고사

이름: 김경민

학번: 32200327

제출 파일:

1. 노트북 파일: 2022-1-ImgProc-MidtermHandin-이름-학번.ipynb
2. PDF 파일: 2022-1-ImgProc-MidtermHandin-이름-학번.pdf

**** 파일 이름 형식이 맞지 않는 경우 -5점**

**** 제출물이 하나라도 빠진 경우, 채점에서 제외하고 0점 처리**

구글 주피터 노트북 사용을 위한 초기 작업을 작성하시오

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

```
%cd /gdrive/My Drive/Classroom/[22-1 영상정보처리] 2000004793-2022-1/ImageProcClass/Notebook-week8
!pwd
```

```
sample_img_folder = "../2022-1-Midterm-SampleImages"
```

```
Drive already mounted at /gdrive; to attempt to forcibly remount, call drive.mount("/gdrive",
/gdrive/My Drive/Classroom/[22-1 영상정보처리] 2000004793-2022-1/ImageProcClass/Notebook-week
/gdrive/My Drive/Classroom/[22-1 영상정보처리] 2000004793-2022-1/ImageProcClass/Notebook-week
```



다음 셀에 있는 코드는 수정하지 말고 수행만 할 것.

함수 `show_with_matplotlib(img, title)` 는 BGR 형태의 컬러 이미지 `img` 와 이미지 상단에 출력되는 타이틀 문자열 `title` 을 매개변수로 하여, 주어진 이미지와 타이틀을 출력합니다.

함수 `show_with_matplotlib_gray(img, title)` 는 그레이스케일 형태의 이미지와 타이틀을 매개변수로 받아 출력합니다.

***** 동일 이미지 형태 체크 함수 검수**

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

```
# matplotlib color display
def show_with_matplotlib(img, title):
    """Shows an image using matplotlib capabilities"""
    if img is None:
        print("show_with_matplotlib: Could not read the image.")
    return
```

```
if img.shape[2] != 3:
    print()
    print("show_with_matplotlib: given image does not contains 3 channels")
    return

# Convert BGR image to RGB:
img_RGB = img[:, :, ::-1]

# Show the image using matplotlib:
plt.imshow(img_RGB)
plt.title(title)
plt.show()

# matplotlib grayscale display
def show_with_matplotlib_gray(img, title):
    if img is None:
        print("show_with_matplotlib_gray: Could not read the image.")
        return

    if img is None:
        print()
        print("show_with_matplotlib: given image has more than 2 dim")
        return

    plt.imshow(img, cmap="gray")
    plt.title(title)
    plt.show()

# image equivalence check
def test_same_image(img1, img2):
    result = True

    if img1.shape != img2.shape:
        print("img1.shape = ", img1.shape, " and img2.shape = ", img2.shape, ' are different')
        result = False

    difference = cv2.subtract(img1, img2)
    b, g, r = cv2.split(difference)
    if np.bitwise_xor(img1, img2).any() :
        print("img1 and img2 are not identical")
        result = False
    else :
        print("img1 and img2 are identical.O.K.")

    return result

def test_same_image_3ch(img1, img2):
    if img1 is None:
        print("test_same_image_3ch: img1 is None")

    if img2 is None:
        print("test_same_image_3ch: img2 is None")

    if img1.shape != img2.shape:
        print("img1.shape = ", img1.shape, " and img2.shape = ", img2.shape, ' are different')
```

```

return False

for x in range(0, img1.shape[0]):
    for y in range(0, img1.shape[1]):
        if img1[x,y,0] != img2[x,y,0] or img1[x,y,1] != img2[x,y,1] or img1[x,y,2] != img2[x,y,2]:
            print("Diff pixel", img1[x,y], img2[x,y])
            return False

return True

```

중간고사 시험 자료에 첨부된 이미지(들)을 중간고사 답안지를 작성하는 폴더를 기준으로 `"/2022-1-Midterm-SampleImages"` 라는 폴더를 만들고 주어진 이미지를 전부 새로 만든 폴더에 복사하여 사용하시오.

- 답안 작성에 필요한 이미지는 202201-Midterm-Lena.png 일 뿐이며, 나머지 영상은 입력으로 사용할 수 없음. 또한 이 이미지는 **input_image** 로 입력을 받도록 세팅되어 있으며, **문제 전체**에 사용되는 변수이므로 수정하지 말고, 문제 풀이에 그대로 사용해야 하면, 동일 이미지를 다시 읽지 말고 문제를 풀 것.
- **202201-Midterm-Lena.png 이외의 이미지를 임의로 입력할 경우, 전체 시험을 0점 처리.**
- 자세한 내용은 문제설명용 파워포인트 참조.

