

Mark X | Home Page X | IMG\_2020 X | ML\_final - X | final\_oper X | read\_a\_vide X | (65) How X | +

books/read\_a\_video\_stream\_from\_camera.ipynb#Read-a-video-Stream-form-Camer-(Frame-by-Frame)

YouTube Maps

read\_a\_video\_stream\_from\_camera Last Checkpoint: 23 minutes ago (unsaved changes)

View Insert Cell Kernel Widgets Help

Logout

Run C Code

Face Detection using Haar

```
import cv2
cap=cv2.VideoCapture(0)
face_cascade=cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
while True:
    ret,frame=cap.read()
    gray_frame=cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
    if ret==False:
        continue

    faces=face_cascade.detectMultiScale(gray_frame)

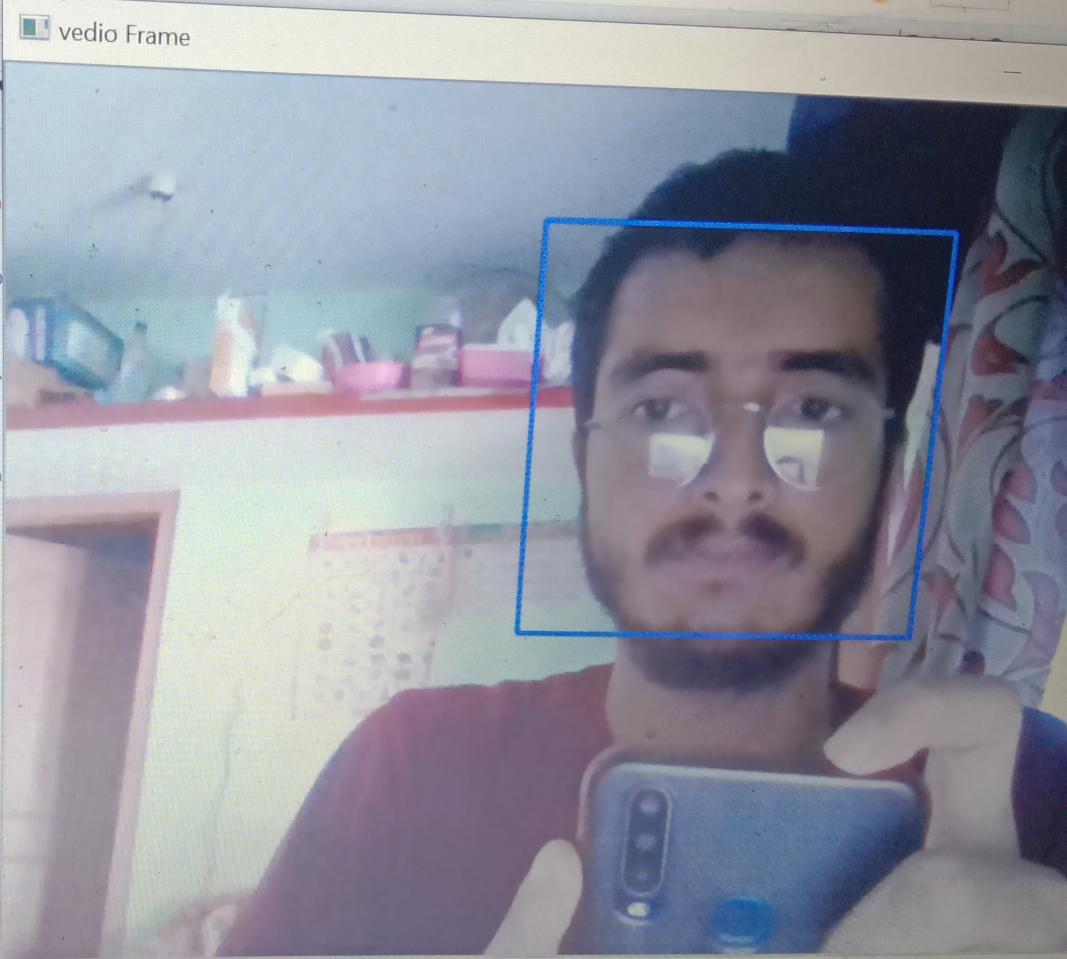
    for(x,y,w,h) in faces:
        cv2.rectangle(frame,(x,y),(x+w,y+h),255,2)

    cv2.imshow("vedio Frame",frame)

    key_pressed=cv2.waitKey(1) & 0xFF
    if key_pressed==ord('q'):
        break

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

anfis (1) (1).zip



