FACE RECOGNITION USING VIOLA & JONES ALGORITHM AND CSRT TRACKING (OPENCV) IN HOSPITALS

Project from AI Application course

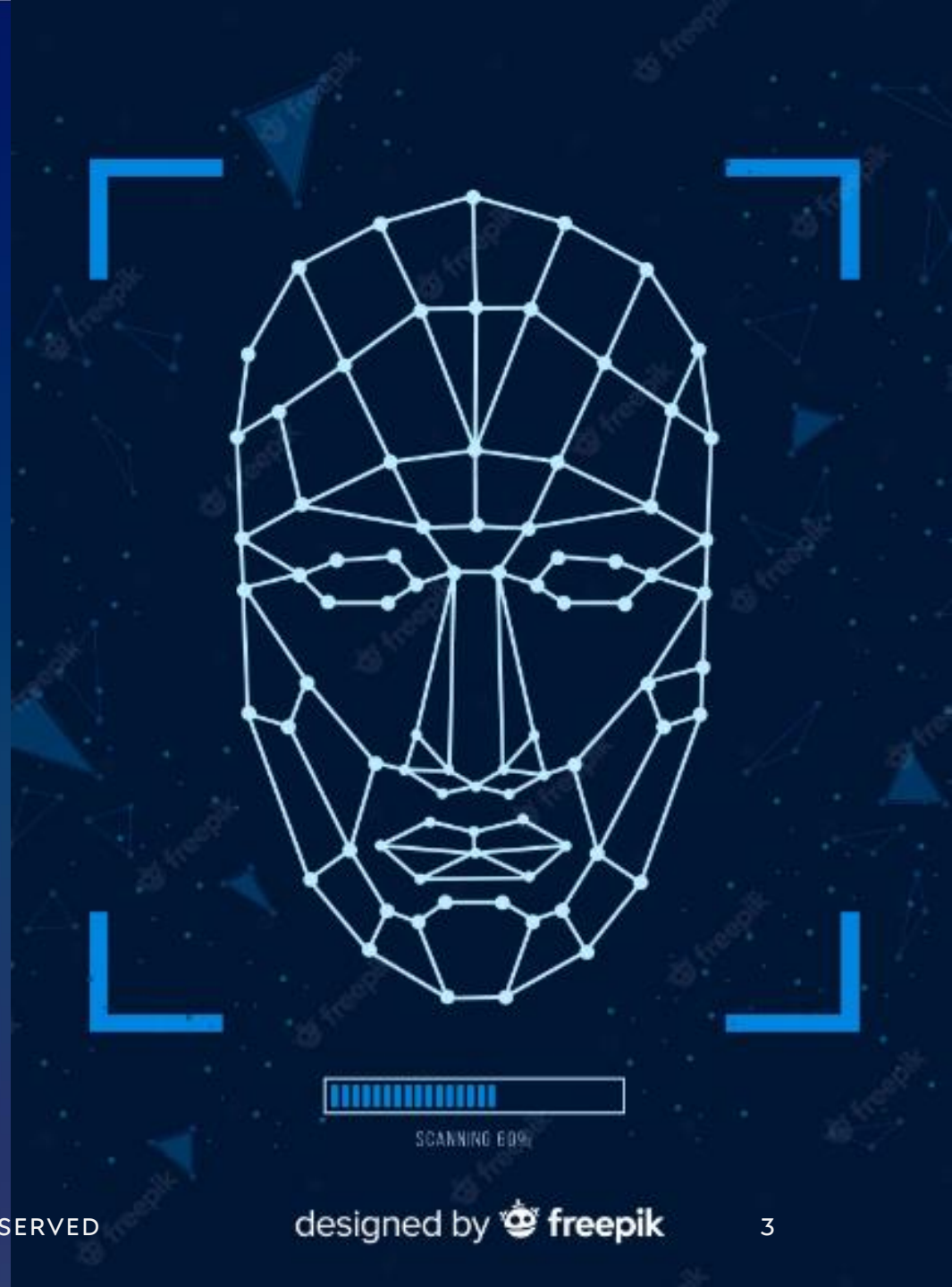
Group C

ABOUT US

Students from INHA UNIVERSITY in South Korea working in a small project from AI Application course leaded by Professor from ISE department Mehdi Pirahandeh based on deep learning and AI. Our project is based on face recognition on healthcare field in order to help both the citizens and assistants in hospital to save time and work efficiently. Let’s take a look at how face recognition impacts healthcare.

◦ HOW DOES FACE RECOGNITION WORK?

- Facial recognition is a technology that helps identify or verify a person based on their specific facial features. Detect human faces in images, videos or live feeds and compare them to a face database
- In other words, all biometric models are mathematical representations of people's faces. In this way, biometric and personal information are strictly separated, ensuring the highest level of privacy, even when dealing with highly sensitive data.
- The facial recognition software calculates the correspondence between the entered facial descriptor and all facial descriptors previously stored in a database. The goal is to find the face or faces from the database that most closely resemble the input face. The higher the match, the more likely the face is to be identified accurately.