**** 만일 코딩 시 선분 또는 도형 요소가 필요한 경우, 선의 형태는 디폴트값을 사용할 것. 만일 다른 선 형태를 이용하여 채점 코딩에서 매칭하지 않는 경우 0점 처리. ****

다음의 이미지 경로는 변경할 수 없습니다. 경로 변경 시 받은 점수의 **0% 처리**.

다음 셀에 있는 코드는 수정하지 말고 수행만 할 것.

```

# DONOT CHANGE and Run this shell
org_image_path = sample_img_folder + "/202201-Midterm-Lena.png"

circle_rect_path = sample_img_folder + "/202201-Midterm-Circle-Rect.png"
prob1_and_result_path = sample_img_folder + "/202201-Midterm-Prob1-AND-Result.png"
prob1_or_result_path = sample_img_folder + "/202201-Midterm-Prob1-OR-Result.png"

up_roi_black_path = sample_img_folder + "/202201-Midterm-Prob2-1-Up-Black.png"
down_roi_black_path = sample_img_folder + "/202201-Midterm-Prob2-1-Down-Black.png"
up_down_roi_black_3ch_gray_path = sample_img_folder + "/202201-Midterm-Prob2-2-UpDown-Black-3CH-Gra

up_blue_down_red_combined_path = sample_img_folder + "/202201-Midterm-Prob3-UpBlue-DownRed.png"

final_res_path = sample_img_folder + "/202201-Midterm-Final-Res.png"

# grading list
prob_numbers = 10
grades_for_number = np.zeros(10, np.uint8)

def sum_of_scores():
    sum = 0;
    for score in grades_for_number:

```

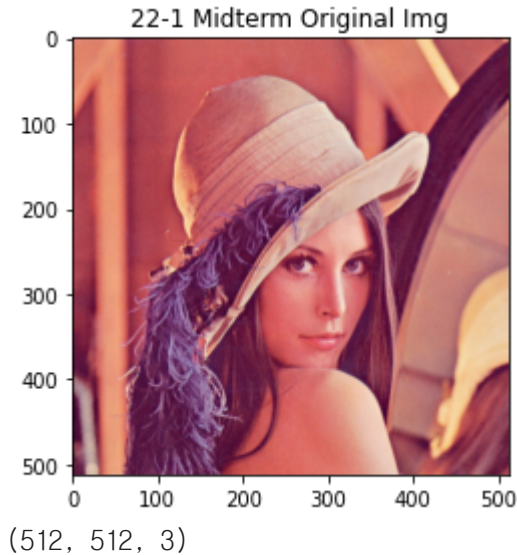
```

    sum = sum + score
    return sum

input_image = cv2.imread(org_image_path)
if input_image is None:
    print("image read failed")

# impute image test
show_with_matplotlib(input_image, '22-1 Midterm Original Img')
print(input_image.shape)

```



문제 1: 총 10점. AND case 10점, OR case 10점. 각 케이스별 부분 점수 없음.

다음의 셀에 함수 `copyRoiWithMask(img, mask, operation)` 를 작성하시오.

- AND operation test: 10점
- OR operation test: 10점

Parameters:

- `img`: the original 3-channel image
- `mask`: 3-channel roi
- `operation`: AND(and) or OR(or) operation

Return: 주어진 이미지에 마스크를 이용하여 주어진 동작의 결과

- 이 함수는 주어진 3 채널 이미지에 3 채널 마스크에 적용하며, 동작은 AND 또는 OR 을 행한다.
- 이 함수는 주어진 매개변수와 지역변수 외에 전역변수를 사용할 수 없음. 만일 사용 시, 해당 문제 0점 처리.

```

# Problem 1 Code Part: 여기에 코드를 작성하시오. 이 셀을 벗어난 코드는 인정하지 않음.
def copyRoiWithMask(img, mask, operation):

