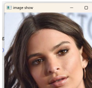
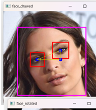
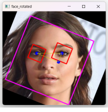
1208 > 01face얼굴정렬.py > ...

```python
import cv2
import numpy as np

# Creating face_cascade and eye_cascade objects
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
eye_cascade=cv2.CascadeClassifier('haarcascade_eye.xml')

# Lodaing the image - 얼굴 이미지 데이터 읽기
img = cv2.imread('./face.png')
cv2.imshow('image show',img)
cv2.waitKey(0)

# Converting the image into grayscale 그레이로 바꿔줌
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

faces = face_cascade.detectMultiScale(gray, 1.1, 4)

# Defining and drawing the rectangles around the face
for (x,y,w,h) in faces:
    cv2.rectangle(img, (x,y), (x+w, y+h), (255, 0, 255), 2 )
                                            # 색상      # 선의굵기

roi_gray = gray[y:(y+h), x:(x+w)]
roi_color = img[y:(y+h), x:(x+w)]

# eyes
eyes = eye_cascade.detectMultiScale(roi_gray, 1.1, 4)
print(eyes)
'''
2개의 좌표가 잡혀있음
[[41 82 45 45]
 [112 48 52 52]]
'''
index = 0

# Creating for loop in order to divide one eye from another
for (ex, ey, ew, eh) in eyes :
    if index == 0:
        eye_1 = (ex, ey, ew, eh)
    elif index == 1:
        eye_2 = (ex, ey, ew, eh)

    cv2.rectangle(roi_color, (ex, ey), (ex+ew, ey+eh), (0,0,255), 2)
    index = index + 1

# 왼쪽 오른쪽 눈 파악
if eye_1[0] < eye_2[0] :
    left_eye = eye_1
    right_eye = eye_2
else:
    right_eye = eye_1
    left_eye = eye_2

# central points of the rectangles
left_eye_center = (int(left_eye[0] + (left_eye[2]/2)),
                   int(left_eye[1] + (left_eye[3]/2)))

left_eye_center_x = left_eye_center[0]
left_eye_center_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_center_x = right_eye_center[0]
right_eye_center_y = right_eye_center[1]

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

if left_eye_center_y > right_eye_center_y :
    A = (right_eye_center_x, left_eye_center_y)
    direction = -1 # 정수 -1은 이미지가 시계방향으로 회전함을 나타냅니다.
else :
    A = (left_eye_center_x, right_eye_center_y)
    direction = 1 # 정수 1은 이미지가 시계 반대방향으로 회전함을 나타냅니다.

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

# 각도 구하기
# np.arctan 함수 단위는 라디안 단위
# 라디안 단위 -> 각도 : (theta * 180) / np.pi
delta_x = right_eye_center_x - left_eye_center_x
delta_y = right_eye_center_y - left_eye_center_y
angle = np.arctan(delta_y / delta_x)
angle = (angle*180) /np.pi
print(angle)
# -21.80140948635181 도

h, w = img.shape[:2]
h, w, _ = img.shape

center = (w//2, h//2)
M = cv2.getRotationMatrix2D(center, angle, 1.0)

rotated = cv2.warpAffine(img, M, (w,h))

cv2.imshow('face_drawed', img)
cv2.waitKey(0)

cv2.imshow('face_rotated', rotated)
cv2.waitKey(0)

dist_1 = np.sqrt((delta_x * delta_x) + (delta_y * delta_y))
dist_2 = np.sqrt((delta_x * delta_x) + (delta_y * delta_y))

ratio = dist_1/dist_2
```



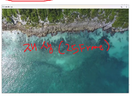




1208 > 02동영상파일읽기.py > ...

```python
import cv2

###### 동영상 속성 확인 ######
cap = cv2.VideoCapture('./video01.mp4') # 0을 넣으면 웹 캠을 가져오게 됩니다. ( 연결 되어있는 카메라 디바이스 가져오는
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('fps :', fps)
print('frame_count: ', frame_count)

###### 동영상 파일 읽기 ######
if cap.isOpened(): # 캡쳐 객체 초기화 확인
    while True:
        ret, frame = cap.read()
        if ret:
            cv2.imshow("video file show", frame)
            cv2.waitKey(25)
        else : # 읽기 실패시 브레이크
            break
else:
    print('비디오 파일 읽기 실패')

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

```
width : 1280.0 height : 720.0
fps : 25.0
frame_count:  323.0
```



1208 > 03동영상프레임나눠저장.py > ...