PROBLEM

SERVICE IN HOSPITALS

- In the waiting hall when patients take their waiting number, sometimes it takes too much time waiting for your turn, so therefore we should make an automated or checking mechanism that checks the people who left the hall.

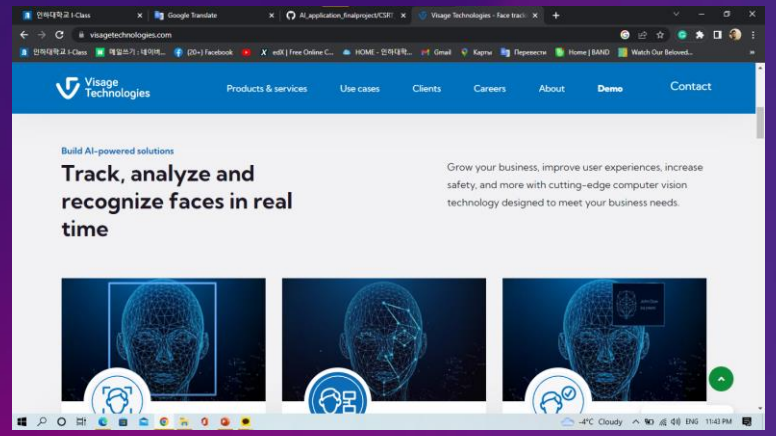
SOLUTION

- Our program makes managing tasks easier, and no other program on the market offers the same benefits. It checks the people faces when they are taking the waiting number and selects how many people they are, so mechanism automatically cancels the number who left the hall or waiting room

PREVIOUS RESEARCHES

1. VISAGE TECHNOLOGIES

https://visagetechnologies.com/face-recognition-in-healthcare/



They have already made a face recognition mechanism in healthcare including security

- Facility security
- Patient diagnosis and monitoring
- Patient check in and check out
- Access control

2.GENETIC BASED FACE RECOGNITION FOR HEALTHCARE APPLICATIONS

- http://hdl.handle.net/10995/84113

They have already made a face recognition research in the field of security authentication

- 3-LEVEL DWT
- GENETICALLY CLOSER
- MOBILE TERMINAL

PROJECT OVERVIEW

UNIQUE

Only program specifically dedicated to hospital management

FIRST TO MARKET

First beautifully designed program that's both stylish and functional

TESTING

Conducted testing with datasets we have and planning to test in a real atmosphere in future

AUTHENTIC

Designed with the help and input of professors in this area

FACE DATA COLLECTION(DATASETS)

Kaggle Dataset's

1. Kaggle

We are going to use Kaggle for datasets that already exists.

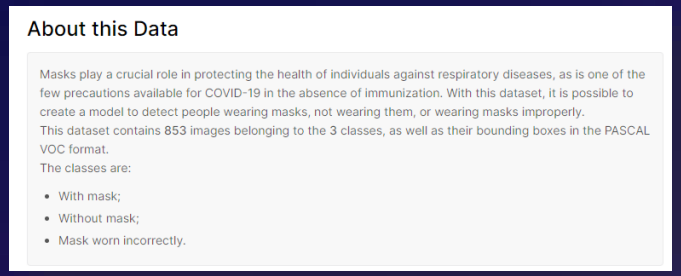
https://www.kaggle.com/datasets/andrewmvd/face-mask-detection

2. VIS datasets

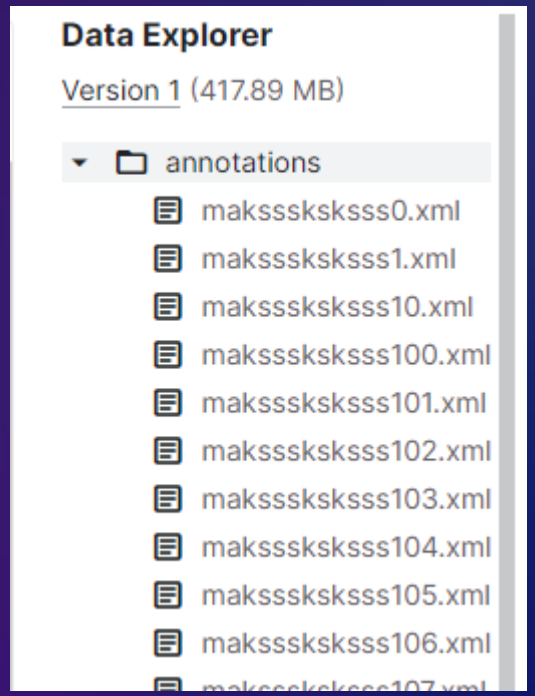
the link, on this one there is 13,000 thousand image, the one I download was name "**all images as zipped tar file**“

