

01. Face

CODE

```
import cv2
import numpy as np

face_xml_path =
"./22.12.08_d47_image/data/haarcascade_frontalface_default.xml"
eye_xml_path = "./22.12.08_d47_image/data/haarcascade_eye.xml"
image_path1 = "./22.12.08_d47_image/data/face.png"
image_path2 = "./22.12.08_d47_image/data/iksu.jpg"

#### 1. face_cascade & eye_cascade objects 생성
face_cascade = cv2.CascadeClassifier(face_xml_path)
eye_cascade = cv2.CascadeClassifier(eye_xml_path)

#### 2. 얼굴 데이터
image = cv2.imread(image_path1)
cv2.imshow("1. Original", image)
cv2.waitKey(0)

#### 3. 얼굴 감지 바운딩 박스
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY) # gray
faces = face_cascade.detectMultiScale(gray_image, 1.1, 4) # face variable
# detectMultiScale(그레이 이미지, 축소할 이미지 배율 인수, 이웃의 최소 수)
for (x, y, w, h) in faces: # 얼굴 주변 rectangle 정의
    cv2.rectangle(image, (x, y), (x+w, y+h), (0,255,0), 3)
cv2.imshow("2. Face", image)
cv2.waitKey(0)

#### 4. 눈 감지 바운딩 박스
roi_gray = gray_image[y:(y+h), x:(x+w)]
roi_image = image[y:(y+h), x:(x+w)]
eyes = eye_cascade.detectMultiScale(roi_gray, 1.1, 4) # eyes variable
index = 0
for (ex, ey, ew, eh) in eyes: # 눈 두개 분리
    if index == 0:
        eye_1 = (ex, ey, ew, eh)
    elif index == 1:
        eye_2 = (ex, ey, ew, eh)
    # 눈 주변 rectangle 정의
    cv2.rectangle(roi_image, (ex,ey), (ex+ew, ey+eh), (0,0,255), 3)
    index += 1
cv2.imshow("3. Eyes", image)
# cv2.imwrite("./22.12.07_d46_image/data/man_cascade.png", image)
cv2.waitKey(0)
```

```

#### 5. 좌우 눈 설정
if eye_1[0] < eye_2[0]:
    left_eye = eye_1
    right_eye = eye_2
else:
    left_eye = eye_2
    right_eye = eye_1

#### 6. 직사각형 중심점의 좌표 계산
left_eye_center = (int(left_eye[0] + (left_eye[2]/2)), int(left_eye[1] +
(left_eye[3]/2)))
left_eye_x = left_eye_center[0]
left_eye_y = left_eye_center[1]

right_eye_center = (int(right_eye[0] + (right_eye[2]/2)), int(right_eye[1] +
(right_eye[3]/2)))
right_eye_x = right_eye_center[0]
right_eye_y = right_eye_center[1]

cv2.line(roi_image, right_eye_center, left_eye_center, (0,200,200), 3)
cv2.circle(roi_image, left_eye_center, 5, (255,0,0), -1)
cv2.circle(roi_image, right_eye_center, 5, (255,0,0), -1)

cv2.imshow("4. Center Eyes", image)
cv2.waitKey(0)

#### 7. 두 눈의 중심점을 연결하는 선 사이의 각도 계산
if left_eye_y > right_eye_y:
    A = (right_eye_x, left_eye_y)
    direction = -1 # clockwise 로 회전
else:
    A = (left_eye_y, right_eye_x)
    direction = 1 # counterclockwise 로 회전

cv2.line(roi_image, right_eye_center, left_eye_center, (0,200,200), 3)
cv2.line(roi_image, left_eye_center, A, (0,200,200), 3)
cv2.line(roi_image, right_eye_center, A, (0,200,200), 3)
cv2.circle(roi_image, A, 5, (255,0,0), -1)

cv2.imshow("5. Center Eyes Triangle", image)
cv2.waitKey(0)

#### 8. 각도 구하기
# np.arctan = 함수 단위는 라디안
# 라디안 -> 각도 : (theta * 100) / np.pi
delta_x = right_eye_x - left_eye_x
delta_y = right_eye_y - left_eye_y
angle = np.arctan(delta_y/delta_x)

