CODE for FACE-RECOGNITION

Haar cascade to separate images and then face recognition library to compare images

That's it for now DATE-10 September

import cv2

import numpy as np

import face\_recognition

import os

import sys

imagePath = 'ImagesBasic/img.jpg'

image = cv2.imread(imagePath)

gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

faceCascade = cv2.CascadeClassifier(cv2.data.haarcascades + "haarcascade\_frontalface\_default.xml")

faces = faceCascade.detectMultiScale(

gray,

scaleFactor=1.3,

minNeighbors=3,

minSize=(30, 30)

)

print("[INFO] Found {0} Faces.".format(len(faces)))

for (x, y, w, h) in faces:

cv2.rectangle(image, (x, y), (x + w, y + h), (0, 255, 0), 2)

roi\_color = image[y:y + h, x:x + w]

print("[INFO] Object found. Saving locally.")

path = 'cascaded images'

cv2.imwrite('cascaded images/' + str(w) + str(h) + '\_faces.jpg', roi\_color)

status = cv2.imwrite('faces\_detected.jpg', image)

print("[INFO] Image faces\_detected.jpg written to filesystem: ", status)

for i in (os.listdir("ImageAttendance")):

for j in (os.listdir("cascaded images")):

predefineImg = face\_recognition.load\_image\_file(f'ImageAttendance/{i}')

predefineImg = cv2.cvtColor(predefineImg, cv2.COLOR\_BGR2RGB)

imgCollected = face\_recognition.load\_image\_file(f'cascaded images/{j}')

imgCollected = cv2.cvtColor(imgCollected, cv2.COLOR\_BGR2RGB)

faceLoc = face\_recognition.face\_locations(predefineImg)[0]

encodeElon = face\_recognition.face\_encodings(predefineImg)[0]

cv2.rectangle(predefineImg, (faceLoc[3], faceLoc[0]), (faceLoc[1], faceLoc[2]), (255, 0, 255), 2)

faceLocTest = face\_recognition.face\_locations(imgCollected)[0]

encodeTest = face\_recognition.face\_encodings(imgCollected)[0]

cv2.rectangle(imgCollected, (faceLocTest[3], faceLocTest[0]), (faceLocTest[1], faceLocTest[2]), (255, 0, 255), 2)

results = face\_recognition.compare\_faces([encodeElon], encodeTest)

faceDis = face\_recognition.face\_distance([encodeElon], encodeTest)

print(results, faceDis)

cv2.putText(predefineImg, f'{results} {round(faceDis[0], 2)}', (50, 50), cv2.FONT\_HERSHEY\_COMPLEX, 1, (0, 0, 255), 2)

cv2.imshow('people', predefineImg)

cv2.imshow('test subject', imgCollected)

cv2.waitKey(0)