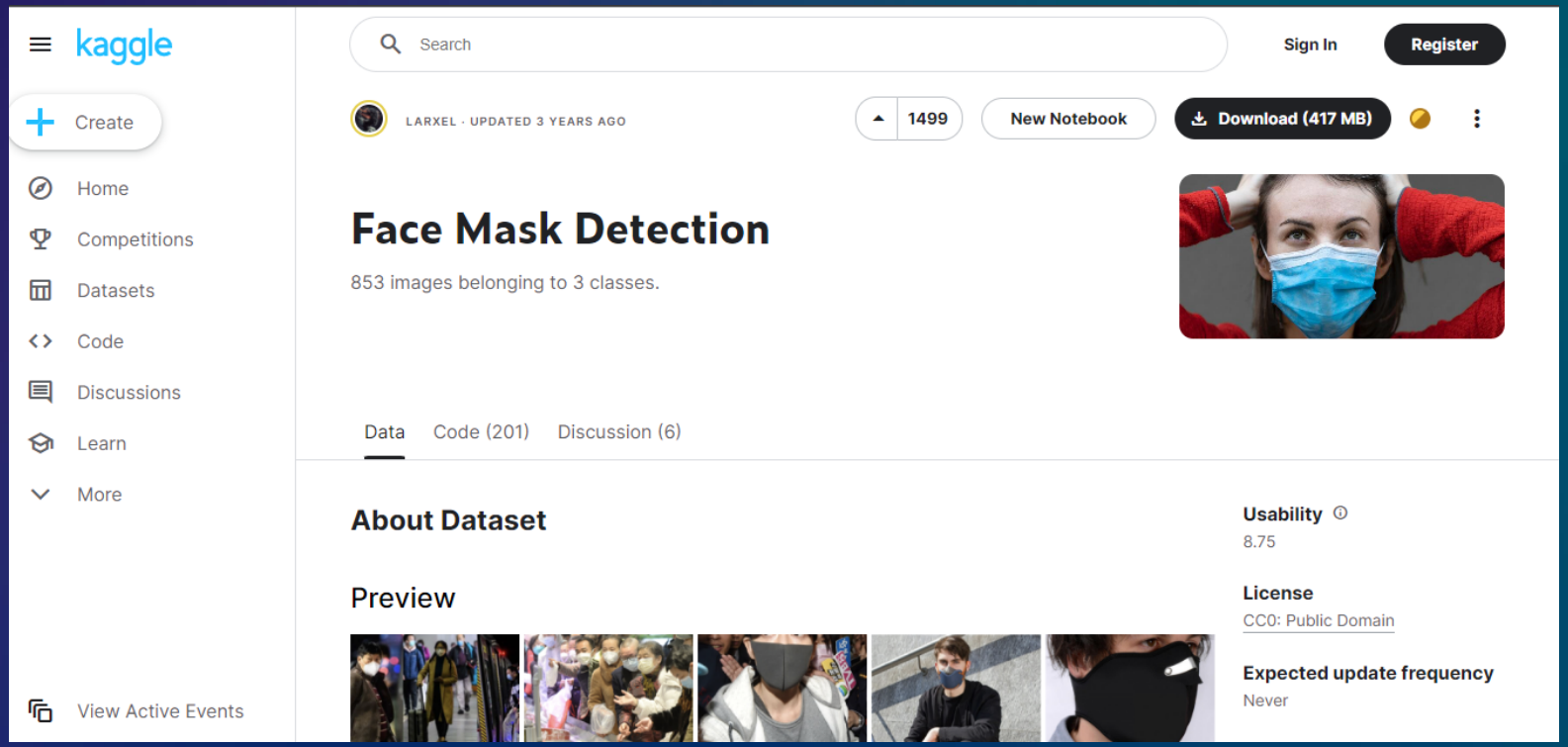
http://vis-www.cs.umass.edu/lfw/

Viola & Jones algorithm using OpenCV



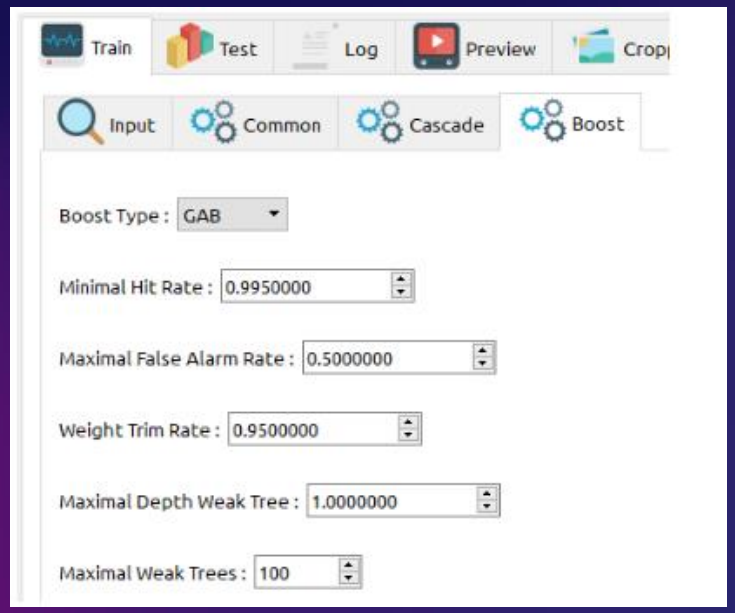
- Ready images
- Already annotated
- in a cascade classifier format xml
- Can be directly trained

