**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\_casecase & eye\_cascase 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  # clockwise로 회전

else:

    A = (left\_eye\_y, right\_eye\_x)

    direction = 1   # coutner clockwise로 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 -21도)

#### 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**

텍스트, 사람, 여자, 숙녀이(가) 표시된 사진

자동 생성된 설명사람, 소녀, 녹색, 다채로운이(가) 표시된 사진

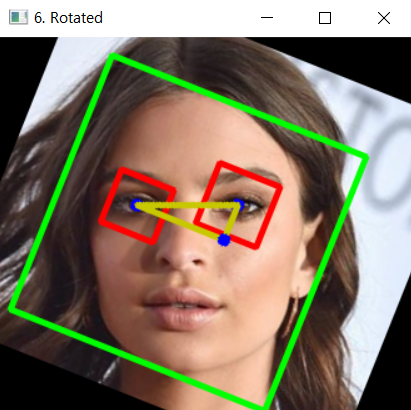
자동 생성된 설명사람, 실외, 녹색, 다채로운이(가) 표시된 사진

자동 생성된 설명

Original Face Face + Eyes

사람, 실외, 녹색, 오렌지이(가) 표시된 사진

자동 생성된 설명사람, 실외, 가발, 다채로운이(가) 표시된 사진

자동 생성된 설명

Center of Eye Triangle Rotated

**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**



**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 1초마다 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:

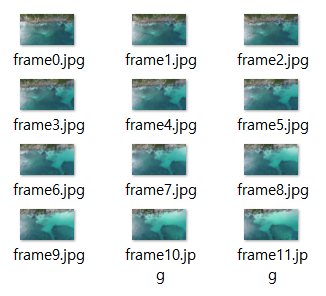
    print("비디오 열기 실패")

cap.release()

cv2.destroyAllWindows()

**RESULT**

텍스트, 전자기기, 키보드이(가) 표시된 사진

자동 생성된 설명

**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")

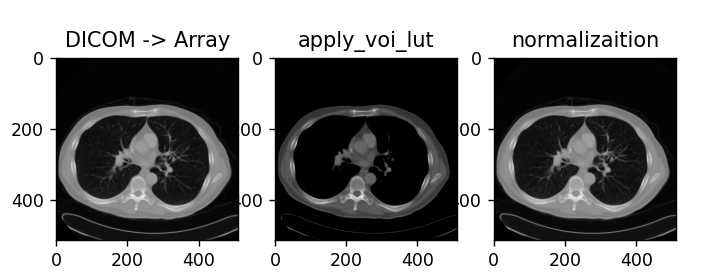
plt.imshow(image, cmap= 'gray')

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)

# img1은 0.9만큼, img2 0.1만큼 진하게

# 1로 설정

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

cv2.imshow("Image Large", blended)

cv2.waitKey(0)

**RESULT**

테니스, 사람, 스포츠, 채이(가) 표시된 사진

자동 생성된 설명

**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("Bitewise And")

plt.imshow(bitwiseAnd, cmap= 'gray')

bitwiseOr  = cv2.bitwise\_or(image\_rectangle, img\_circle)

plt.subplot(2,3,4)

plt.title("Bitewise Or")

plt.imshow(bitwiseOr, cmap= 'gray')

bitwiseXor  = cv2.bitwise\_xor(image\_rectangle, img\_circle)

plt.subplot(2,3,5)

plt.title("Bitewise Xor")

plt.imshow(bitwiseXor, cmap= 'gray')

rec\_not = cv2.bitwise\_not(image\_rectangle, img\_circle)

plt.subplot(2,3,6)

plt.title("Bitewise 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\_casecase 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]