```

```

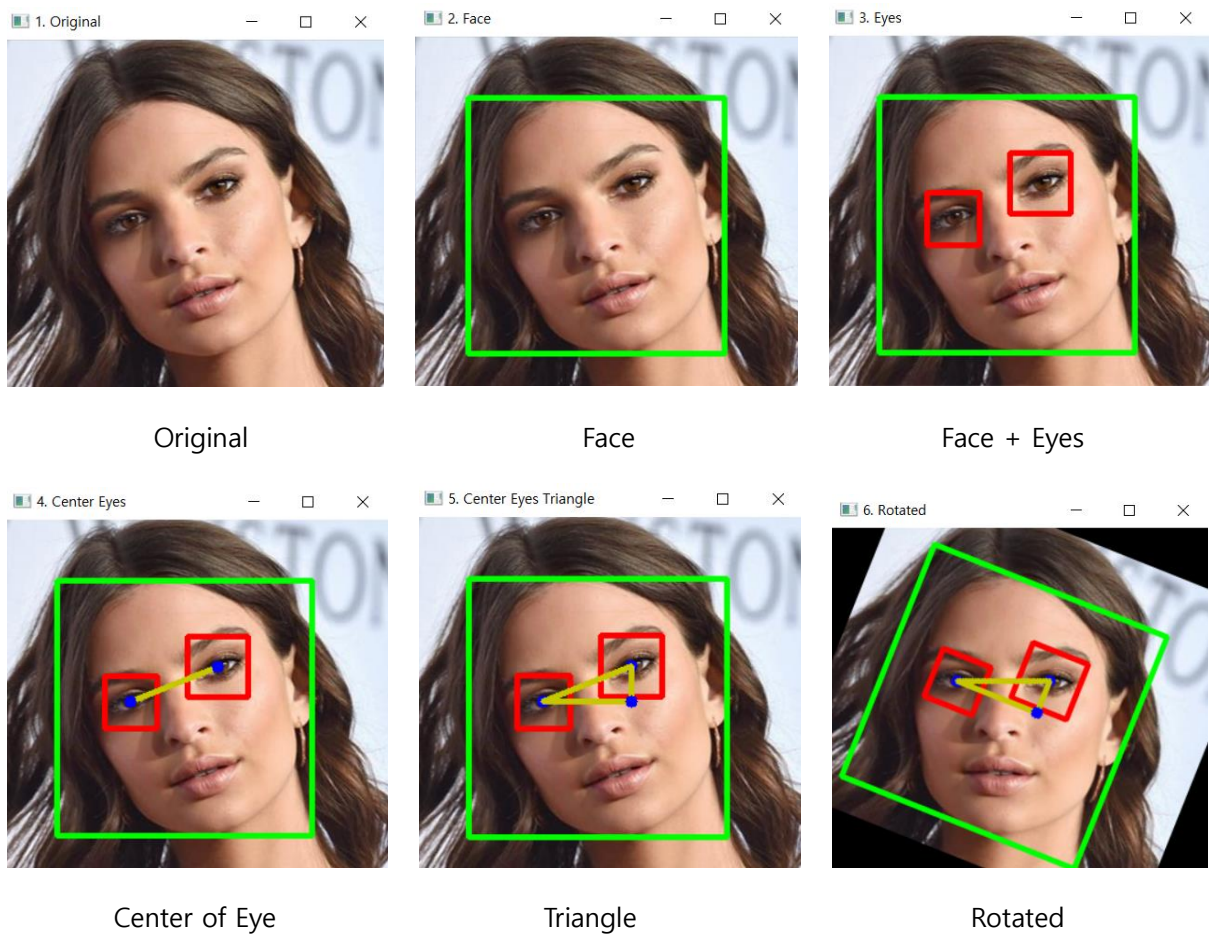
angle = (angle*180) / np.pi # 각도 (약 -21 도)

#### 9. 회전 하기
h, w = image.shape[:2]
center = (w//2, h//2)
M = cv2.getRotationMatrix2D(center, (angle), 1.0) # 회전
rotated = cv2.warpAffine(image, M, (w, h))

cv2.imshow("6. Rotated", rotated)
cv2.waitKey(0)

```

RESULT



02. Video

CODE

```
import cv2

## 1. 동영상 속성 확인
video_path = "./22.12.08_d47_image/data/video01.mp4"
cap = cv2.VideoCapture(video_path)
width = cap.get(cv2.CAP_PROP_FRAME_WIDTH)
height = cap.get(cv2.CAP_PROP_FRAME_HEIGHT)
frame_count = cap.get(cv2.CAP_PROP_FRAME_COUNT)
fps = cap.get(cv2.CAP_PROP_FPS)

print("Width: ", width, ", Height: ", height)
print("Frame Count: ", frame_count)
print("fps: ", fps)

## 2. 동영상 파일 읽기
if cap.isOpened(): # 캡처 객체 초기화 확인
    while True:
        ret, frame = cap.read()
        if ret:
            cv2.imshow("Video File Show", frame)
            cv2.waitKey(25) # 25 fps 기준으로 프레임 나눠
        else:
            break
    else:
        print("비디오 파일 읽기 실패")

cap.release()
cv2.destroyAllWindows()
```