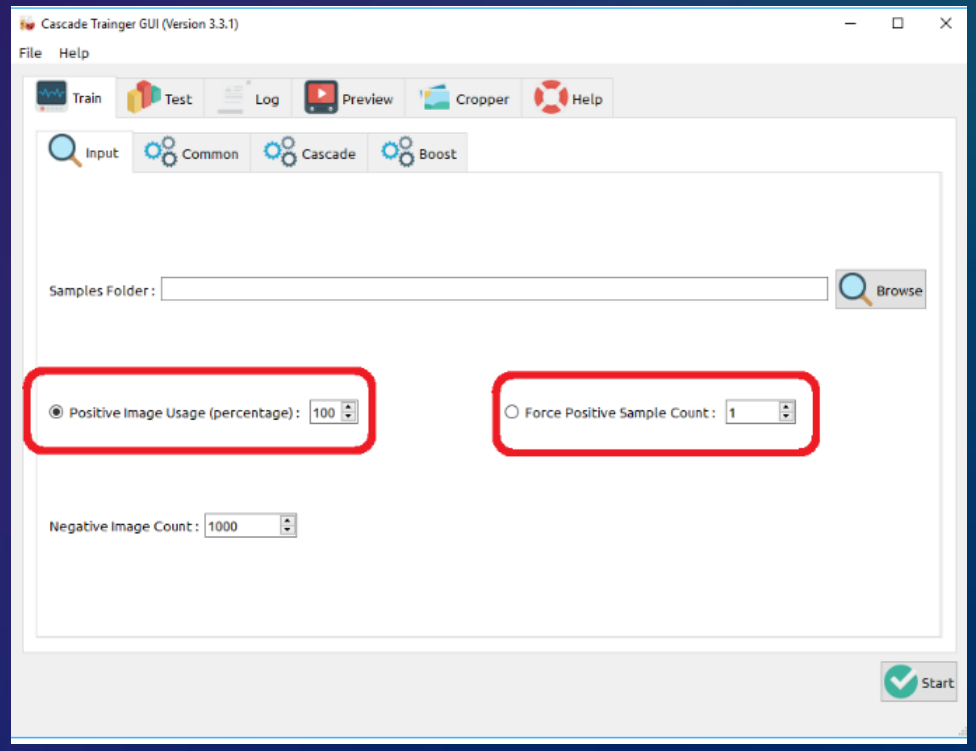


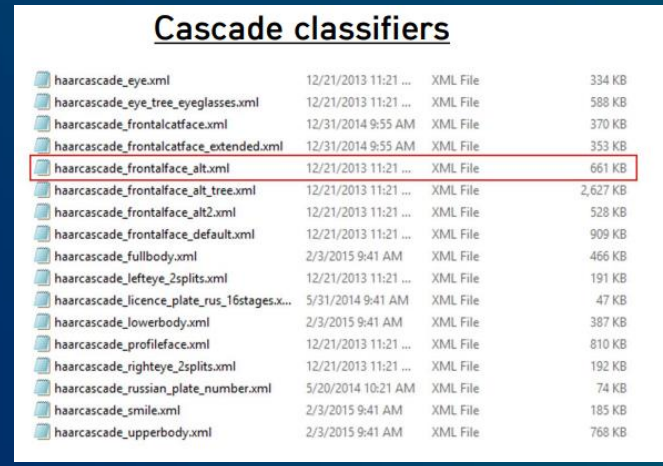


https://www.kaggle.com/datasets/andrewmvd/face-mask-detection?select=annotations

GUI cascade trainer for training our datasets with giving positive and negative images



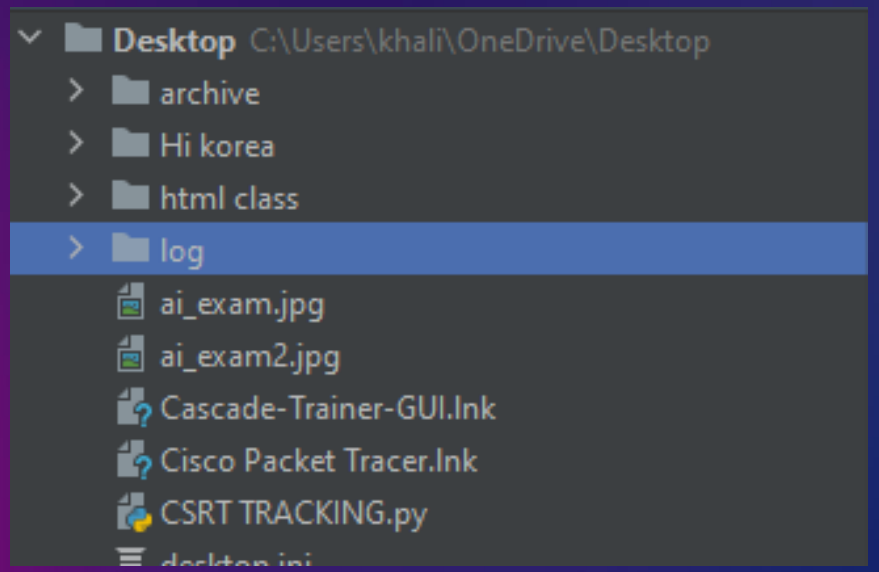




CSRT (Discriminative Correlation Filter with Channel and Spatial Reliability) tracker