```python
import cv2
import os

video_file_path = './video01.mp4'

# 동영상 캡쳐 객체 생성
cap = cv2.VideoCapture(video_file_path)

fps = cap.get(cv2.CAP_PROP_FPS)
print('fps >> ', fps)
count = 0
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('./frame_image_save', exist_ok=True) # exist_ok 가 트루면 폴더가 있을경우 예외처리 해줌
                cv2.imwrite('./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()
```

출력

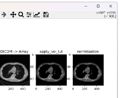
```
fps >>  25.0
save frame number >> 25
save frame number >> 50
save frame number >> 75
save frame number >> 100
save frame number >> 125
save frame number >> 150
save frame number >> 175
save frame number >> 200
save frame number >> 225
save frame number >> 250
save frame number >> 275
save frame number >> 300
```

파일

frame_image_save
- frame0.jpg
- frame1.jpg
- frame2.jpg
- frame3.jpg
- frame4.jpg
- frame5.jpg
- frame6.jpg
- frame7.jpg
- frame8.jpg
- frame9.jpg
- frame10.jpg
- frame11.jpg

1208 > 04DICOM파일읽기.py > ...

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

window_center = -600
window_width = 1600

# DICOM 파일 읽어는 함수 -> pydicom.read_file() , pydicom.dcmread()
slice = pydicom.read_file("./ID_0000_AGE_0060_CONTRAST_1_CT.dcm")
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')

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

plt.subplot(1, 3, 3)
plt.title("normlization")
plt.imshow(image3, cmap='gray')
plt.show()
```

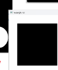


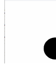
1208 > 05고급-이미지블렌딩.py > ...

```python
# 같은 크기의 이미지 블렌딩 실험

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

large_img = cv2.imread('./ex_image.png')
watermark = cv2.imread('./ex_image_logo.png')

print('large_image size >>', large_img.shape)
print('watermarker_image size >>', watermark.shape)

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

# 혼합 진행

# 처음 0.5 값 설정
# blended = cv2.addWeighted(img1, 0.5, img2, 0.5, 0)

# 높은 값으로 전달
# blended = cv2.addWeighted(img1, 0.8, img2, 0.2, 0)

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

cv2.imshow("img show", blended)
cv2.waitKey(0)
```

1208 > 06고급-비트연산관련.py > ...

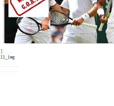
```python
import cv2
import numpy as np

# ex-01
img_rectangle = np.ones((400,400), dtype='uint8')
cv2.rectangle(img_rectangle, (50,50),(300,300),(255,255,255),-1
cv2.imshow("image show",img_rectangle)
cv2.waitKey(0)

# ex-02
img_circle = np.ones((400,400), dtype='uint8')
cv2.circle(img_circle, (300,300),70,(255,255,255),-1)
cv2.imshow("image show",img_circle)
cv2.waitKey(0)

# ex -03
bitwiseAnd = cv2.bitwise_and(img_rectangle, img_circle)
cv2.imshow("image bitwiseAnd", bitwiseAnd)
cv2.waitKey(0)

bitwiseOr = cv2.bitwise_or(img_rectangle, img_circle)
cv2.imshow("image bitwiseOr", bitwiseOr)
cv2.waitKey(0)

bitwiseXor = cv2.bitwise_xor(img_rectangle, img_circle)
cv2.imshow("image bitwiseXor", bitwiseXor)
cv2.waitKey(0)

rec_not = cv2.bitwise_not(img_rectangle)
cv2.imshow('rectangle not ', rec_not)
cv2.waitKey(0)

cir_not = cv2.bitwise_not(img_circle)
cv2.imshow('circle not ', cir_not)
cv2.waitKey(0)
```





1208 > 08고급-블렌딩 없이 오버레이-붙여넣기.py > ...

```python
import cv2

large_img = cv2.imread('./ex_image.png')
watermark = cv2.imread('./ex_image_logo.png')

small_img = cv2.resize(watermark, (300,300))
# cv2.imshow('test', small_img)
# cv2.waitKey(0)

x_offset = 30
y_offset = 170

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

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



1208 > 09고급-이미지에 배경없이 글자만 띄우기.py > ...

```python
import cv2

large_img = cv2.imread('ex_image.png')
watermark = cv2.imread('ex_image_logo.png')

x_offset = 400
y_offset = 170

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

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)

bg = cv2.bitwise_or(roi, roi, mask = mask)
mask_inv = cv2.bitwise_not(mask)

fg = cv2.bitwise_and(small_img, small_img, mask=mask_inv)

final_roi = cv2.add(bg, fg)

large_img[y_offset:y_offset + small_img.shape[0],
          x_offset:x_offset + small_img.shape[1]] = final_roi

cv2.imshow('test', large_img)
cv2.waitKey(0)
```