Face recognition

####################################################

# Modified by pavan kumar #

# All right reserved to the respective owner #

####################################################

# Import OpenCV2 for image processing

import cv2

# Import numpy for matrices calculations

import numpy as np

import os

def assure\_path\_exists(path):

dir = os.path.dirname(path)

if not os.path.exists(dir):

os.makedirs(dir)

# Create Local Binary Patterns Histograms for face recognization

recognizer = cv2.face.LBPHFaceRecognizer\_create()

assure\_path\_exists("trainer/")

# Load the trained mode

recognizer.read('trainer/trainer.yml')

# Load prebuilt model for Frontal Face

cascadePath = "haarcascade\_frontalface\_default.xml"

# Create classifier from prebuilt model

faceCascade = cv2.CascadeClassifier(cascadePath);

# Set the font style

font = cv2.FONT\_HERSHEY\_SIMPLEX

# Initialize and start the video frame capture

cam = cv2.VideoCapture(0)

# Loop

while True:

# Read the video frame

ret, im =cam.read()

# Convert the captured frame into grayscale

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

# Get all face from the video frame

faces = faceCascade.detectMultiScale(gray, 1.2,5)

# For each face in faces

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

# Create rectangle around the face

cv2.rectangle(im, (x-20,y-20), (x+w+20,y+h+20), (0,255,0), 4)

# Recognize the face belongs to which ID

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

# Check the ID if exist

if(Id == 1):

Id = "Parthu {0:.2f}%".format(round(100 - confidence, 2))

if(Id == 2):

Id = "Pavan {0:.2f}%".format(round(100 - confidence, 2))

if(Id == 3):

Id = "Bhargava {0:.2f}%".format(round(100 - confidence, 2))

# Put text describe who is in the picture

cv2.rectangle(im, (x-22,y-90), (x+w+22, y-22), (0,255,0), -1)

cv2.putText(im, str(Id), (x,y-40), font, 1, (255,255,255), 3)

# Display the video frame with the bounded rectangle

cv2.imshow('im',im)

# If 'q' is pressed, close program

if cv2.waitKey(10) & 0xFF == ord('q'):

break

# Stop the camera

cam.release()

# Close all windows

cv2.destroyAllWindows()