FACE RECOGNITION

USING OPENCV

IN PYTHON

BY

SHUBHAM GUPTA (CSE) – 12100115044

SUBHAJIT DEY (CSE) – 12100115055

SWATI JHA (IT) – 12100215055

WAHAB KUKASWADIA (IT) - 12100215058

COLLEGE: BENGAL INSTITUTE OF TECHNOLOGY

***Face Recognition***

Introduction:

Face is most commonly used biometric to recognize people. Face recognition has received substantial attention from researchers due to human activities found in various applications of security like airport, criminal detection, face tracking, forensic etc. Compared to other biometric traits like palm print, Iris, finger print etc., face biometrics can be non-intrusive. They can be taken even without user’s knowledge and further can be used for security based applications like criminal detection, face tracking, airport security, and forensic surveillance systems. Face recognition involves capturing face image from a video or from a surveillance camera. They are compared with the stored database. Face biometrics involves training known images, classify them with known classes and then they are stored in the database. When a test image is given to the system it is classified and compared with stored database. Face biometrics is a challenging field of research with various limitations imposed for a machine face recognition like variations in head pose, change in illumination, facial expression, aging, occlusion due to accessories etc.,. Various approaches were suggested by researchers in overcoming the limitations stated. 72 Automatic face recognition involves face detection, feature extraction and face recognition. Face recognition algorithms are broadly classified into two classes as image template based and geometric feature based. The template based methods compute correlation between face and one or more model templates to find the face identity. Principal component analysis, linear discriminate analysis, kernel methods etc. are used to construct face templates. The geometric feature based methods are used to analyze explicit local features and their geometric relations (elastic bung graph method). Multi resolution tools such as contour lets, ridge lets were found to be useful for analyzing information content of images and found its application in image processing, pattern recognition, and computer vision. Curvelets transform is used for texture classification and image de-noising. Application of Curvelets transform for feature extraction in image processing is still under research.

Source Code:

import cv2

import sys

*# Get user supplied values*

imagePath = sys.argv[1]

cascPath = "haarcascade\_frontalface\_default.xml"

*# Create the haar cascade*

faceCascade = cv2.CascadeClassifier(cascPath)

*# Read the image*

image = cv2.imread(imagePath)

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

*# Detect faces in the image*

faces = faceCascade.detectMultiScale(

gray,

scaleFactor=1.1,

minNeighbors=5,

minSize=(30, 30)

)

print("Found {0} faces!".format(len(faces)))

*# Draw a rectangle around the faces*

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

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

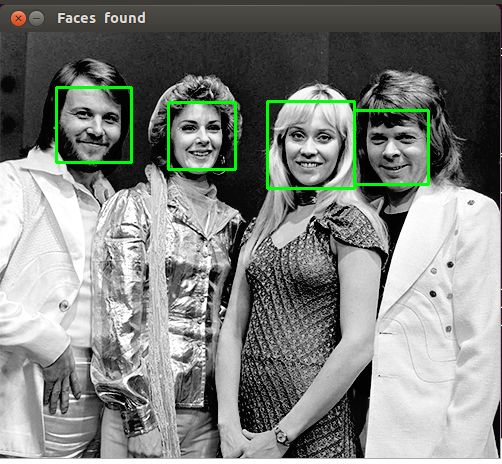
cv2.imshow("Faces found", image)

cv2.waitKey(0)

Input:



Output:



Description:

OpenCV is the most popular library for computer vision. Originally written in C/C++, it now provides bindings for Python.

OpenCV uses machine learning algorithms to search for faces within a picture. Because faces are so complicated, there isn’t one simple test that will tell you if it found a face or not. Instead, there are thousands of small patterns and features that must be matched. The algorithms break the task of identifying the face into thousands of smaller, bite-sized tasks, each of which is easy to solve. These tasks are also called classifiers.

For something like a face, you might have 6,000 or more classifiers, all of which must match for a face to be detected (within error limits, of course). But therein lies the problem: for face detection, the algorithm starts at the top left of a picture and moves down across small blocks of data, looking at each block, constantly asking, “Is this a face? … Is this a face? … Is this a face?” Since there are 6,000 or more tests per block, you might have millions of calculations to do, which will grind your computer to a halt.

To get around this, OpenCV uses cascades. What’s a cascade? The best answer can be found in the dictionary: “a waterfall or series of waterfalls.” Like a series of waterfalls, the OpenCV cascade breaks the problem of detecting faces into multiple stages. For each block, it does a very rough and quick test. If that passes, it does a slightly more detailed test, and so on. The algorithm may have 30 to 50 of these stages or cascades, and it will only detect a face if all stages pass.

The advantage is that the majority of the picture will return a negative during the first few stages, which means the algorithm won’t waste time testing all 6,000 features on it. Instead of taking hours, face detection can now be done in real time.

Since face detection is such a common case, OpenCV comes with a number of built-in cascades for detecting everything from faces to eyes to hands to legs. There are even cascades for non-human things

Future Aspects:

Over the past few years, we have seen major developments to facial recognition technology. We now live in an age that our predecessors have only dreamed in fiction and film. Face recognition is one of the newer developments of biometric identifiers that doesn’t require as much time or intrude on the person its verifying. Other biometric identifiers, such as fingerprint scanners and voice recognition, requires many different pieces in order to function. Face recognition is a highly effective biometric technology that holds a lot of potential.

Today, one of the fields that uses facial recognition the most is security. Facial recognition is a very effective tool that can help law enforcers recognize criminals and software companies are leveraging the technology to help users access their technology. This technology can be further developed to be used in other avenues such as ATMs, accessing confidential files, or other sensitive materials. This can make other security measures such as passwords and keys obsolete.

Another way that innovators are looking to implement facial recognition is within subways and other transportation outlets. They are looking to leverage this technology to use faces as credit cards to pay for your transportation fee. Instead of having to go to a booth to buy a ticket for a fare, the face recognition would take your face, run it through a system, and charge the account that you’ve previously created. This could potentially streamline the process and optimize the flow of traffic drastically.