```

```

masks = cv2.threshold(mask, 1,255, cv2.THRESH_BINARY)[1] #roi 영역은 하양 그 외 영역은 검정
masks = cv2.split(masks)
fg_pass_mask = cv2.bitwise_or(masks[0],masks[1])
fg_pass_mask = cv2.bitwise_or(masks[2],fg_pass_mask)

if(operation=="AND"):

    return cv2.bitwise_and(img, img, mask=fg_pass_mask)

elif(operation=="OR"):

    bg_pass_mask = cv2.bitwise_not(fg_pass_mask)

    foreground = cv2.bitwise_and(mask, mask, mask=fg_pass_mask)
    background = cv2.bitwise_and(img, img, mask=bg_pass_mask)

    return cv2.add(foreground,background)

```

다음의 셀에 있는 코드는 검증 및 채점용 코드이므로 수정하지 말고 수행만 시키시오.

```

# DONOT CHANGE and Run this
# solution 1-1, 1-2 check

# compose an partial image for mask
circle_rect_img = cv2.imread(circle_rect_path, cv2.IMREAD_COLOR)
if circle_rect_img is None:
    print("circle_rect_img is None")

prob1_and_res = copyRoiWithMask(input_image, circle_rect_img, "AND")

# save the result for and op
prob1_and_sol_res = cv2.imread(prob1_and_result_path, cv2.IMREAD_COLOR)
if prob1_and_sol_res is None:
    print("prob1_and_sol_res is None")

# grading
cmp_res = test_same_image_3ch(prob1_and_sol_res, prob1_and_res)
if cmp_res:
    print("#1-1 is right + 10")
    grades_for_number[0] = 10
else:
    print("#1-1 is wrong + 0")

print("Your current scroe after 1-1: " + str(sum_of_scores()))

show_with_matplotlib(prob1_and_sol_res, "prob1-1 Solution")
show_with_matplotlib(prob1_and_res, "your result")

# grade 1-2
prob1_or_res = copyRoiWithMask(input_image, circle_rect_img, "OR")

```

```
# save the result for or op
print()
prob1_or_sol_res = cv2.imread(prob1_or_result_path, cv2.IMREAD_COLOR)
if prob1_or_sol_res is None:
    print("prob1_or_sol_res is None")

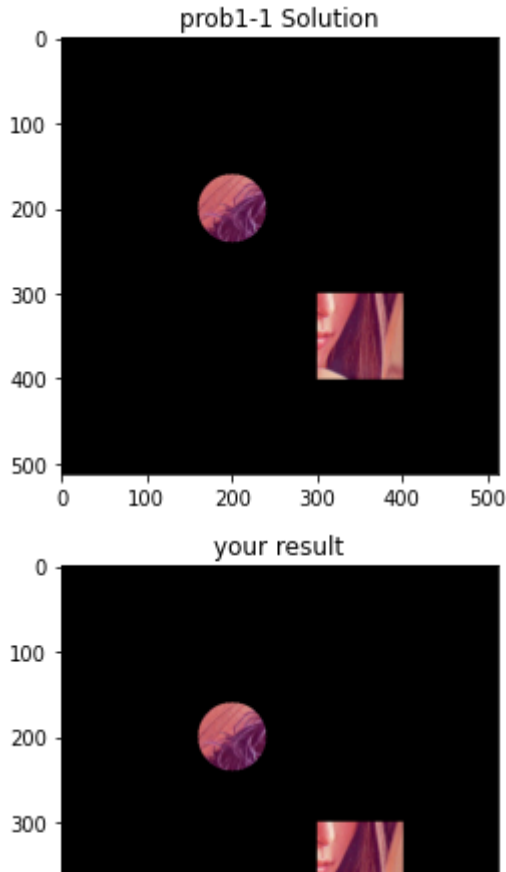
# grading
cmp_res = test_same_image_3ch(prob1_or_sol_res, prob1_or_res)
if cmp_res:
    print("#1-2 is right + 10")
    grades_for_number[1] = 10
else:
    print("#1-2 is wrong + 0")

print("Your current score after 1-2: " + str(sum_of_scores()))

show_with_matplotlib(prob1_or_sol_res, "prob1-2 Solution")
show_with_matplotlib(prob1_or_res, "your result")
```

#1-1 is right + 10

Your current score after 1-1: 20



▼ 가정

주어진 입력 이미지의 4 코너를 다음과 명명한다: Corner1(왼쪽 위), Corner2(오른쪽 위), Corner3(왼쪽 아래), Corner4(오른쪽 아래)

주어진 입력 이미지의 y 축과 x 축의 중심에 위치한 점을 Center 라고 명명한다.

