Computer vision – HW10

Name: WeiWen Wu

Student ID: 10901007A

Question 1

A、 Source code

```
from numpy import ndarray
import cv2
from typing import Optional
FONT:int = cv2.FONT_HERSHEY_SIMPLEX

def bgr2gray(img:ndarray) -> ndarray:
    """Convert bgr to gray and remove noise points."""
    return cv2.equalizeHist(cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)) # BGR to gray.

def plot_from_cascade(img:ndarray,object_name:str,object:ndarray) ->
Optional[tuple[ndarray,ndarray]]:
    """Read cascade data and draw pictures."""
    if len(object)==0: return None
    for (x, y, w, h) in object:
        cv2.rectangle(img, (x, y), (x+w, y+h), (255, 0, 0), 2)
        roi_gray = gray[y:y+h, x:x+w]
        roi_color = img[y:y+h, x:x+w]
        cv2.putText(img, object_name , (x, y), FONT, .5, (0, 255, 255), 2)
    return roi_gray,roi_color

folder = "./HW10/Haar cascade classifier xml files"
object_cascade = cv2.CascadeClassifier(f'{folder}/haarcascade_smile.xml')
cap = cv2.VideoCapture(0)
cv2.namedWindow("cascade")

### Create a track bar ### cv2.createTrackbar('Slider name', 'window name', min, max,
fn)
cv2.createTrackbar("scaleFactor", "cascade", 150, 500, lambda _:_)
cv2.createTrackbar("minNeighbors", "cascade", 0, 20, lambda _:_)
cv2.setTrackbarPos("scaleFactor", "cascade", 400)
cv2.setTrackbarPos("minNeighbors", "cascade", 18)

while True:
    ret, img = cap.read()
    gray = bgr2gray(img)

    ### Get trackbar position. ### cv2.setTrackbarPos('Slider name', 'window name',