RESULT

```
Width: 1280.0 , Height: 720.0
Frame Count: 323.0
fps: 25.0
```

03. Video Cut

CODE

```
import cv2
import os

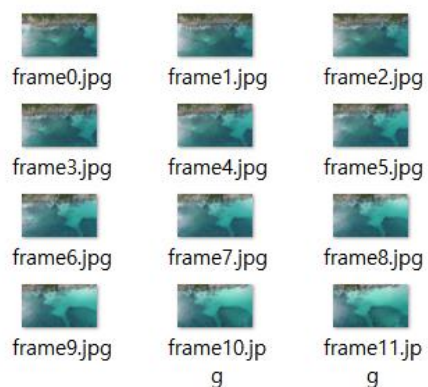
## 1. 동영상 속성 확인
video_path = "./22.12.08_d47_image/data/video01.mp4"
cap = cv2.VideoCapture(video_path)
fps = cap.get(cv2.CAP_PROP_FPS)
count = 0

## 2. 동영상 파일 읽기
if cap.isOpened():
    while True:
        ret, frame = cap.read()
        if ret:
            if (int(cap.get(1)) % fps == 0): # fps 값을 사용하여 1 초마다 추출
                os.makedirs("./22.12.08_d47_image/data/frame_image_save",
exist_ok=True)
                cv2.imwrite("./22.12.08_d47_image/data/frame_image_save/" +
"frame%d.jpg" %
                        count, frame)
                print("save frame number >> ", str(int(cap.get(1))))
                count += 1
            else:
                break
        else:
            print("비디오 열기 실패")

cap.release()
cv2.destroyAllWindows()
```

RESULT

```
save frame number >> 25
save frame number >> 50
save frame number >> 75
save frame number >> 100
save frame number >> 125
save frame number >> 150
save frame number >> 175
save frame number >> 200
save frame number >> 225
save frame number >> 250
save frame number >> 275
save frame number >> 300
```



04. Pydicom

CODE

```
import pydicom
import matplotlib.pyplot as plt
import numpy as np
from pydicom.pixel_data_handlers.util import apply_modality_lut, apply_voi_lut

file_path = "./22.12.08_d47_image/data/ID_0000_AGE_0060_CONTRAST_1_CT.dcm"
window_center = -600
window_width = 1600

# DICOM 파일 읽는 함수 -> pydicom.read_file(), pydicom.dcmread()
slice = pydicom.read_file(file_path)
print(slice)

s = int(slice.RescaleSlope)
b = int(slice.RescaleIntercept)
image = s * slice.pixel_array + b

plt.subplot(1,3,1)
plt.title("DICOM -> Array")
plt.imshow(image, cmap= 'gray')

slice.WindowCenter = window_center
slice.WindowWidth = window_width
image = apply_modality_lut(image, slice)
image2 = apply_voi_lut(image, slice)

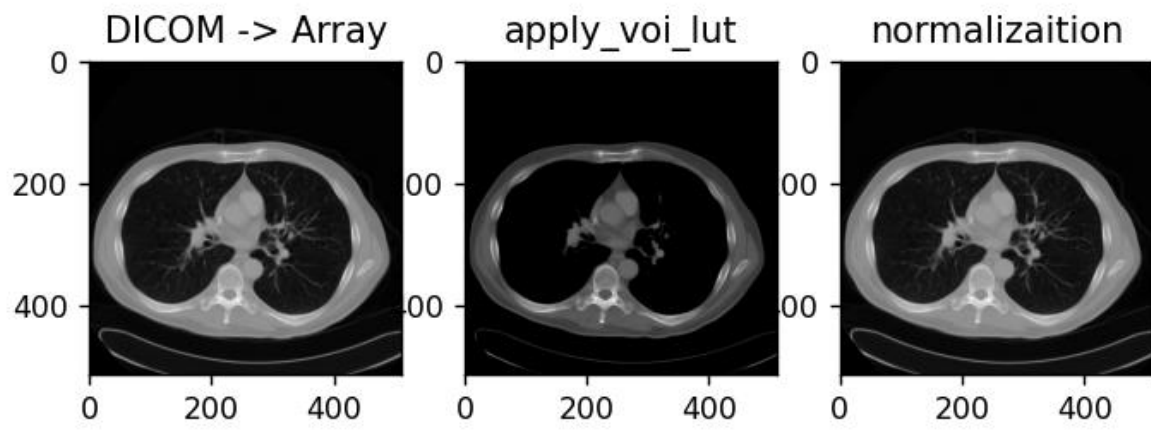
plt.subplot(1, 3, 2)
plt.title("apply_voi_lut")
plt.imshow(image2, cmap='gray')

# Normalization
image3 = np.clip(image, window_center-(window_width/2),
                  window_center+(window_width/2))

plt.subplot(1,3,3)
plt.title("normalizaition")
plt.imshow(image, cmap= 'gray')
plt.show()

# 참고사항 : https://pydicom.github.io/pydicom/stable/auto\_examples/index.html
#           https://pydicom.github.io/pydicom/stable/tutorials/dataset\_basics.html
```

