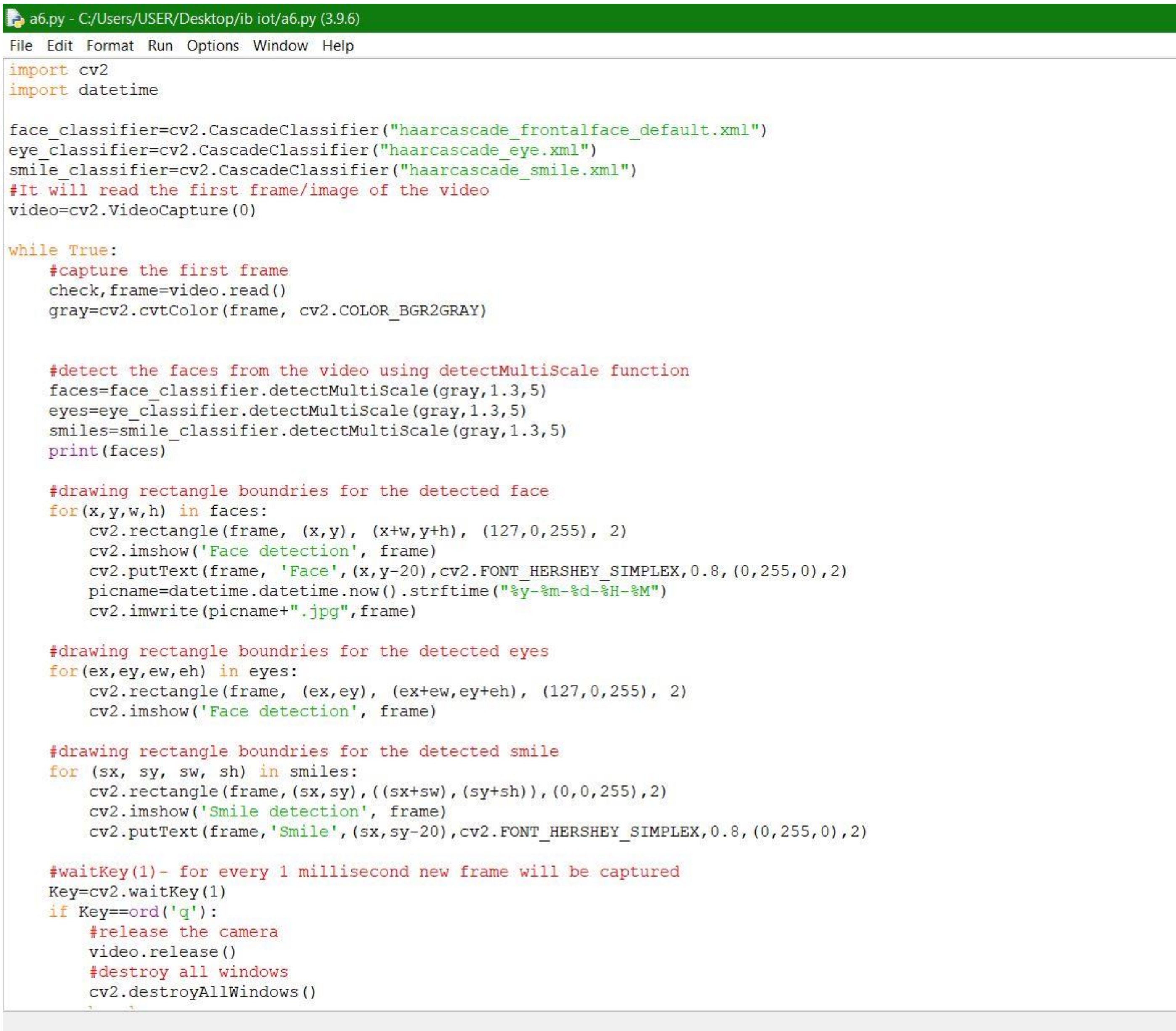
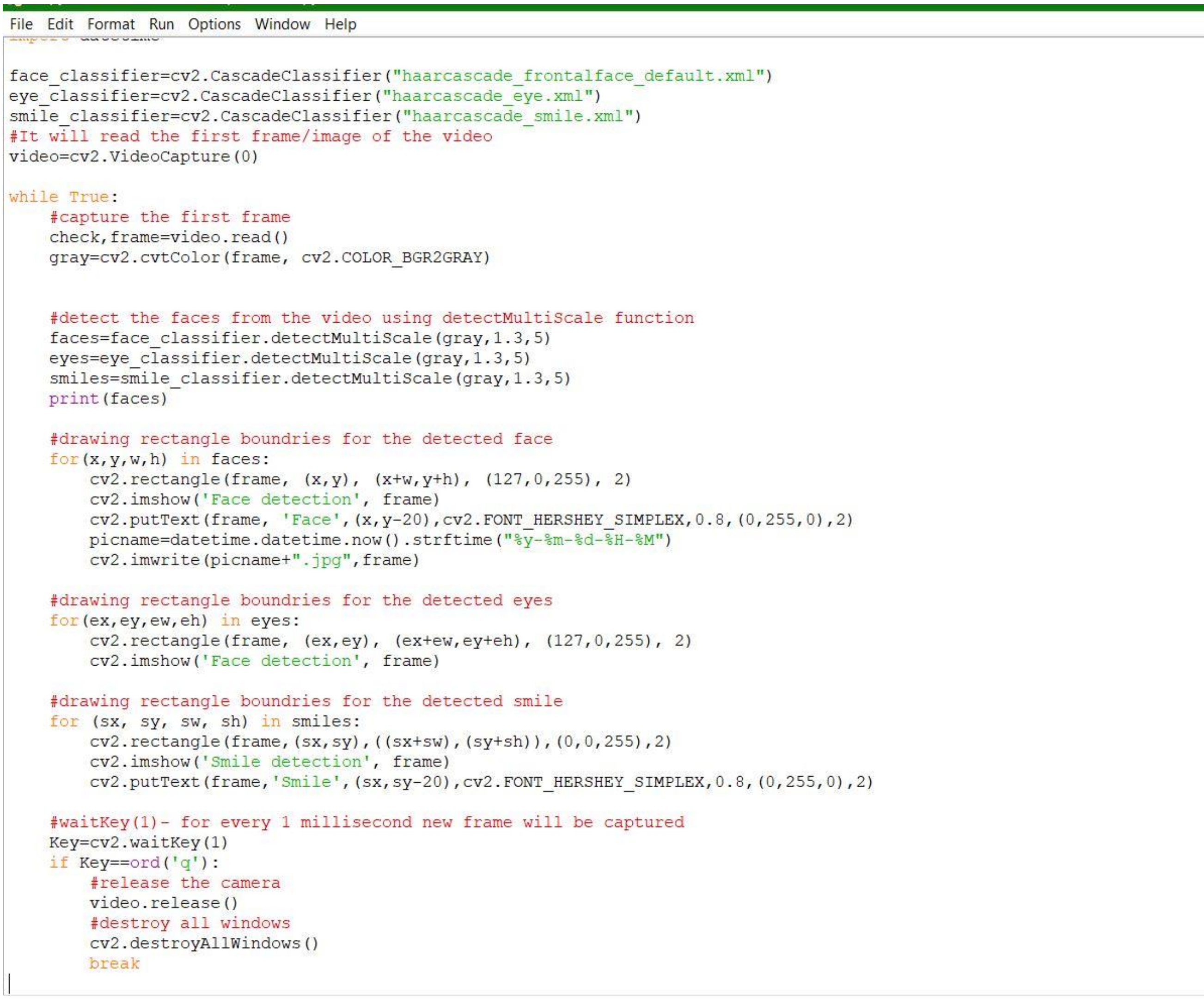
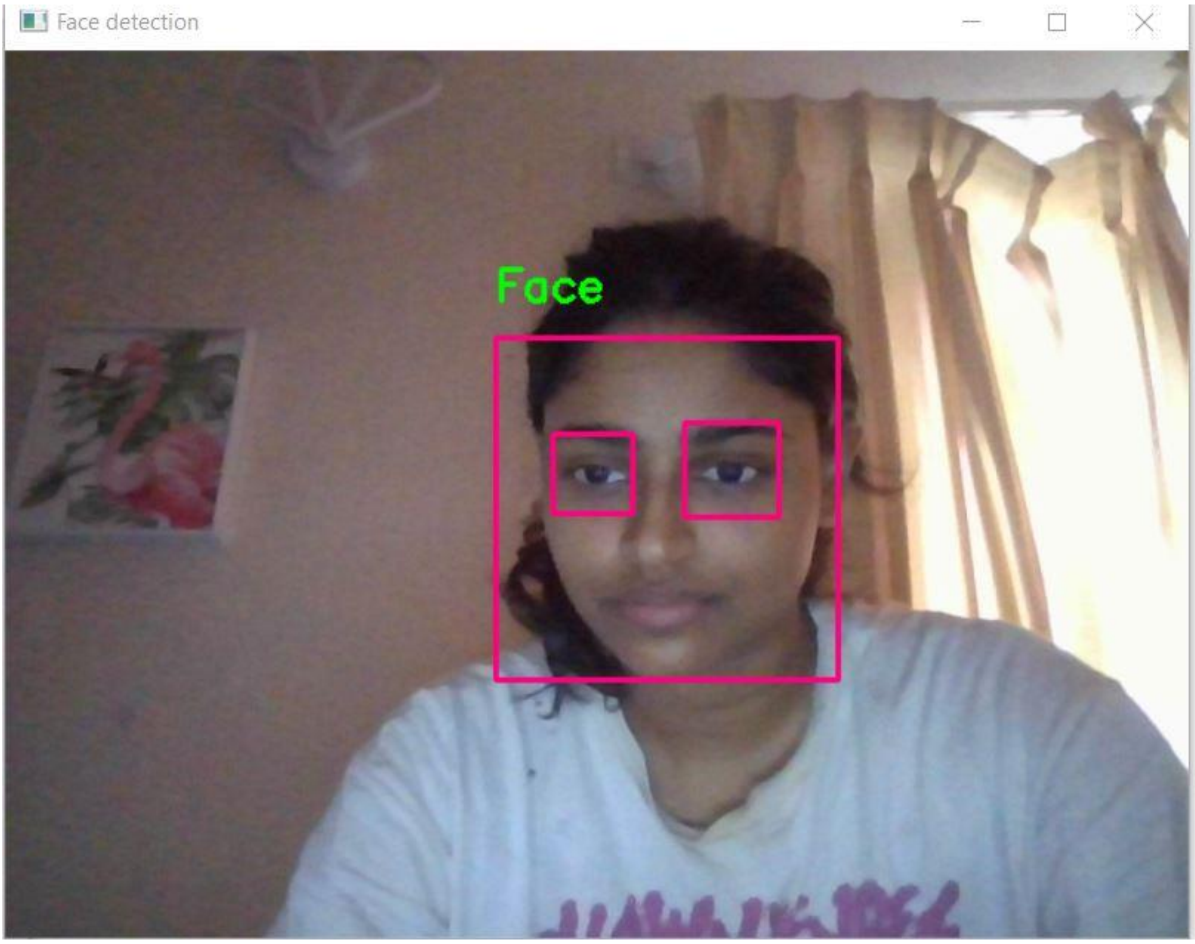

a6.py - C:/Users/USER/Desktop/ib iot/a6.py (3.9.6)
File Edit Format Run Options Window Help

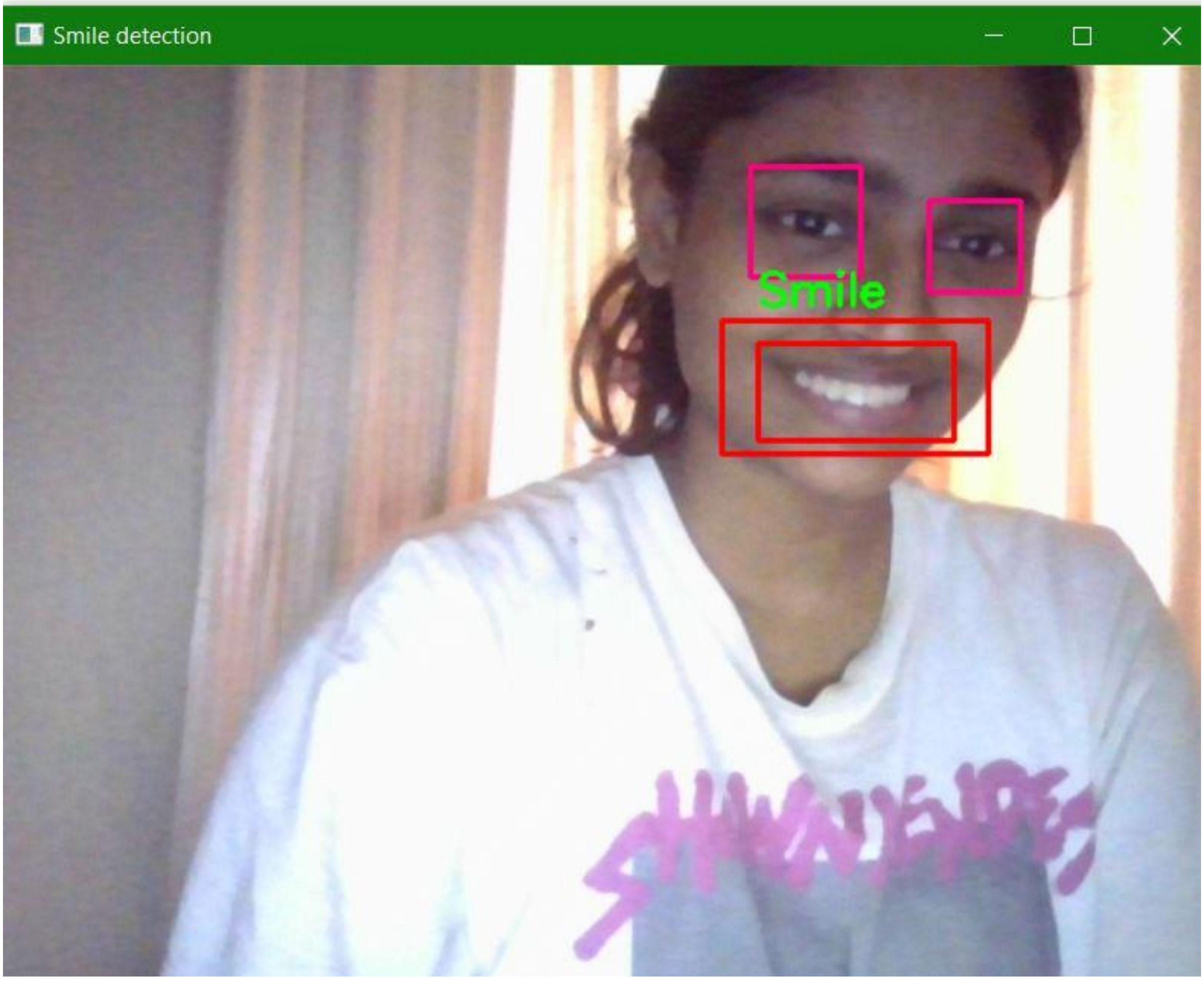
```
import cv2
import datetime

face_classifier=cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
eye_classifier=cv2.CascadeClassifier("haarcascade_eye.xml")
smile_classifier=cv2.CascadeClassifier("haarcascade_smile.xml")
#It will read the first frame/image of the video
video=cv2.VideoCapture(0)

while True:
    #capture the first frame
    check,frame=video.read()
    gray=cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)


    #detect the faces from the video using detectMultiScale function
    faces=face_classifier.detectMultiScale(gray,1.3,5)