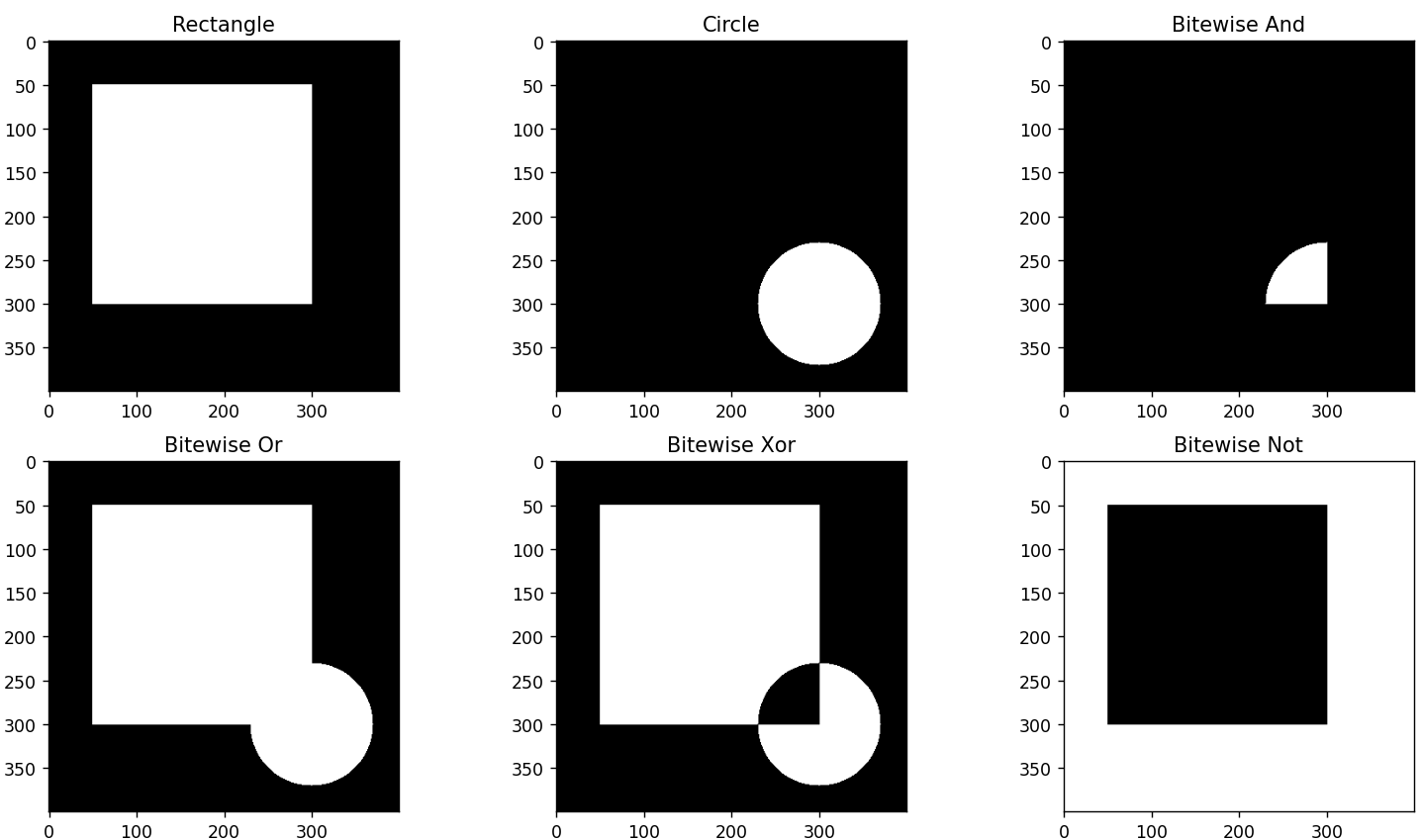
    print(mask[y\_offset[i]: y\_offset[i] + resize\_crop\_faces[i].shape[0],

        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)

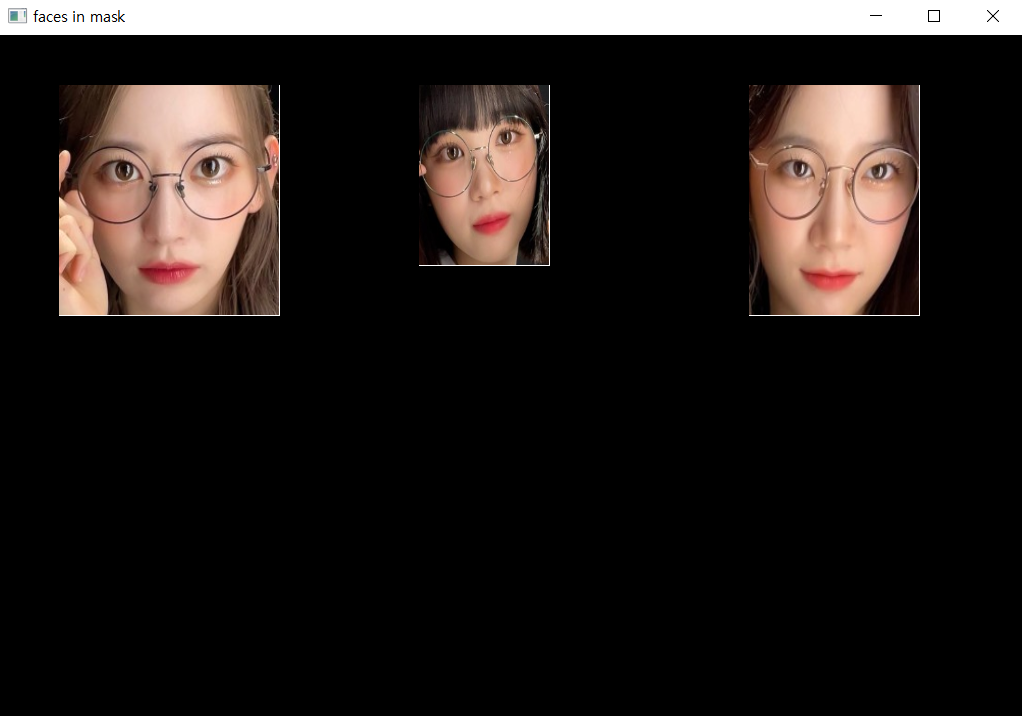
**RESULT**



사람, 가장, 그룹이(가) 표시된 사진

자동 생성된 설명

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**

사람, 테니스, 스포츠, 운동경기이(가) 표시된 사진

자동 생성된 설명

**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 필토링 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**

텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명

Mask Mask Inversed

텍스트이(가) 표시된 사진

자동 생성된 설명 텍스트이(가) 표시된 사진

자동 생성된 설명

Bitwise ROI