- 입력 이미지의 크기가 (512, 512) 이므로 Center 는 (256, 256) 에 위치한다고 가정한다.

주어진 입력 이미지에 대각선 2개를 설정하여 4 등분을 가정하고 각 영역의 이름은 다음과 정의한다.

- up_roi: 위쪽 삼각형 영역
- right_roi: 오른쪽 삼각형 영역
- down_roi: 아래쪽 삼각형 영역
- left_roi: 왼쪽 삼각형 영역

위의 영역들로 이미지를 분할할 때, 픽셀 단위에서는 선 (Corner1, Center), 선 (Corner2, Center), 선 (Corner3, Center), 선 (Corner4, Center) 에 위치하는 픽셀은 두개의 영역에 중복하여 포함되므로, 각 선에 포함되는 픽셀들은 각기 up_roi, down_roi 에 포함된다고 가정한다.

문제 2-1: 10점 부분점수 없음

다음의 코드 셀에 위에서 정의에 따라, 주어진 입력 이미지와 크기가 동일한 다음의 두개의 3채널 마스크를 작성하시오. 변수 이름은 다음과 같이 사용하시오.

- up_roi_black: up_roi 가 블랙, 나머지는 화이트인 영상
- down_roi_black: down_roi 가 블랙, 나머지는 화이트인 영상



Provide solution for problem 2-1 here: 여기에 코드를 작성하시오. 이 셀을 벗어난 코드는 인정하지 않

```
up_roi_black = np.zeros((512,512,3), dtype = np.uint8)+255
```

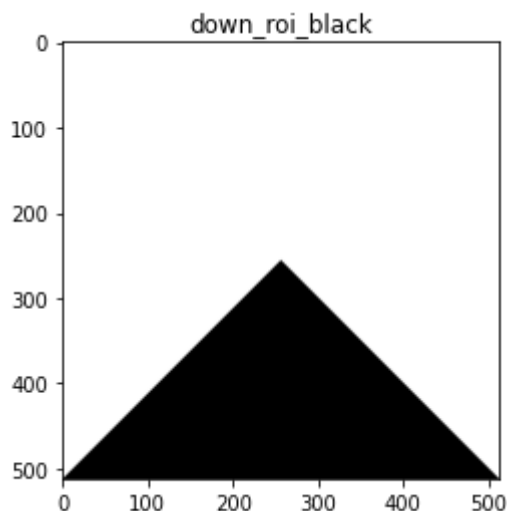
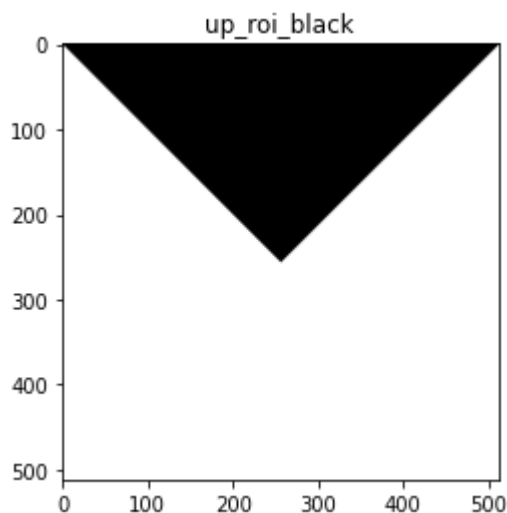
```
for i in range(257):
    up_roi_black[i,i:512-i+1,:] = 0
```

```
show_with_matplotlib(up_roi_black,"up_roi_black")
```

```
down_roi_black = np.zeros((512,512,3), dtype = np.uint8)+255
```

```
for i in range(256):
    down_roi_black[511-i,i+1:512-i,:] = 0
```

```
show_with_matplotlib(down_roi_black,"down_roi_black")
```