PyCharm

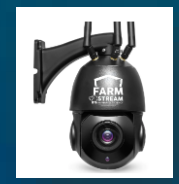
OpenCV + Python




```python
#create tracker
tracker = cv2.TrackerCSRT_create()
```

```python
# Read video
video = cv2.VideoCapture("walking.mp4")
```

```python
# Read video
video = cv2.VideoCapture(0)
```



```python
# Initialize tracker with first frame and bounding box
ok = tracker.init(frame, bbox)

while True:
    # Read a new frame
    ok, frame = video.read()
    if not ok:
        break
```

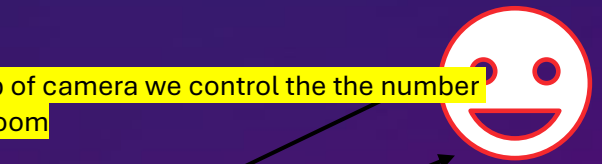
```python
# Display result
cv2.imshow("Tracking", frame)
```

- We are using python code, with trained datasets
- Using PyCharm to run and boot
- we used a video to test it
- We can also use live camera capture to track

HOW DOES IT WORKS?

1. Face detection with a camera when in the first time

2. With the help of camera we control the the number of people in a room

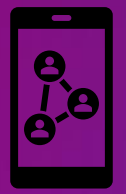


people

Live camera on air

5. People with their ID wil l be stored and when it loses detection on specific ID , it deletes the from list

MEMORY



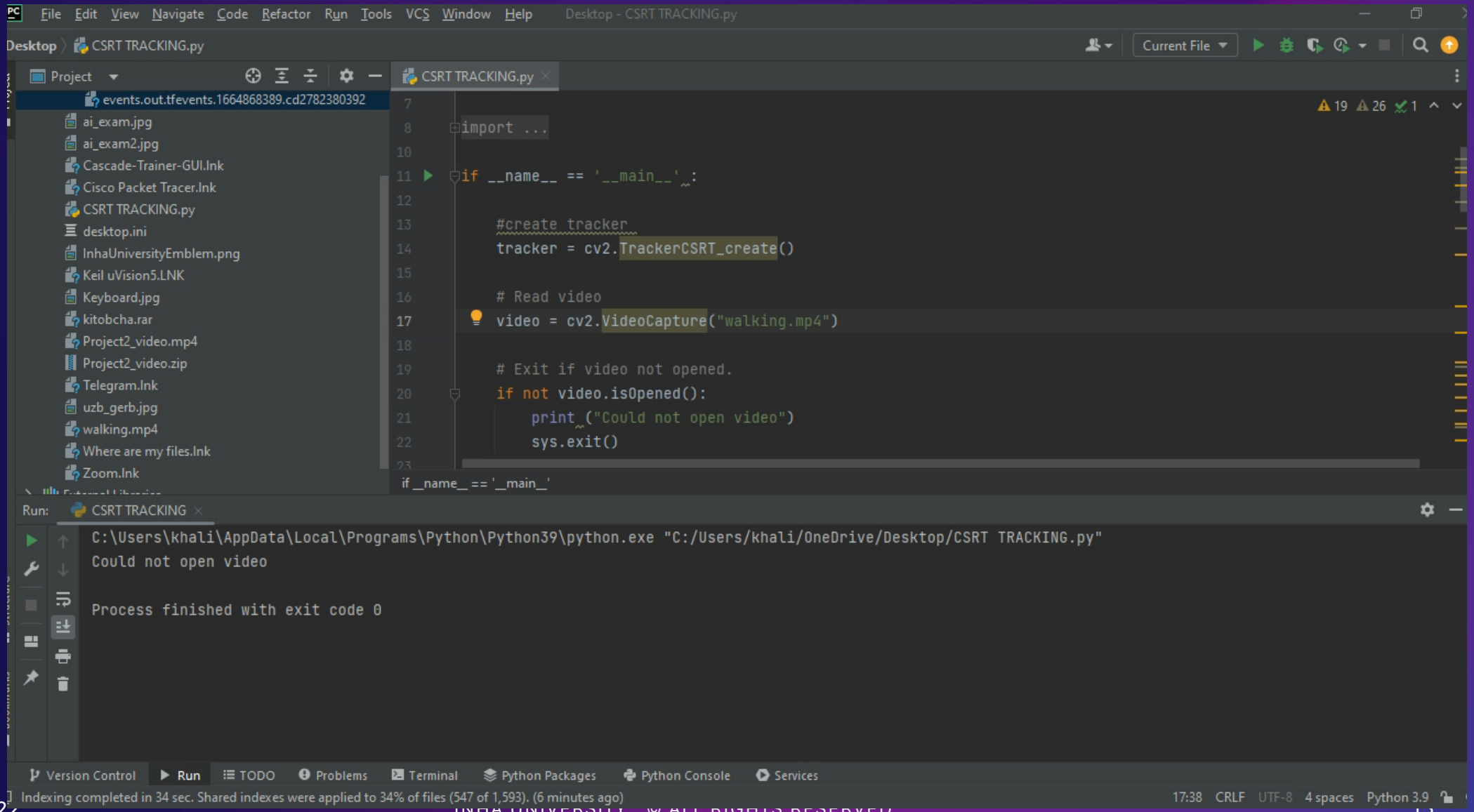
PC

OpenCV

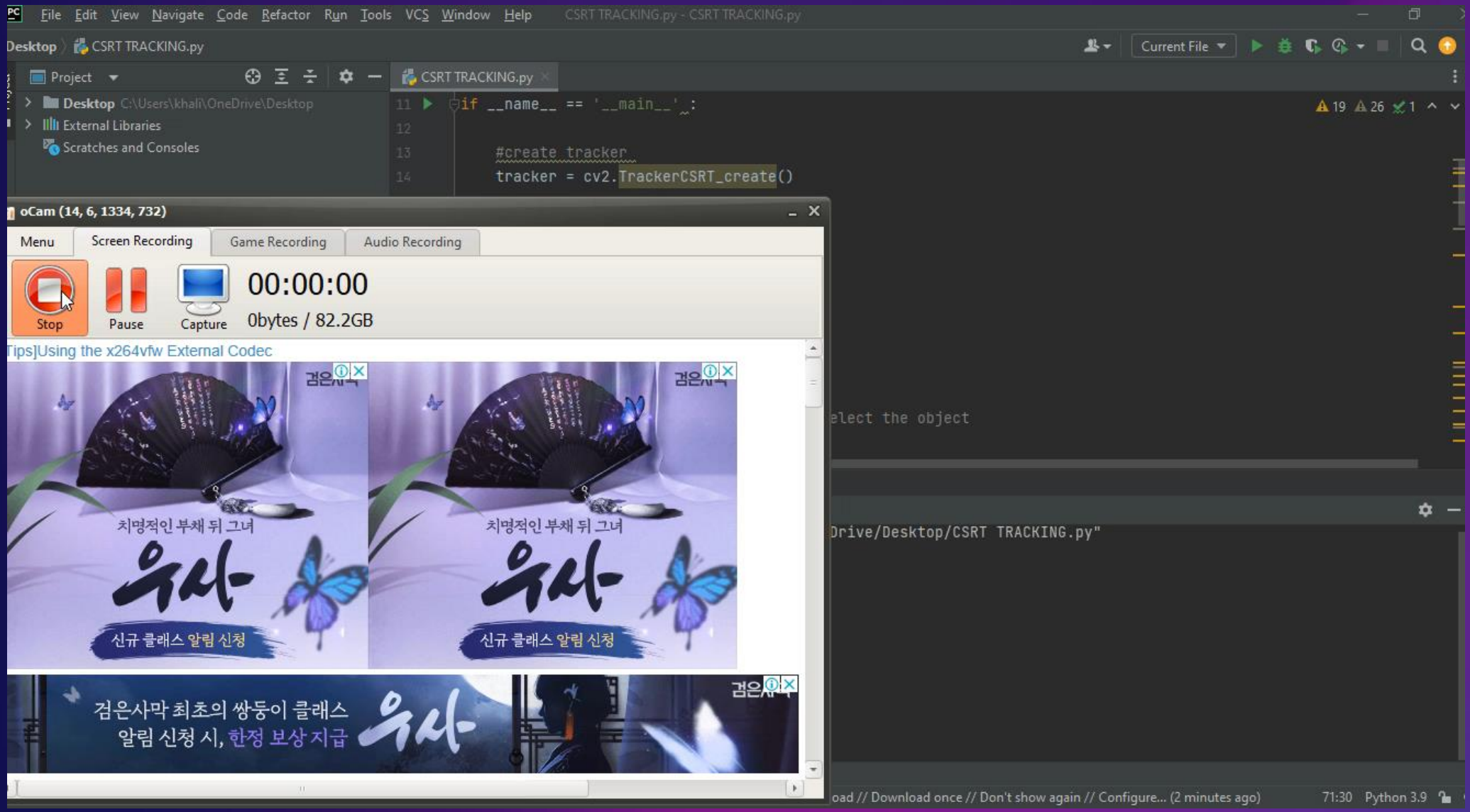
3. Local PC stores the video and uses tracker with Opencv and manage every persons face with specific ID

