SANDHIYA KANTHAKUMAR - 118A1031

USHA KONAR - 118A1035

GUIDE- MR. SUNIL PUNJABI

**PROJECT COMPLETE CODE**

from cv2 import cv2

import os

import numpy as np

from PIL import Image

import datetime

import csv

from tkinter import \*

import pandas as pd

import time

window=Tk()

window.title("Attendance System")

message = Label(window, text="Attendance-Management-System" ,bg="Green" ,fg="white" ,width=50 ,height=3,font=('times', 30, 'italic bold underline'))

message.place(x=200, y=20)

lbl =Label(window, text="Enter ID",width=20 ,height=2 ,fg="red" ,bg="yellow" ,font=('times', 15, ' bold ') )

lbl.place(x=400, y=200)

txt = Entry(window,width=20 ,bg="yellow" ,fg="red",font=('times', 15, ' bold '))

txt.place(x=700, y=215)

lbl2 = Label(window, text="Enter Name",width=20 ,fg="red" ,bg="yellow" ,height=2 ,font=('times', 15, ' bold '))

lbl2.place(x=400, y=300)

txt2 =Entry(window,width=20 ,bg="yellow" ,fg="red",font=('times', 15, ' bold ') )

txt2.place(x=700, y=315)

message2 = Label(window, text="" ,fg="red" ,bg="yellow",activeforeground = "green",width=30 ,height=2 ,font=('times', 15, ' bold '))

message2.place(x=700, y=650)

def dataset():

Id=(txt.get())

name=(txt2.get())

face\_classifier=cv2.CascadeClassifier('haarcascade\_frontalface\_default.xml')

#file\_name=input("Enter name:")

path='/Users/usha/Desktop/project/data/'

for files in os.walk(path):

file\_name = '{}'.format(Id)

file=os.path.join(path,file\_name)

if not os.path.exists(file):

os.mkdir(file)

def face\_crop(img):

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

faces=face\_classifier.detectMultiScale(gray,1.3,5)

if faces is():

return None

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

offset=5

cropped\_face=frame[y-offset:y+h+offset,x-offset:x+w+offset]

return cropped\_face

cap=cv2.VideoCapture(0)

img\_id=0

while(True):

ret,frame=cap.read()

if face\_crop(frame) is not None:

img\_id+=1

face=cv2.resize(face\_crop(frame),(500,500))

face=cv2.cvtColor(face,cv2.COLOR\_BGR2GRAY)

#file\_path='file/'+str(img\_id)+".jpg"

#file\_path="data/"+file\_name+"\_"+str(img\_id)+".jpg"

file\_path=os.path.join(file,"{}.jpg").format(str(img\_id))

cv2.imwrite(file\_path,face)

cv2.putText(face,str(img\_id),(50,50),cv2.FONT\_HERSHEY\_COMPLEX,1,(0,255,0),2)

cv2.imshow("Cropped face",face)

if cv2.waitKey(1)==13 or int(img\_id)==60:

break

cap.release()

cv2.destroyAllWindows()

row = [Id , name]

with open('StudentDetails.csv','a+') as csvFile:

writer = csv.writer(csvFile)

writer.writerow(row)

csvFile.close()

def train\_img():

recognizer = cv2.face.LBPHFaceRecognizer\_create() # instantiate the lbph recognizer

path = '/Users/usha/Desktop/project/data/' #setting out path of folders with images

file='/Users/usha/Desktop/project/recognizer/'

if not os.path.exists(file):

os.makedirs(file) # making a directory for yml file which will be generated after training

def getImageswithId(path):

faces = []

labels = []

for root,directory,filenames in os.walk(path):

for filename in filenames:

label = os.path.basename(root) #this directly assigns folder name ie 0,1..

img\_path = os.path.join(root,filename)

print('img\_path:',img\_path)

print('label:',label)

test\_img = cv2.imread(img\_path)

# test\_img = np.float32(test\_img)

if test\_img is None:

print('image not loaded poperly - cv2 cant read!!')