    eyes=eye_classifier.detectMultiScale(gray,1.3,5)
    smiles=smile_classifier.detectMultiScale(gray,1.3,5)
    print(faces)

    #drawing rectangle boundries for the detected face
    for(x,y,w,h) in faces:
        cv2.rectangle(frame, (x,y), (x+w,y+h), (127,0,255), 2)
        cv2.imshow('Face detection', frame)
        cv2.putText(frame, 'Face',(x,y-20),cv2.FONT_HERSHEY_SIMPLEX,0.8,(0,255,0),2)
        picname=datetime.datetime.now().strftime("%y-%m-%d-%H-%M")
        cv2.imwrite(picname+".jpg",frame)

    #drawing rectangle boundries for the detected eyes
    for(ex,ey,ew,eh) in eyes:
        cv2.rectangle(frame, (ex,ey), (ex+ew,ey+eh), (127,0,255), 2)
        cv2.imshow('Face detection', frame)

    #drawing rectangle boundries for the detected smile
    for (sx, sy, sw, sh) in smiles:
        cv2.rectangle(frame,(sx,sy),((sx+sw),(sy+sh)),(0,0,255),2)
        cv2.imshow('Smile detection', frame)
        cv2.putText(frame,'Smile',(sx,sy-20),cv2.FONT_HERSHEY_SIMPLEX,0.8,(0,255,0),2)

    #waitKey(1)- for every 1 millisecond new frame will be captured
    Key=cv2.waitKey(1)
    if Key==ord('q'):
        #release the camera
        video.release()
        #destroy all windows
        cv2.destroyAllWindows()
```



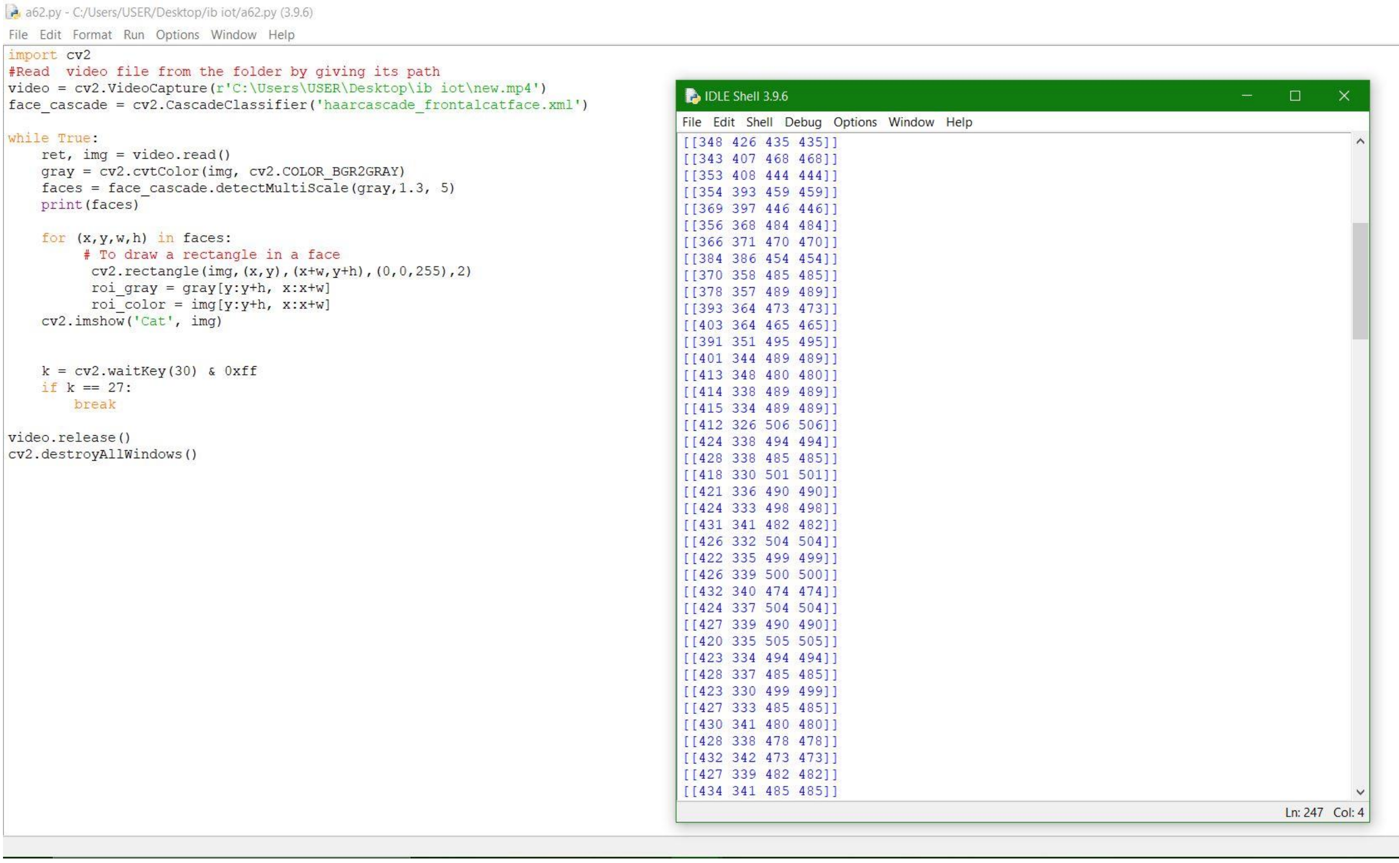
```
face_classifier=cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
eye_classifier=cv2.CascadeClassifier("haarcascade_eye.xml")
smile_classifier=cv2.CascadeClassifier("haarcascade_smile.xml")
#It will read the first frame/image of the video
video=cv2.VideoCapture(0)

while True:
    #capture the first frame