4. OpenCV support different trackers and libraries

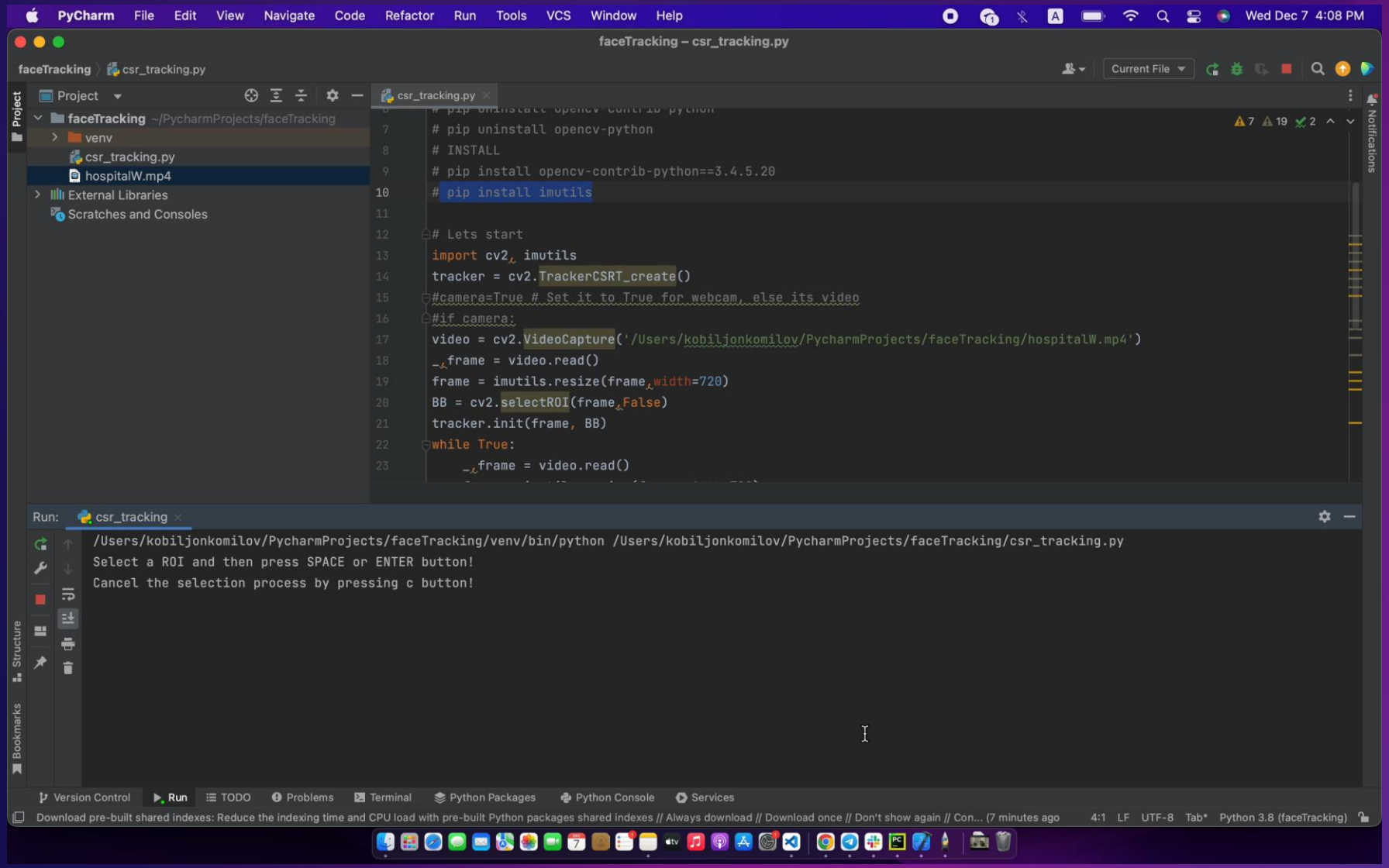
THE OUTPUT EXAMPLE OF CSRT TRACKING USING OPENCV



THE OUTPUT EXAMPLE OF CSRT TRACKING USING OPENCV



THE OUTPUT EXAMPLE OF TRACKING USING OPENCV



SPECIFICATION

In our scenario, we are planning to implement face recognition using OpenCV and Python.

Here are the necessary libraries that we will use in our project:

1.OPENCV

2.DLIB

3.FACE_RECOGNITION

EXTRACT FACIAL FEATURES

First, it is necessary to have a dataset. The dataset should contain all images in folders with each folder containing images of the people.