RESULT



exp_01. 이미지 블렌딩 (addWeighted)

CODE

```
# 같은 크기의 이미지 블렌딩 실험
import cv2
import matplotlib.pyplot as plt
import numpy as np

large_img = cv2.imread("./22.12.08_d47_image/data/1.png")
watermark = cv2.imread("./22.12.08_d47_image/data/2.png")

# print("Large Image Size: ", large_img.shape)
# print("Watermark Image Size: ", watermark.shape)
"""
Large Image Size:  (683, 1024, 3)
Watermark Image Size:  (480, 640, 3)
"""

img1 = cv2.resize(large_img, (800,600) )
img2 = cv2.resize(watermark, (800,600) )

# print("Image 1 Resize: ", img1.shape)
# print("Image 2 Resize: ", img2.shape)
"""
Image 1 Resize:  (600, 800, 3)
Image 2 Resize:  (600, 800, 3)
"""

## 혼합 진행
blended = cv2.addWeighted(img1, 0.8, img2, 0.2, 0)
# img1 은 0.9 만큼, img2 0.1 만큼 진하게

# 1로 설정
# blended = cv2.addWeighted(img1, 1, img2, 1, 0)

cv2.imshow("Image Large", blended)
cv2.waitKey(0)
```


RESULT

Image Large

— □ ×



exp_02. 비트 연산

CODE

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

# ex-01
image_rectangle = np.ones((400,400), dtype= 'uint8')
cv2.rectangle(image_rectangle, (50,50), (300,300), (255,255,255), -1)
plt.figure(figsize=(15,15))
plt.subplot(2,3,1)
plt.title("Rectangle")
plt.imshow(image_rectangle, cmap= 'gray')

## ex-02
img_circle = np.ones((400,400), dtype= 'uint8')
cv2.circle(img_circle, (300, 300), 70, (255,255,255), -1)
plt.subplot(2,3,2)
plt.title("Circle")
plt.imshow(img_circle, cmap= 'gray')

## ex-03
bitwiseAnd = cv2.bitwise_and(image_rectangle, img_circle)
plt.subplot(2,3,3)
plt.title("Bitwise And")
plt.imshow(bitwiseAnd, cmap= 'gray')

bitwiseOr = cv2.bitwise_or(image_rectangle, img_circle)
plt.subplot(2,3,4)
plt.title("Bitwise Or")
plt.imshow(bitwiseOr, cmap= 'gray')

bitwiseXor = cv2.bitwise_xor(image_rectangle, img_circle)
plt.subplot(2,3,5)
plt.title("Bitwise Xor")
plt.imshow(bitwiseXor, cmap= 'gray')

rec_not = cv2.bitwise_not(image_rectangle, img_circle)
plt.subplot(2,3,6)
plt.title("Bitwise Not")
plt.imshow(rec_not, cmap= 'gray')
plt.show()
```