다음의 코드 블록은 문제 2-1 을 채점하는 코드이니 수정하지 말고 수행하시오.

```
# DONOT CHANGE and Run this
# Grading 2-1
up_roi_sol_img = cv2.imread(up_roi_black_path)
if up_roi_sol_img is None:
    print("up_roi_sol_img is None")

down_roi_sol_img = cv2.imread(down_roi_black_path)
if down_roi_sol_img is None:
    print("down_roi_sol_img is None")

# grading
cmp_res_up = False
cmp_res_down = False

if up_roi_black is not None:
    cmp_res_up = test_same_image_3ch(up_roi_sol_img, up_roi_black)

if down_roi_black is not None:
    cmp_res_down = test_same_image_3ch(down_roi_sol_img, down_roi_black)

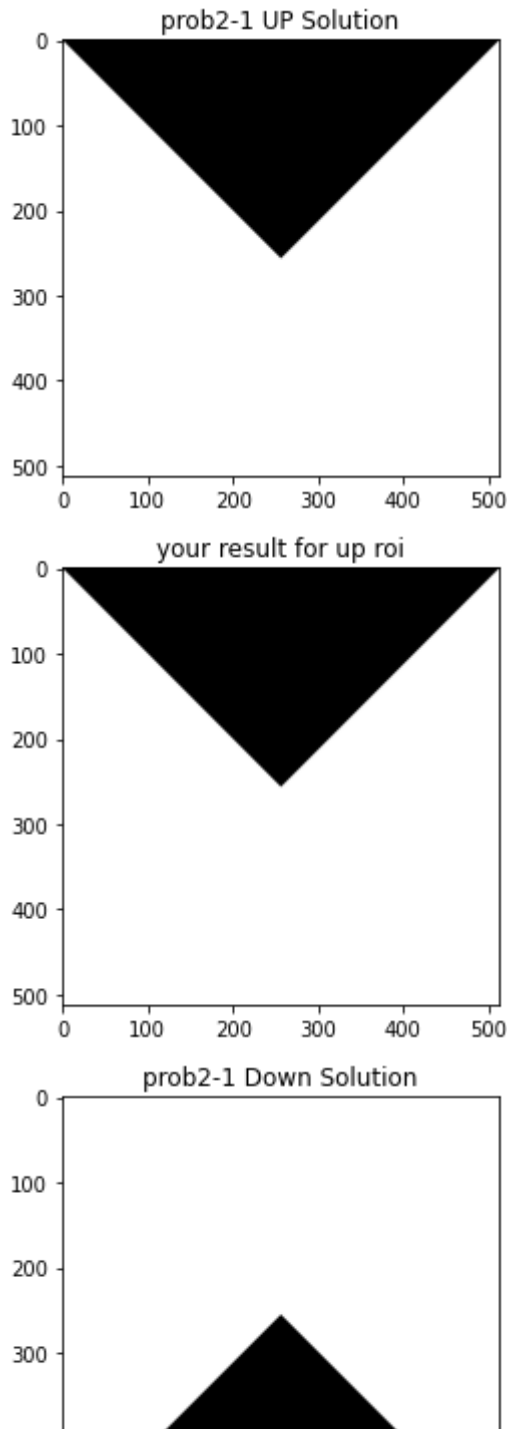
if cmp_res_up and cmp_res_down:
    print("#2-1 is right + 10")
    grades_for_number[2] = 10
else:
    print("#2-1 is wrong + 0")

print("Your current score after 2-1: " + str(sum_of_scores()))

show_with_matplotlib(up_roi_sol_img, "prob2-1 UP Solution")
show_with_matplotlib(up_roi_black, "your result for up roi")
show_with_matplotlib(down_roi_sol_img, "prob2-1 Down Solution")
show_with_matplotlib(down_roi_black, "your result for down roi")
```

#2-1 is right + 10

Your current score after 2-1: 30



▼ 문제 2-2: 10점 부분점수 없음

위에서 만든 `up_roi_black` 과 `down_roi_black` 을 이용하여 원본 이미지의 3-채널 그레이스케일 이미지에서 `up_roi` 와 `down_roi` 부분이 블랙으로 지워진 영상을 변수 `left_right_3ch_gray` 에 배정하십시오.

변수 이름을 주어진대로 사용하지 않는 경우, 0점 처리.

마스크를 이용하는 부분은 위에서 정의한 `copyRoiWithMask` 를 사용하여야 하며 임의로 유사 동작을 하는 코드를 사용하는 경우, 0점 처리.

Code for Solution 2-2: 여기에 코드를 작성하십시오. 이 셀을 벗어난 코드는 인정하지 않음.

