

```
In [1]: pip install opencv-python
```

```
Requirement already satisfied: opencv-python in c:\users\geethanjali\anaconda3\lib\site-packages (4.7.0.72)
Requirement already satisfied: numpy>=1.19.3 in c:\users\geethanjali\anaconda3\lib\site-packages (from opencv-python) (1.23.5)
Note: you may need to restart the kernel to use updated packages.
```

```
In [2]: import cv2
import numpy as np
import time
import datetime
```

```
In [3]: video =cv2.VideoCapture(0)
while True:
    check, frame= video.read()
    gray=cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
    cv2.imshow('Video',gray)
    if cv2.waitKey(1)==ord('q'):
        break
video.release()
cv2.destroyAllWindows()
```



```
In [6]: face_cascade=cv2.CascadeClassifier('stats/Haarcascades/haarcascade_frontalface_cascade.xml')
eye_cascade=cv2.CascadeClassifier('stats/Haarcascades/haarcascade_eye.xml')
body_cascade=cv2.CascadeClassifier('stats/Haarcascades/haarcascade_body.xml')

detection = False
detection_stopped_time=None
timer_started=False
SECONDS_TO_RECORD_AFTER_DETECTION=5

frame_size=(int(video.get(3)),int(video.get(4)))
fourcc=cv2.VideoWriter_fourcc(*"mp4v")

def detect(gray,frame):
    faces= face_cascade.detectMultiScale(gray,1.3,5)
    for(x,y,w,h) in faces:
        cv2.rectangle(frame,(x,y),(x+w,y+h),(250,0,0),2)
        roi_gray =gray[y:y+h, x:x+w]
        roi_color=frame[y:y+h, x:x+w]
        eyes=eye_cascade.detectMultiScale(roi_gray,1.1,3)
        for(ex,ey,ew,eh) in eyes:
            cv2.rectangle(roi_color,(ex,ey),(ex+ew,ey+eh),(0,255,0),2)
    return frame
video =cv2.VideoCapture(0)
while True:
    check, frame= video.read()
    gray=cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
    faces= face_cascade.detectMultiScale(gray,1.3,5)
    bodies= body_cascade.detectMultiScale(gray,1.3,5)

    if len(faces)+ len(bodies)>0:
        if detection:
            timer_started=False
        else:
            detection=True
            current_time=datetime.datetime.now().strftime("%d-%m-%Y-%H-%M-%S")
            out=cv2.VideoWriter(f"{current_time}.mp4",fourcc,20,frame_size)
            print("Started recording!")
    elif detection:
        if timer_started:
            if time.time()- detection_stopped_time>=SECONDS_TO_RECORD_AFTER_DETECTION:
                detection=False
                timer_started=False
                out.release()
                print('Stop Recording!')
        else:
            timer_started= True
            detection_stopped_time=time.time()
    if detection:
        out.write(frame)

    canvas=detect(gray,frame)
    cv2.imshow('Video',canvas)
    if cv2.waitKey(1)==ord('q'):
        break
video.release()
out.release()
```

```
cv2.destroyAllWindows()
```

```
Started recording!  
Stop Recording!  
Started recording!
```

In []:

In []: