Face detect

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

# Modified by pavan kumar #

# All right reserved to the respective owner #

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

# Import OpenCV2 for image processing

import cv2

import os

def assure\_path\_exists(path):

dir = os.path.dirname(path)

if not os.path.exists(dir):

os.makedirs(dir)

# Start capturing video

vid\_cam = cv2.VideoCapture(0)

# Detect object in video stream using Haarcascade Frontal Face

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

# For each person, one face id

face\_id = 3

# Initialize sample face image

count = 0

assure\_path\_exists("dataset/")

# Start looping

while(True):

# Capture video frame

\_, image\_frame = vid\_cam.read()

# Convert frame to grayscale

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

# Detect frames of different sizes, list of faces rectangles

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

# Loops for each faces

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

# Crop the image frame into rectangle

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

# Increment sample face image

count += 1

# Save the captured image into the datasets folder

cv2.imwrite("dataset/User." + str(face\_id) + '.' + str(count) + ".jpg", gray[y:y+h,x:x+w])

# Display the video frame, with bounded rectangle on the person's face

cv2.imshow('frame', image\_frame)

# To stop taking video, press 'q' for at least 100ms

if cv2.waitKey(100) & 0xFF == ord('q'):

break

# If image taken reach 100, stop taking video

elif count>100:

break

# Stop video

vid\_cam.release()

# Close all started windows

cv2.destroyAllWindows()