continue

#if images in dataset are not in gray scale then use below 3 line

gray\_img=cv2.cvtColor(test\_img,cv2.COLOR\_BGR2GRAY)#convert color image to grayscale

face\_cascade=cv2.CascadeClassifier('haarcascade\_frontalface\_default.xml')#Load haar classifier

face=face\_cascade.detectMultiScale(gray\_img,scaleFactor=1.32,minNeighbors=5)#detectMultiScale returns rectangles

if len(face)!=1:

continue # since we are asuuming only single person images are being fed to classifier

(x,y,w,h) = face[0]

gray = gray\_img[y:y+h,x:x+h]

equ = cv2.equalizeHist(gray)

final = cv2.medianBlur(equ, 3)

faces.append(final)

labels.append(int(label))

cv2.destroyAllWindows()

return faces,labels

faces,labels = getImageswithId(path)

recognizer.train(faces,np.array(labels))

recognizer.save('/Users/usha/Desktop/project/recognizer/trainingData.yml')

def TrackImages():

recognizer = cv2.face.LBPHFaceRecognizer\_create()#cv2.createLBPHFaceRecognizer()

recognizer.read('/Users/usha/Desktop/project/recognizer/trainingData.yml')

faceCascade = cv2.CascadeClassifier('haarcascade\_frontalface\_default.xml')

df=pd.read\_csv('StudentDetails.csv')

cam = cv2.VideoCapture(0)

font = cv2.FONT\_HERSHEY\_SIMPLEX

col\_names = ['Id','Name','Date','Time']

attend = pd.DataFrame(columns=col\_names)

while True:

ret, im=cam.read()

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

equ = cv2.equalizeHist(gray)

final = cv2.medianBlur(equ, 3)

faces=faceCascade.detectMultiScale(final, 1.3,5)

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

cv2.rectangle(im,(x,y),(x+w,y+h),(0,255,0),3)

label, conf = recognizer.predict(gray[y:y+h,x:x+w])

print(conf)

print(label)

if(conf < 90):

ts = time.time()

date = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d')

timeStamp = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')

aa=df.loc[df['Id'] == label]['name'].values

tt=str(label)+"-"+aa

#Attendance.loc[len(attend)] = [label,aa,date,timeStamp]

row=[label,aa,date,timeStamp]

with open('Attendance.csv','r+') as csvFile:

writer = csv.writer(csvFile)

writer.writerow(row)

csvFile.close()

else:

Id='Unknown'

tt=str(Id)

# if(conf > 75):

# noOfFile=len(os.listdir("data"))+1

# cv2.imwrite("data"+str(noOfFile) + ".jpg", im[y:y+h,x:x+w])

cv2.putText(im,str(tt),(x,y+h), font, 1,(255,255,255),2)

attend=attend.drop\_duplicates(subset=['Id'],keep='first')

cv2.imshow('im',im)

if (cv2.waitKey(1)==ord('q')):

break

Id=label

Name=aa

#attendance(Name,Id)

ts = time.time()

date = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d')

timeStamp = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')

Hour,Minute,Second=timeStamp.split(":")

#filename="Attendance\_list\_"+date+"\_"+Hour+"-"+Minute+"-"+Second+".csv"

attend.to\_csv('Attendance\_list.csv',index=False)

cam.release()

cv2.destroyAllWindows()

res=attend

message2.configure(text= res)

#row=[Id,Name,date,timeStamp]

#with open('Attendance.csv','r+') as csvFile:

# writer = csv.writer(csvFile)

# writer.writerow(row)

#csvFile.close()

#print(attendance)

btn1=Button(window, text="New Attendee" ,command=dataset, fg="red" ,bg="yellow" ,width=20 ,height=2 ,activebackground = "Red" ,font=('times', 15, ' bold '))

btn1.place(x=400, y=400)

btn2=Button(window, text="Train Dataset" , command=train\_img, fg="red" ,bg="yellow" ,width=20 ,height=2 ,activebackground = "Red" ,font=('times', 15, ' bold '))

btn2.place(x=700, y=400)

btn3=Button(window, text="Mark attendance" , command= TrackImages, fg="red" ,bg="yellow" ,width=20 ,height=2 ,activebackground = "Red" ,font=('times', 15, ' bold '))

btn3.place(x=1000, y=400)

window.mainloop()