Markdo X | Home Page X | Preprocessin X | ML\_final - Ju X | final\_opencv X | (68) How to X | +

s/Preprocessing\_and\_geting\_info

YouTube Maps

eprocessing\_and\_geting\_info\_fro

v Insert Cell Kernel Widgets

Run C Markdown

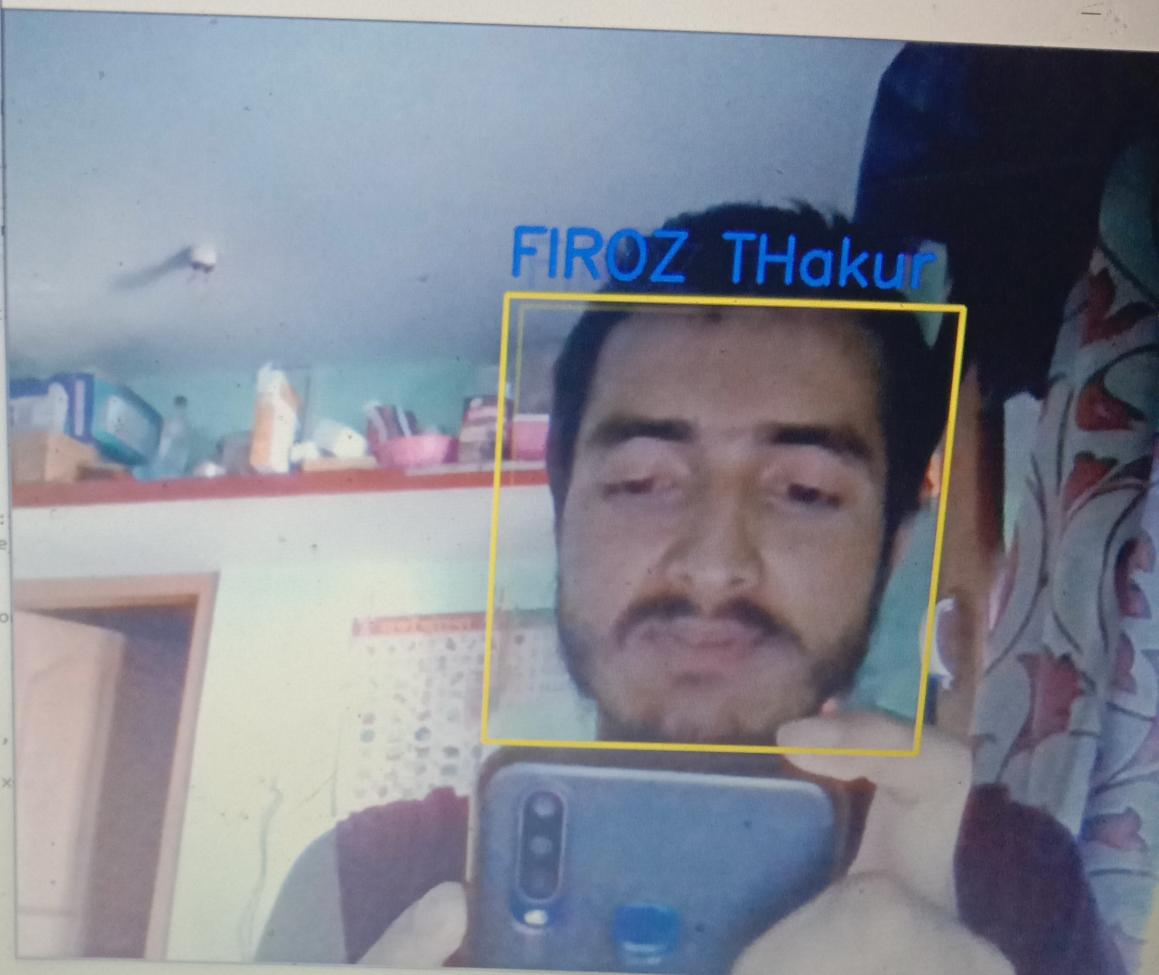
continue

```
faces=face_cascade.detectMultiScal
```

```
for face in faces:  
    x,y,w,h=face  
  
    #get the face ROI  
    offset=10  
    face_section=frame[y-offset:  
    face_section=cv2.resize(face  
  
    #predicted Label(out)  
    out=knn(trainset,face_sectio  
  
#display on the screen the name  
  
    pred_name=names[int(out)]  
    cv2.putText(frame,pred_name,  
    cv2.rectangle(frame,(x,y),(x  
cv2.imshow("Face",frame)  
  
key=cv2.waitKey(1) & 0xff  
if key==ord('q'):  
    break  
  
cap.release()  
cv2.destroyAllWindows()
```

```
Loaded abhi chotu.npy  
Loaded Bohemia sir.npy  
Loaded Chaman Lal.npy  
Loaded chinalo.npy  
Loaded diksha rana.npy  
Loaded FIROZ THakur.npy  
Loaded pinn...
```

Face



^ ENG 04-1