```

# ex-04 마스킹 과제는 흰색대신 이미지를 넣어주시면 됩니다. (원하는 이미지 혹은
얼굴이미지)
face_xml_path =
"./22.12.08_d47_image/data/haarcascade_frontalface_default.xml"
image = cv2.imread("./22.12.08_d47_image/data/faces.jpg")
cv2.imshow("Original Image", image)
cv2.waitKey(0)

face_cascade = cv2.CascadeClassifier(face_xml_path) # face_cascade objects
생성
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY) # gray
faces = face_cascade.detectMultiScale(gray_image, 1.1, 4) # face variable

crop_faces = []
for (x, y, w, h) in faces: # 얼굴 주변 rectangle 정의
    crop_faces.append(image[y:(y+h), x:(x+w)])

mask = np.zeros((683, 1024, 3), dtype='uint8')
cv2.rectangle(mask, (60, 50), (280, 280), (255, 255, 255), -1)
cv2.rectangle(mask, (420, 50), (550, 230), (255, 255, 255), -1)
cv2.rectangle(mask, (750, 50), (920, 280), (255, 255, 255), -1)
cv2.imshow("test", mask)
cv2.waitKey(0)

# pt1, pt2 : 사각형의 두 꼭지점 좌표. (x, y) 튜플
# (x, y, w, h)

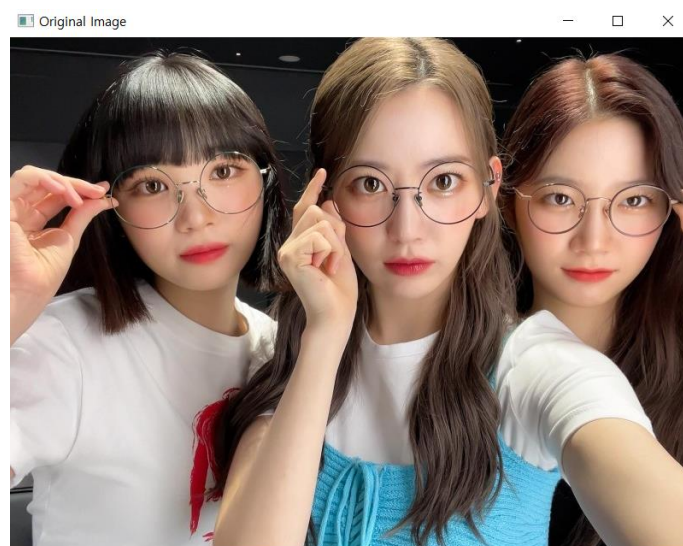
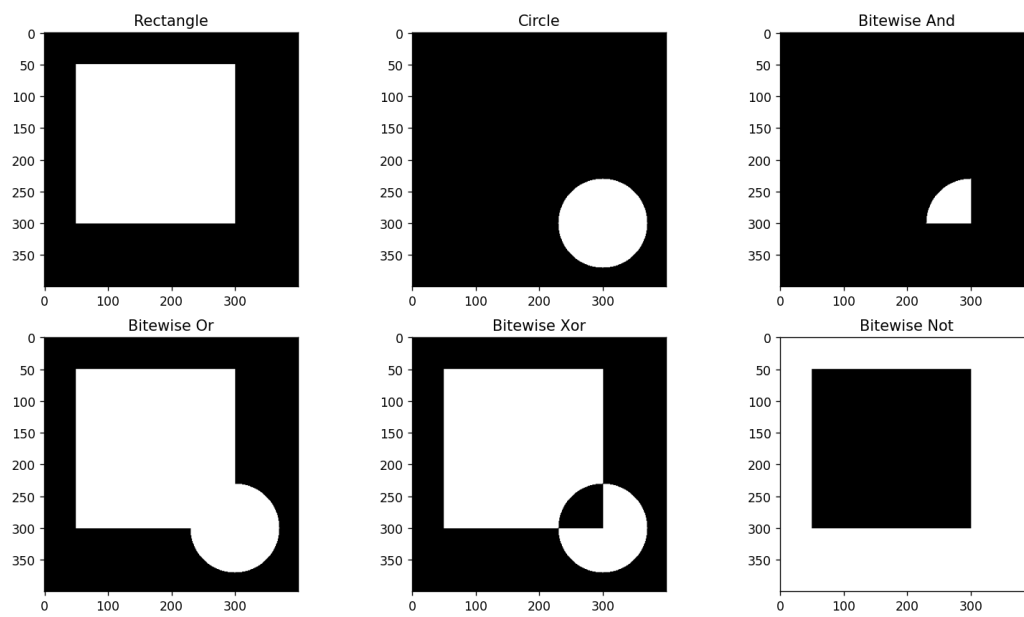
x_offset = [60, 420, 750]
y_offset = [50, 50, 50]

x_end = [280, 550, 920]
y_end = [280, 230, 280]
resize_crop_faces = []