    check,frame=video.read()
    gray=cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)


    #detect the faces from the video using detectMultiScale function
    faces=face_classifier.detectMultiScale(gray,1.3,5)
    eyes=eye_classifier.detectMultiScale(gray,1.3,5)
    smiles=smile_classifier.detectMultiScale(gray,1.3,5)
    print(faces)

    #drawing rectangle boundries for the detected face
    for(x,y,w,h) in faces:
        cv2.rectangle(frame, (x,y), (x+w,y+h), (127,0,255), 2)
        cv2.imshow('Face detection', frame)
        cv2.putText(frame, 'Face',(x,y-20),cv2.FONT_HERSHEY_SIMPLEX,0.8,(0,255,0),2)
        picname=datetime.datetime.now().strftime("%y-%m-%d-%H-%M")
        cv2.imwrite(picname+".jpg",frame)

    #drawing rectangle boundries for the detected eyes
    for(ex,ey,ew,eh) in eyes:
        cv2.rectangle(frame, (ex,ey), (ex+ew,ey+eh), (127,0,255), 2)
        cv2.imshow('Face detection', frame)

    #drawing rectangle boundries for the detected smile
    for (sx, sy, sw, sh) in smiles:
        cv2.rectangle(frame,(sx,sy),((sx+sw),(sy+sh)),(0,0,255),2)
        cv2.imshow('Smile detection', frame)
        cv2.putText(frame,'Smile',(sx,sy-20),cv2.FONT_HERSHEY_SIMPLEX,0.8,(0,255,0),2)

    #waitKey(1)- for every 1 millisecond new frame will be captured
    Key=cv2.waitKey(1)
    if Key==ord('q'):
        #release the camera
        video.release()
        #destroy all windows
        cv2.destroyAllWindows()
        break
```



Face detection


Smile detection
Smile

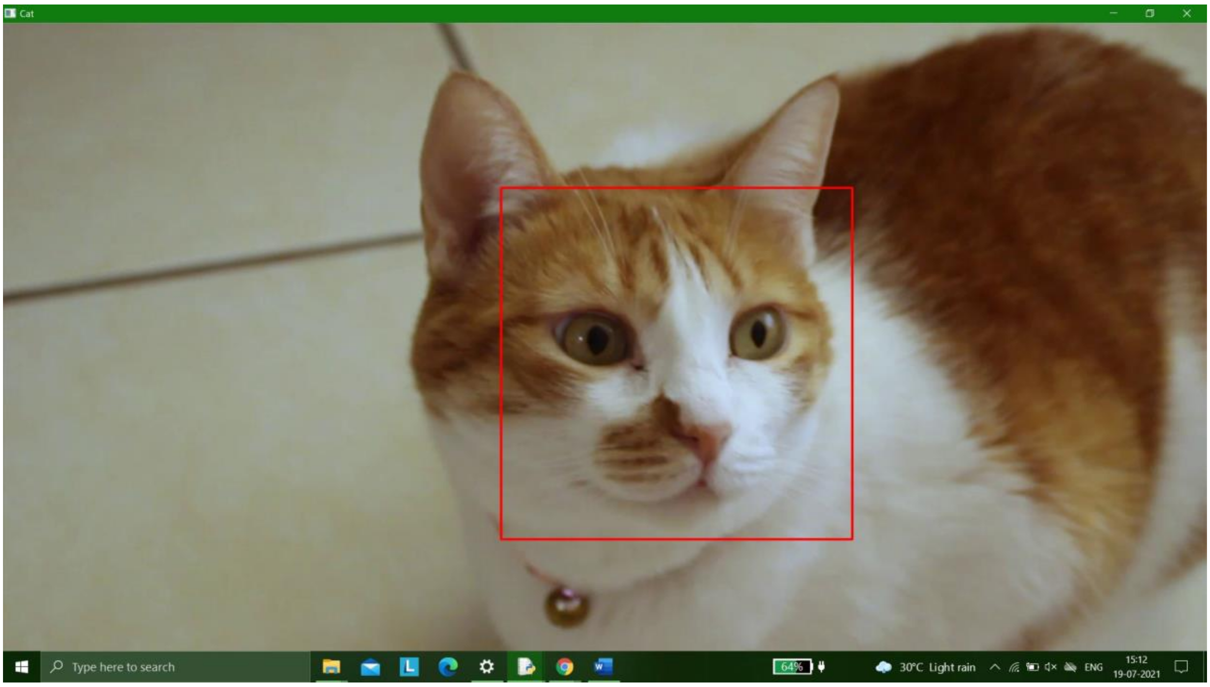
CAT



a62.py - C:/Users/USER/Desktop/ib iot/a62.py (3.9.6)