default)
```

```
    scaleFactor= cv2.getTrackbarPos('scaleFactor','cascade')
    minNeighbors= cv2.getTrackbarPos('minNeighbors','cascade')

    smile = object_cascade.detectMultiScale(gray, scaleFactor/100, minNeighbors)
    plot_from_cascade(img,"smile",smile)
    cv2.imshow('cascade', img)
    k = cv2.waitKey(30) & 0xff
    if k == 27:
        break

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

B、 Analyze

原來的圖片，如 Figure 1 所示，使用 `Trackbar` 來調整 `detectMultiScale` 中 scaleFactor 與 minNeighbors 之值。

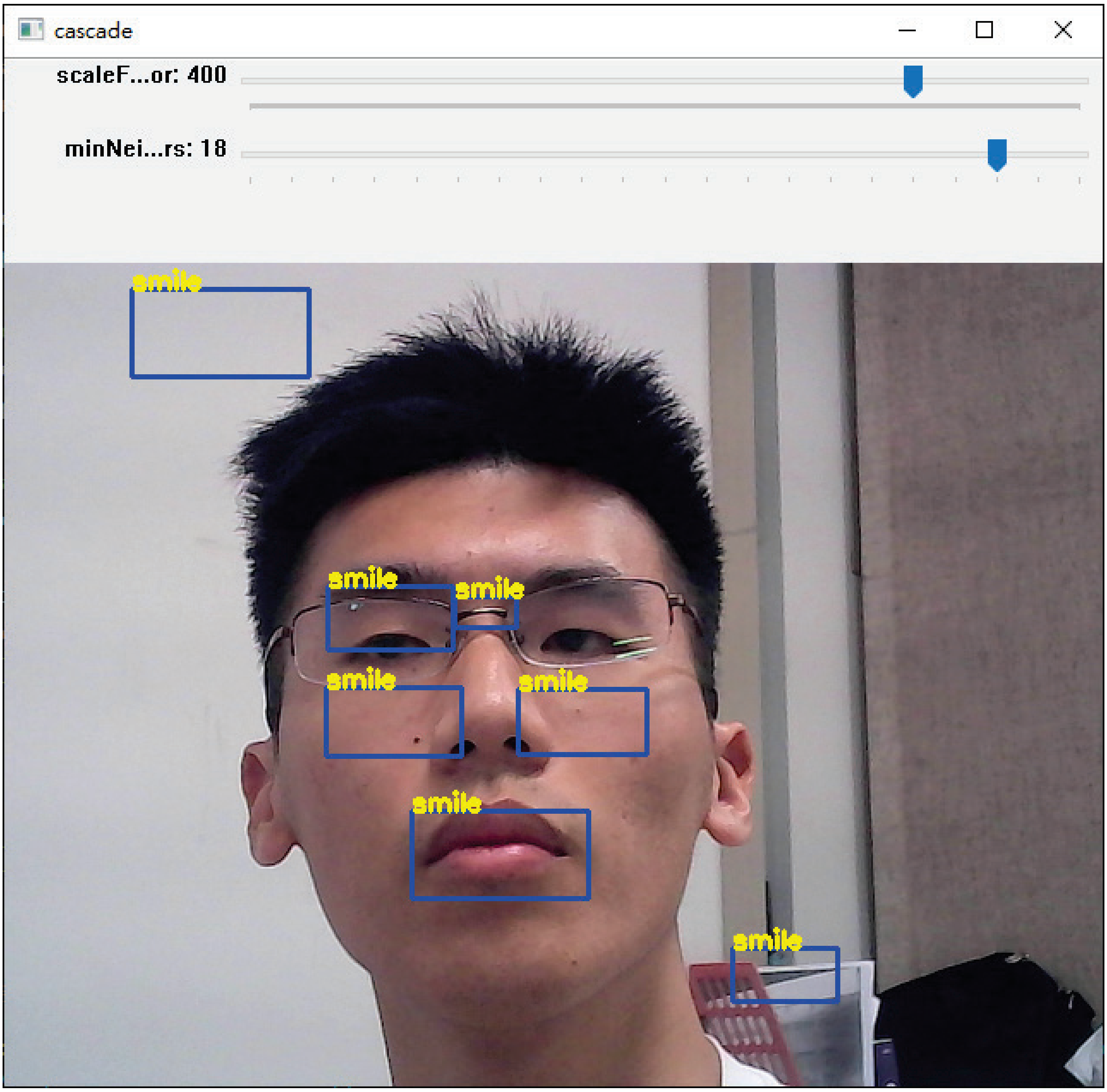
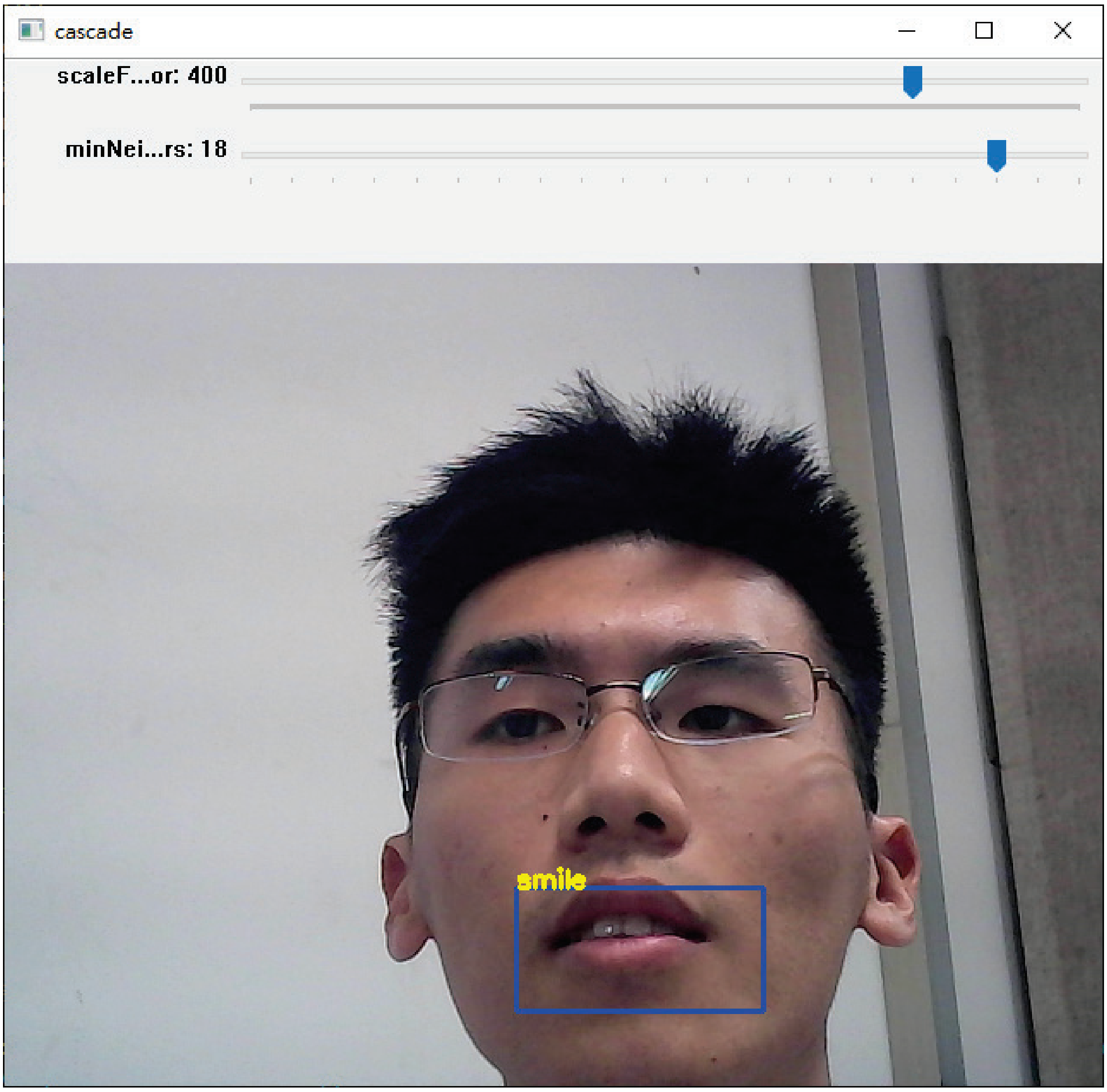


Figure 1 Original picture

C、 Result map



scaleF...or: 400

minNei...rs: 18

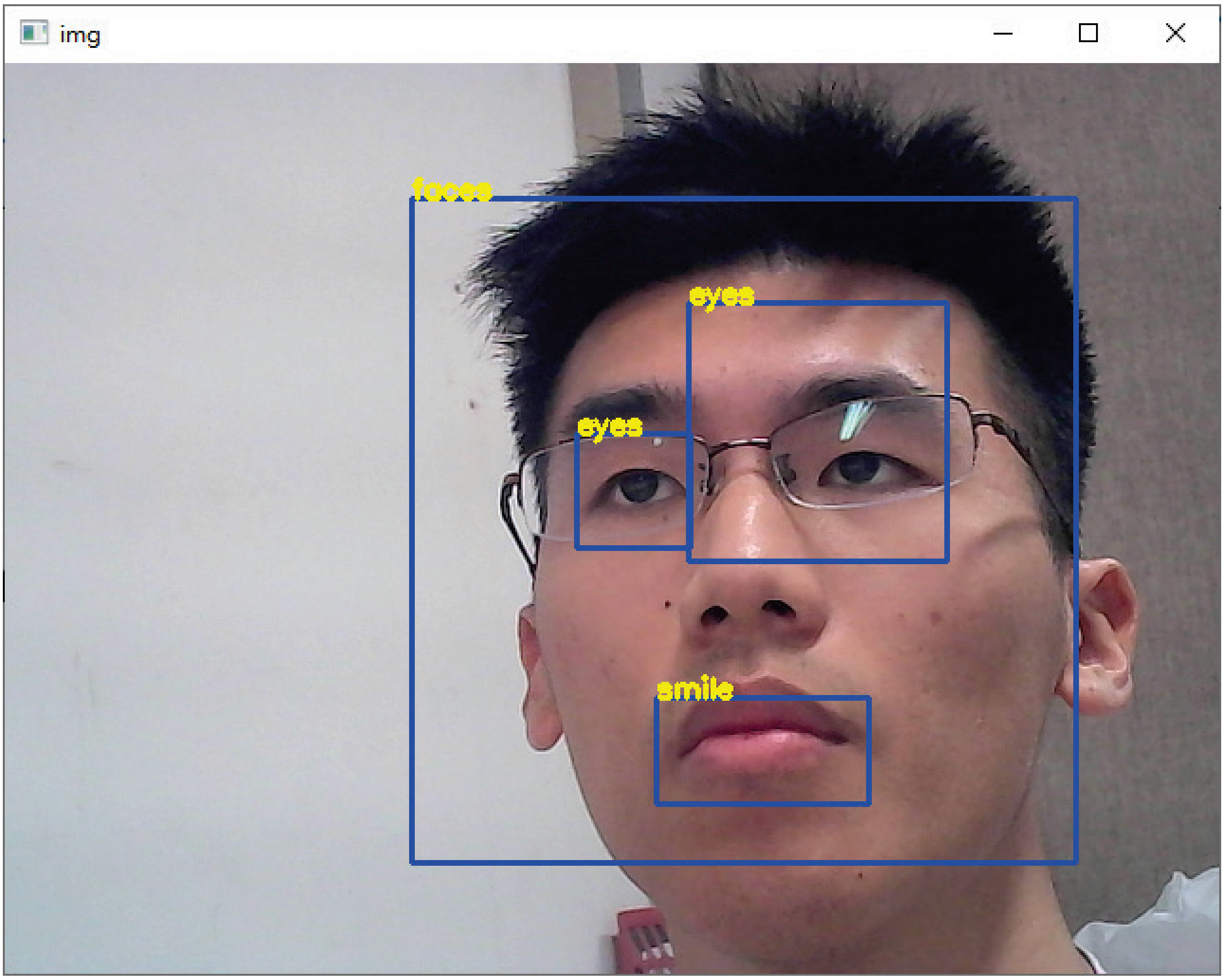
Question 2

A、 Source code