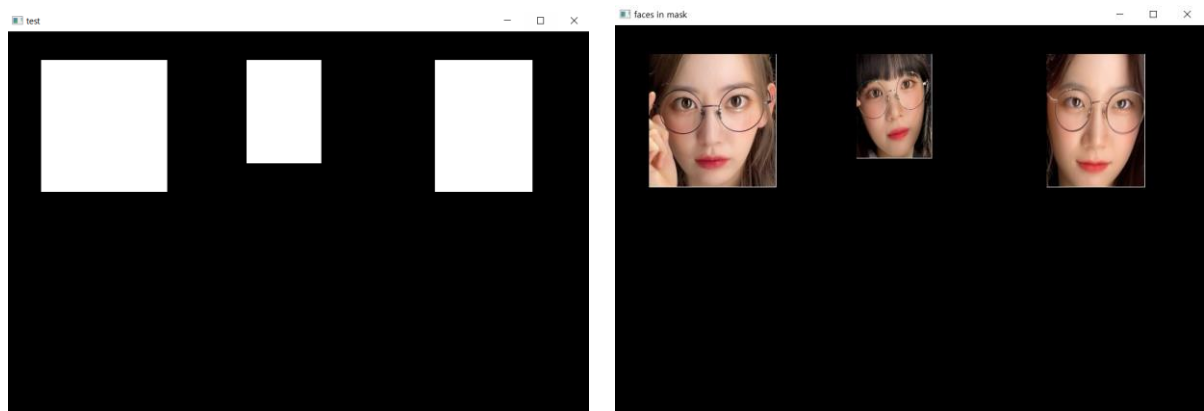
for i in range(len(crop_faces)):
    # resize
    resize_crop_faces.append(cv2.resize(crop_faces[i],
                                         (x_end[i]-x_offset[i], y_end[i]-
                                         y_offset[i])))
    # add resized crop faces in mask
    mask[y_offset[i]: y_offset[i] + resize_crop_faces[i].shape[0],
        x_offset[i] : x_offset[i] + resize_crop_faces[i].shape[1]] =
    resize_crop_faces[i]
    print(mask[y_offset[i]: y_offset[i] + resize_crop_faces[i].shape[0],
        x_offset[i] : x_offset[i] + resize_crop_faces[i].shape[1]].shape,
    resize_crop_faces[i].shape)
cv2.imshow("faces in mask", mask)
cv2.waitKey(0)

```

RESULT



Original



Mask

Face in Mask

exp_03. 블렌딩 없이 오버레이

CODE

```
import cv2
import numpy as np

large_img = cv2.imread("./22.12.08_d47_image/data/1.png")
watermark = cv2.imread("./22.12.08_d47_image/data/2.png")

small_img = cv2.resize(watermark, (300,300))

x_offset = 30
y_offset = 170

x_end = x_offset + small_img.shape[0]
y_end = y_offset + small_img.shape[1]

large_img[y_offset:y_end, x_offset:x_end] = small_img
cv2.imshow("test", large_img)
cv2.waitKey(0)
```

RESULT



exp_04. 블렌딩 + exp_03

CODE

```
import cv2
import numpy as np

large_img = cv2.imread("./22.12.08_d47_image/data/1.png")
watermark = cv2.imread("./22.12.08_d47_image/data/2.png")
small_img = cv2.resize(watermark, (300,300))

x_offset = 400
y_offset = 170

rows, columns, channels = small_img.shape
roi = large_img[y_offset:470, x_offset:700]

# logo image 빨간색 부분을 제외한 모든 것을 필터링 하도록 -> 회색조 이미지로 변경
small_img_gray = cv2.cvtColor(small_img, cv2.COLOR_BGR2GRAY)
ret, mask = cv2.threshold(small_img_gray, 120, 255, cv2.THRESH_BINARY)
cv2.imshow("Mask", mask)
cv2.waitKey(0)

bg = cv2.bitwise_or(roi, roi, mask= mask)
mask_inv = cv2.bitwise_not(mask)
cv2.imshow("Mask Inversed", mask_inv)
cv2.waitKey(0)

fg = cv2.bitwise_and(small_img, small_img, mask= mask_inv)
cv2.imshow("Bitwise", fg)
cv2.waitKey(0)

final_roi = cv2.add(bg, fg)
cv2.imshow("ROI(Region Of Interest)", final_roi)
cv2.waitKey(0)
```


RESULT



Mask



Mask Inversed



Bitwise



ROI