```python
import cv2
import numpy as np

# # ex-04 마스킹 과제는 흰색대신 이미지를 넣어주시면 됩니다. (원하는 이미지 혹은 얼굴이미지)
# 마스킹에 흰색 대신 이미지 넣기
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("...", mask)
# cv2.waitKey(0)

face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

# Lodaing the image - 얼굴 이미지 데이터 읽기
img = cv2.imread('./muhan.jpg')
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)
detected_faces = []
# 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
    detected_faces.append(img[y:(y+h), x:(x+w)])

cv2.imshow("check face detecting", img)
cv2.waitKey(0)

resize_size = [((60,50), (280,280)), ((420,50), (550,230)), ((750,50), (920,280))]

for i in range(len(detected_faces)):
    x_width = abs(resize_size[i][0][0]-resize_size[i][1][0])
    y_width = abs(resize_size[i][0][1]-resize_size[i][1][1])

    detected_faces[i] = cv2.resize(detected_faces[i], (x_width, y_width))

    # cv2.imshow("check face detecting", detected_faces[i])
    # cv2.waitKey(0)

    # cv2.imshow("check face detecting", mask[resize_size[i][0][1]:resize_size[i][0][1]+y_width, resize_size[i][0][0]:resize_size
    # cv2.waitKey(0)

    mask[resize_size[i][0][1]:resize_size[i][0][1]+y_width, resize_size[i][0][0]:resize_size[i][0][0]+x_width] = detected_faces[i]

cv2.imshow("check face detecting", mask)
cv2.waitKey(0)
```

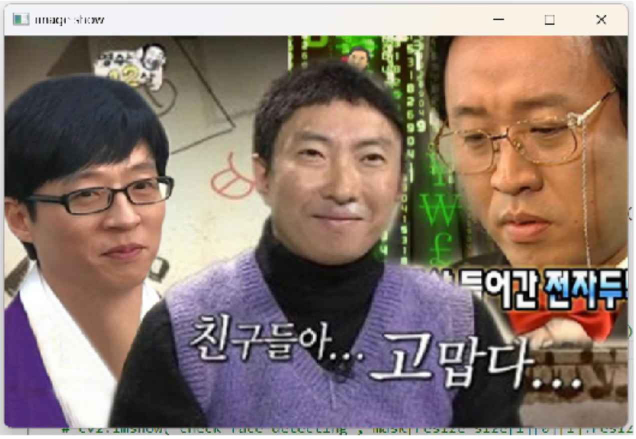
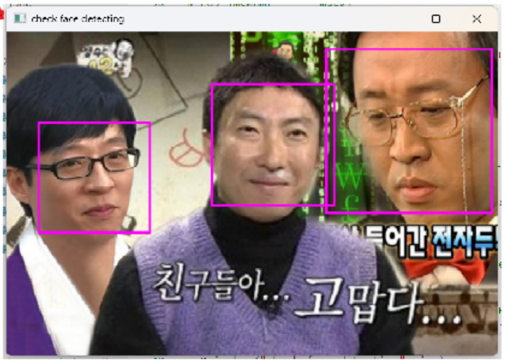


image show
드어간 전자두
친구들아... 고맙다...


check face detecting
드어간 전자두
친구들아... 고맙다...