File Edit Format Run Options Window Help

```
import cv2
#Read  video file from the folder by giving its path
video = cv2.VideoCapture(r'C:\Users\USER\Desktop\ib iot\new.mp4')
face_cascade = cv2.CascadeClassifier('haarcascade_frontalcatface.xml')

while True:
    ret, img = video.read()
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray,1.3, 5)
    print(faces)

    for (x,y,w,h) in faces:
         # To draw a rectangle in a face
          cv2.rectangle(img,(x,y),(x+w,y+h),(0,0,255),2)
          roi_gray = gray[y:y+h, x:x+w]
          roi_color = img[y:y+h, x:x+w]
    cv2.imshow('Cat', img)


    k = cv2.waitKey(30) & 0xff
    if k == 27:
        break

video.release()
cv2.destroyAllWindows()
```

```
[[348 426 435 435]]
[[343 407 468 468]]
[[353 408 444 444]]
[[354 393 459 459]]
[[369 397 446 446]]
[[356 368 484 484]]
[[366 371 470 470]]
[[384 386 454 454]]
[[370 358 485 485]]
[[378 357 489 489]]
[[393 364 473 473]]
[[403 364 465 465]]
[[391 351 495 495]]
[[401 344 489 489]]
[[413 348 480 480]]
[[414 338 489 489]]
[[415 334 489 489]]
[[412 326 506 506]]
[[424 338 494 494]]
[[428 338 485 485]]
[[418 330 501 501]]
[[421 336 490 490]]
[[424 333 498 498]]
[[431 341 482 482]]
[[426 332 504 504]]
[[422 335 499 499]]
[[426 339 500 500]]
[[432 340 474 474]]
[[424 337 504 504]]
[[427 339 490 490]]
[[420 335 505 505]]
[[423 334 494 494]]
[[428 337 485 485]]
[[423 330 499 499]]
[[427 333 485 485]]
[[430 341 480 480]]
[[428 338 478 478]]
[[432 342 473 473]]
[[427 339 482 482]]
[[434 341 485 485]]
```


Cat
Type here to search
64%
30°C Light rain
ENG
15:12
19-07-2021