```
from numpy import ndarray
import cv2
from typing import Optional
FONT = cv2.FONT_HERSHEY_SIMPLEX
def bgr2gray(img:ndarray) -> ndarray:
    """Convert bgr to gray and equalizeHist."""
    return cv2.equalizeHist(cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)) # BGR to gray.

def plot_from_cascade(img:ndarray,object_name:str,object:ndarray) ->
Optional[tuple[ndarray,ndarray]]:
    """Read cascade data and draw pictures."""
    if len(object)==0: return None
    for (x, y, w, h) in object:
        cv2.rectangle(img, (x, y), (x+w, y+h), (255, 0, 0), 2)
        roi_gray = gray[y:y+h, x:x+w]
        roi_color = img[y:y+h, x:x+w]
        cv2.putText(img, object_name , (x, y), FONT, .5, (0, 255, 255), 2)
    return roi_gray,roi_color

cap = cv2.VideoCapture(0)

### Import identification tools. ###
folder             = "./HW10/Haar cascade classifier xml files"
face_cascade        =
cv2.CascadeClassifier(f'{folder}/haarcascade_frontalface_default.xml')
eye_cascade         = cv2.CascadeClassifier(f'{folder}/haarcascade_eye.xml')
smile_cascade       = cv2.CascadeClassifier(f'{folder}/haarcascade_smile.xml')
profileface_cascade = cv2.CascadeClassifier(f'{folder}/haarcascade_profileface.xml')

cv2.namedWindow("cascade")

### Create a track bar ###
## cv2.createTrackbar('Slider name', 'window name', min, max, fn)
cv2.createTrackbar("scaleFactor", "cascade", 150, 500, lambda _:_)
cv2.createTrackbar("minNeighbors", "cascade", 0, 20, lambda _:_)
cv2.setTrackbarPos("scaleFactor", "cascade", 107)
cv2.setTrackbarPos("minNeighbors", "cascade", 1)
```

```
while 1:
    ret, img = cap.read()
    gray = bgr2gray(img)

    # detectMultiScale(roi_gray, scaleFactor=1.02, minNeighbors=3,  minSize=(40,40))
    faces = face_cascade.detectMultiScale(gray, 1.3, 5)
    profileface = profileface_cascade.detectMultiScale(gray, 1.3, 5)

    ### Read cascade data and draw pictures. ###
    roi = plot_from_cascade(img,"faces",faces)
    plot_from_cascade(img,"profileface",profileface)

    ### If no face is detected, eyes and mouth will not be detected. ###
    if roi != None:
        roi_gray,roi_color = roi

        ### Get trackbar position. ###
        scaleFactor= cv2.getTrackbarPos('scaleFactor','cascade')
        minNeighbors= cv2.getTrackbarPos('minNeighbors','cascade')

        # detectMultiScale(roi_gray, scaleFactor=1.02, minNeighbors=3,
minSize=(40,40))
        eyes = eye_cascade.detectMultiScale(roi_gray, 1.1, 1)
        smile = smile_cascade.detectMultiScale(roi_gray, 3.1, 19)

        ### Read cascade data and draw pictures. ###
        plot_from_cascade(roi_color,"eyes",eyes)
        plot_from_cascade(roi_color,"smile",smile)

    cv2.imshow('img', img)
    k = cv2.waitKey(30) & 0xff
    if k == 27:
        break

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

B、 Result map


img
faces
eyes
eyes
smile