THE PYTHON CODE FOR FACE RECOGNITION

```
from imutils import paths #imutils includes opencv functions
import face_recognition
import pickle
import cv2
import os
```

```
#get paths of each file in folder named Images
#Images here that contains data(folders of various people)
imagePath = list(paths.list_images('Images'))
kEncodings = []
kNames = []
```

```
# loop over the image paths
for (i, ip) in enumerate(imagePath):
# extract the person name from the image path
name = ip.split(os.path.sep)[-2]
# load the input image and convert it from BGR
image = cv2.imread(ip)
rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
```

THE PYTHON CODE FOR FACE RECOGNITION(CONT.)

```
boxes = face_recognition.face_locations(rgb,model='hog')
# compute the facial embedding for the any face
encodings = face_recognition.face_encodings(rgb, boxes)
# loop over the encodings
for encoding in encodings:
kEncodings.append(encoding)
kNames.append(name)
```

```
#save emcodings along with their names in dictionary data
data = {"encodings": kEncodings, "names": kNames}
#use pickle to save data into a file for later use
f = open("face_enc", "wb")
f.write(pickle.dumps(data))#to open file in write mode
f.close()#to close file
```

HOW TO RECOGNIZE FACE IN IMAGES

```
import face_recognition
import imutils #imutils includes opencv functions
import pickle
import time
import cv2
import os
```

```
#to find path of xml file containing haarCascade file
cfp = os.path.dirname(cv2.__file__) + "/data/haarcascade_frontalface_alt2.xml"
# load the harcaascade in the cascade classifier
fc = cv2.CascadeClassifier(cfp)
# load the known faces and embeddings saved in last file
data = pickle.loads(open('face_enc', "rb").read())
```

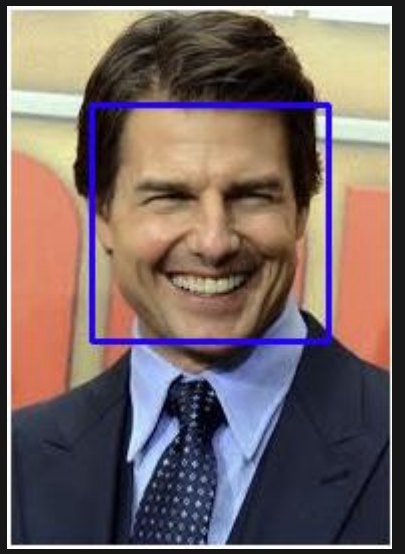
```
#Find path to the image you want to detect face and pass it here
image = cv2.imread(Path-to-img)
rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
#convert image to Greyscale for HaarCascade
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
faces = fc.detectMultiScale(gray,
scaleFactor=1.1,
minNeighbors=6,
minSize=(60, 60),
flags=cv2.CASCADE_SCALE_IMAGE)
```

HOW TO RECOGNIZE FACE IN IMAGES(CONT.)

```
encodings = face_recognition.face_encodings(rgb)
names = []
# loop over the facial embeddings incase
# we have multiple embeddings for multiple fcaes
for encoding in encodings:
#Compare encodings with encodings in data["encodings"]
#Matches contain array with boolean values True and False
matches = face_recognition.compare_faces(data["encodings"],
encoding)
#set name =unknown if no encoding matches
name = "Unknown"
# check to see if we have found a match
if True in matches:
#Find positions at which we get True and store them
matchedIdxs = [i for (i, b) in enumerate(matches) if b]
count = {}
# loop over the matched indexes and maintain a count for
# each recognized face face
for i in matchedIdxs:
#Check the names at respective indexes we stored in matchedIdxs
name = data["names"][i]
#increase count for the name we got
count[name] = count.get(name, 0) + 1
#set name which has highest count
name = max(count, key=count.get)
# will update the list of names
names.append(name)
```

```
# do loop over the recognized faces
for ((x, y, w, h), name) in zip(faces, names):
# rescale the face coordinates
# draw the predicted face name on the image
cv2.rectangle(image, (x, y), (x + w, y + h), (0, 255, 0), 2)
cv2.putText(image, name, (x, y), cv2.FONT_HERSHEY_SIMPLEX,
0.75, (0, 255, 0), 2)
cv2.imshow("Frame", image)
cv2.waitKey(0)
```

OUTPUT



GITHUB LINK

https://github.com/Khai2708/AI_application_finalproject.git
All source files are uploaded to Github

SOURCE FILE

CSRT_FACE_TRACKING.py

```python
# -*- coding: utf-8 -*-
"""
Spyder Editor

This is a temporary script file.
"""

import cv2
import sys

if __name__ == '__main__' :

    #create tracker
    tracker = cv2.TrackerCSRT_create()

    # Read video
    video = cv2.VideoCapture("walking.mp4")

    # Exit if video not opened.
    if not video.isOpened():
        print ("Could not open video")
        sys.exit()

        # Read first frame to be able to select the object
    ok, frame = video.read()
    if not ok:
        print ('Cannot read video file')
        sys.exit()
```

https://github.com/Khai2708/AI_application_finalproject/blob/main/CSRT_FACE_TRACKING.py

SUMMARY

The implementation of facial recognition technology in healthcare has transformed many surgeries for the better. From improved patient safety and identification to better patient tracking and diagnosis, this has helped improve the patient experience and reduce the workload of healthcare workers.

Facial recognition will be even more ubiquitous in the future than it is today. Advanced health monitoring and care robots are just some of the applications being developed. The potential for facial recognition in healthcare is huge, and we haven't seen the best of it yet.

THANK YOU

AI APPLICATION COURSE

Group C

ASADBEK 12194882

ULUGBEK 12194914

ABDULLOKH 12194900

KOBILJON 12194939

KALANDAROV BEKZODBEK