## VIT-IOT-INDUSTRY CERTIFICATE-EXTERNSHIP PROGRAM ASSIGNMENT-6

Assignment 6.py - D:/Assignment 6/Assignment 6.py (3.9.6)

video.release()
#destroy all windows
cv2.destroyAllWindows()

break

## Assignment-6

Develop a python code to detect any object using Haar cascade classifier.

## **PYTHON CODE:**

```
File Edit Format Run Options Window Help
import cv2
face classifier=cv2.CascadeClassifier("haarcascade frontalface default.xml")
eye classifier=cv2.CascadeClassifier("haarcascade eye.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)
    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)
        #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)
    #waitKey(1) - for every 1 millisecond new frame will be captured
    Key=cv2.waitKey(1)
    if Key==ord('q'):
        #release the camera
```

```
face_classifier=cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
eye_classifier=cv2.CascadeClassifier("haarcascade_eye.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)
  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)
    #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)
```

```
#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
```

## **PYTHON SHELL:**

```
*IDLE Shell 3.9.6*
  File Edit Shell Debug Options Window Help
   [[256 157 213 213]]
   [[251 152 229 229]]
[[250 152 229 229]]
   [[251 148 226 226]]
[[251 151 229 229]]
[[248 148 227 227]]
  [[248 148 227 227]]

[[249 150 231 231]]

[[242 149 234 234]]

[[242 147 234 234]]

[[242 147 234 234]]

[[239 150 229 229]]

[[236 146 233 233]]

[[232 150 226 226]]
  [[232 150 226 226]]

[[221 150 228 228]]

[[226 154 221 221]]

[[224 155 224 224]]

[[223 152 230 230]]

[[222 147 230 230]]

[[222 149 234 234]]

[[229 149 229 229]]

[[231 149 23 233]]

[[238 158 227 227]]
   [[238 158 227 227]]
[[238 157 226 226]]
   [[236 157 226 226]]
[[237 154 226 226]]
[[231 148 234 234]]
[[237 152 231 231]]
[[236 153 233 233]]
[[237 157 227 227]]
   [[238 154 227 227]]
[[237 152 231 231]]
  [[237 152 231 231]]
[[237 152 231 231]]
[[239 153 228 228]]
[[235 150 236 236]]
[[237 156 224 224]]
[[232 155 231 231]]
[[236 154 230 230]]
[[235 153 232 232]]
[[237 159 235 235]]
[[238 159 235 235]]
   [[236 151 235 235]]
[[236 151 235 235]]
[[245 157 229 229]]
   [[241 150 239 239]]
[[240 147 240 240]]
[[241 149 239 239]]
   [[236 142 238 238]]
[[236 144 241 241]]
[[244 145 230 230]]
                                                                                              0
```

**PYTHON CODE:** 

```
ar.py - C:\Users\Soujanya\Desktop\car\car.py (3.9.6)
 File Edit Format Run Options Window Help
import cv2
 car_classifier=cv2.CascadeClassifier("haarcascade_car.xml")
 video=cv2.VideoCapture("car.avi")
 frame_width = int(video.get(3))
 frame_height = int(video.get(4))
 out = cv2.VideoWriter('Output.avi',cv2.VideoWriter_fourcc('M','J','P','G'), 10, (frame_width,frame_height))
 while True:
     #capture the first frame
     check, frame=video.read()
     gray=cv2.cvtColor(frame, cv2.COLOR BGR2GRAY)
     cv2.imshow('frame', gray)
     #detect the faces from the video using detectMultiScale function
     cars=car classifier.detectMultiScale(gray, 1.3, 5)
     print(cars)
     #drawing rectangle boundries for the detected face
     for(x,y,w,h) in cars:
         cv2.rectangle(frame, (x,y), (x+w,y+h), (255,0,0), 2)
         cv2.imshow('Face detection', frame)
         out.write(frame)
         #picname=datetime.datetime.now().strftime("%y-%m-%d-%H-%M")
         #cv2.imwrite(picname+".jpg",frame)
     #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
         out.release()
         cv2.destroyAllWindows()
import cv2
car_classifier=cv2.CascadeClassifier("haarcascade_car.xml")
video=cv2.VideoCapture("car.avi")
frame_width = int(video.get(3))
frame_height = int(video.get(4))
out = cv2.VideoWriter('Output.avi',cv2.VideoWriter_fourcc('M','J','P','G'), 10,
(frame_width,frame_height))
while True:
  #capture the first frame
  check,frame=video.read()
```

```
gray=cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
cv2.imshow('frame', gray)
#detect the faces from the video using detectMultiScale function
cars=car_classifier.detectMultiScale(gray,1.3,5)
print(cars)
#drawing rectangle boundries for the detected face
for(x,y,w,h) in cars:
  cv2.rectangle(frame, (x,y), (x+w,y+h), (255,0,0), 2)
  cv2.imshow('Face detection', frame)
  out.write(frame)
  #picname=datetime.datetime.now().strftime("%y-%m-%d-%H-%M")
  #cv2.imwrite(picname+".jpg",frame)
#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
  out.release()
  cv2.destroyAllWindows()
  break
```