다음 코드 블록은 문제 2-2 채점 코드이므로 수정하지 말고 수행하시오.

```
# DONOT CHANGE and Run this
# grading 2-2

prob_2_2_sol_image = cv2.imread(up_down_roi_black_3ch_gray_path)
if prob_2_2_sol_image is None:
    print("prob_2_2_sol_image is None")

# grading
cmp_up_down_black = False
if left_right_3ch_gray is not None:
    cmp_up_down_black = test_same_image_3ch(prob_2_2_sol_image, left_right_3ch_gray)

if cmp_up_down_black:
    print("#2-2 is right + 10")
    grades_for_number[3] = 10
else:
    print("#2-2 is wrong + 0")

print("Your current score after 2-2: " + str(sum_of_scores()))

show_with_matplotlib(prob_2_2_sol_image, "prob2-2 Solution")
show_with_matplotlib(left_right_3ch_gray, "your result for 2-2")
```

▼ 문제 3: 30점 부분점수 없음

위에서 만든 up_roi_black 과 down_roi_black 을 이용하여 원본 이미지의 분리된 색채널을 다음과 같이 left_right_3ch_gray 에 합성하여라.

1. 원본 이미지의 blue 채널 중 up_roi 에 해당하는 부분을 left_right_3ch_gray 동일 부분에 합성
2. 원본 이미지의 red 채널 중 down_roi 에 해당하는 부분을 left_right_3ch_gray 동일 부분에 합성
3. 결과를 left_right_gray_blue_red 에 저장

마스크를 이용하는 부분은 위에서 정의한 copyRoiWithMask 를 사용하여야 하며 임의로 유사 동작을 하는 코드를 사용하는 경우, 0점 처리.

다음 코드 셀에 솔루션을 제공하시오.

```
# Code for Problem 3: 여기에 코드를 작성하시오. 이 셀을 벗어난 코드는 인정하지 않음.
```

다음의 코드 블록은 문제 3을 위한 채점용이니 수정하지 말고 수행하시오.

```
# DONOT CHANGE and Run this
```

```
# Prob 3 Grading
prob_3_sol_img = cv2.imread(up_blue_down_red_combined_path)
if prob_3_sol_img is None:
    print("prob_3_sol_img is None")

prob_3_cmp_res = False
if left_right_3ch_gray is not None:
    prob_3_cmp_res = test_same_image_3ch(prob_3_sol_img, left_right_gray_blue_red)

if prob_3_cmp_res:
    print("#3 is right + 20")
    grades_for_number[4] = 30
else:
    print("#3 is wrong + 0")

print("Your current score after 3: " + str(sum_of_scores()))

show_with_matplotlib(prob_3_sol_img, "prob3 Solution")
show_with_matplotlib(left_right_gray_blue_red, "your result for 3")
```

▼ *문제 4: 30점 부분 점수 없음*

- 주어진 입력 이미지와 같은 크기의 블랙 공백 이미지에 다음의 값을 이용하여 텍스트를 출력하라.
 - 텍스트: "ImgProc Midterm Test!!!"
 - 텍스트 원점: (x, y) = (40, 252)
 - 폰트: FONT_HERSHEY_SIMPLEX
 - 텍스트 색: 흰색
 - 텍스트 굵기: 5
- 출력된 이미지를 수평 방향으로 뒤집고, 즉 원래 이미지와 뒤집힌 이미지는 원래 이미지의 오른쪽 y 축을 중심으로 미러링되는 형태가 된다.
- 원래 입력 이미지 중, 뒤집어진 이미지의 텍스트 부분을 위에서 구한 left_right_gray_blue_red 와 합성하시오. 즉, 최종 이미지는 위에서 구한 결과에 뒤집힌 텍스트 부분이 원래 이미지 부분이 보이는 형태가 된다. 최종 결과는 변수 final_res 에 저장하시오.

Code for Problem 4: 여기에 코드를 작성하시오. 이 셀을 벗어난 코드는 인정하지 않음.

다음의 셀에 있는 코드는 검증 및 채점용 코드이므로 수정하지 말고 수행만 시키시오

```
# DONOT CHANGE and Run this
# grading 4
prob_4_sol_img = cv2.imread(final_res_path)

prob_4_cmp_res = test_same_image_3ch(prob_4_sol_img, final_res)
```

```
if prob_4_cmp_res:
    print("#4 is right + 20")
    grades_for_number[5] = 30
else:
    print("#4 is wrong + 0")

print("Your current score after 4: " + str(sum_of_scores()))

show_with_matplotlib(prob_4_sol_img, "prob4 Solution")
show_with_matplotlib(final_res, "your result for 4")
```

✓ 1초 오후 2:28에 완료됨

