Face Detection and Recognition System

Parvesh Shelke, Ayush G Bokade, Piyush Digambar Vairagade, Tanmay Dinesh Bhoyar, Tejal Hirekhan
Swaminarayan Siddhanta Institute of Technology,
Nagpur, India

Abstract - With every passing day, we are becoming more and more dependent upon technology to carry out even the most basic of our actions. Facial detection and Facial recognition help us in many ways, be it sorting of photos in our mobile phone gallery by recognizing pictures with their face in them or unlocking a phone by a mere glance to adding biometric information in the form of face images in the country's unique ID database (Aadhaar) as an acceptable biometric input for verification. This project lays out the basic terminology required to understand the implementation of Face Detection and Face Recognition using Intel's Computer Vision library called 'OpenCV'. It also shows the practical implementation of the Face Detection and Face Recognition using OpenCV with Python embedding on both Windows as well as macOS platform. The aim of the project is to implement Facial Recognition on faces that the script can be trained for. The input is taken from a webcam and the recognized faces are displayed along with their name in real time. This project can be implemented on a larger scale to develop a biometric attendance system which can save the time-consuming process of manual attendance system.

***How to cite this article**: Parvesh Shelke, Ayush G Bokade, Piyush Digambar Vairagade, Tanmay Dinesh Bhoyar, Tejal Hirekhan "Face Detection and Recognition System" Published in International Journal of Scientific Modern Research and Technology (IJSMRT), ISSN: 2582-8150, Volume-14, Issue-3, Number-1, February 2024, pp.1-5, URL: https://www.ijsmrt.com/wp-content/uploads/2024/05/ IJSMRT-24140301.pdf*

IJSMRT-24140301

I. INTRODUCTION

A face recognition system could also be a technology which is very capable of matching a personality's face from a digital image or a video frame which it has or use it as a reference to map and identify against an info of faces. Researchers' area unit presently developing multiple ways throughout that face recognition systems work. the foremost advanced face recognition methodology, that is to boot used to manifest users through ID verification services, works by pinpointing and mensuration countenance from a given image.

While at first a kind of laptop application, face recognition systems have seen wider uses in recent times on smartphones and in alternative kinds of technology, like artificial intelligence. as a result of computerized face recognition involves the measuring of a human's physiological characteristics face recognition systems area unit classified as bioscience. though the accuracy of face recognition systems as a biometric technology is a smaller amount than iris recognition and fingerprint recognition, it's wide adopted because of its contactless and non-invasive method. Facial recognition systems area unit deployed in advanced human -computer interaction, video police work and automatic compartmentalization of pictures. We have a created a face recognition technology capable of identifying faces.

II. LITERATURE REVIEW

FACE TRACKING: MECHANISMS OF HUMAN FACIAL RECOGNITION

“The logical (computational) role suggested for the primary visual cortex has several components: size standardization, size reduction, and object extraction”.

“The result of processing by the primary visual cortex, it is suggested, is a neural encoding of the Basically what we see in this paper is that it presents an extension and a new way of perception of the author's theory for human visual information processing, which method includes extracting a second sub-image from the second image, where the second sub-image includes a representation of the at least one corresponding facial landmark. “In turn detecting a facial gesture by determining whether a sufficient difference exists between the second sub-image and first sub-image to indicate the facial gesture, and determining, based on detecting the facial gesture, whether to deny authentication to the user with respect to the human recognition system and same was applied” Several indispensable techniques are implicated: encoding of visible photographs into neural patterns, detection of easy facial features, measurement standardization, discount of the neural patterns in dimensionality [2].

Face tracking refers to identifying the features which are then used to detect a Face In this case the example method includes the receiving or we can say that it gets the first image and the second images of a face of a user who is being taken into consideration, where one or both of the images which were used to sort of look for a match have been granted a match by the facial recognition system which also proofs the correct working of the system. “The technique includes taking out a second sub- image coming from the second image, where the second sub-image includes a representation of the at least one corresponding facial landmark, detecting a facial gesture by determining whether a sufficient difference exists between the second sub - image and first sub-image to indicate the facial gesture, and determining, based on detecting the facial gesture, whether to deny authentication to the user with respect to accessing functionalities controlled by the computing” [1]

visual pattern at a size suitable for storage. “(In this context, object extraction is the isolation of regions in the visual field having the same color, texture, or spatial extent.)” It is shown in detail how the topology of the mapping from retina to cortex, the connections between retina, lateral geniculate bodies and primary visual cortex, and the local structure of the cortex itself may combine to encode the visual patterns. Aspects of this theory are illustrated graphically with human faces as the primary stimulus. However, the theory is not limited to facial recognition but pertains to Gestalt recognition of any class of familiar objects or scenes [2].

Few procedures to computerized facial consciousness have employed geometric size of attribute points of a human face. Eye spacing dimension has been recognized as an essential step in reaching this goal. Measurement of spacing has been made by means of software of the Hough radically change method to discover the occasion of a round form and of an ellipsoidal form which approximate the perimeter of the iris and each the perimeter of the sclera and the form of the place under the eyebrows respectively. Both gradient magnitude and gradient direction were used to handle the noise contaminating the feature space. “Results of this application indicate that measurement of the spacing by detection of the iris is the most accurate of these three methods with measurement by detection of the position of the eyebrows the least accurate. However, measurement by detection of the eyebrows' position is the least constrained method. Application of these strategies has led to size of a attribute function of the human face with adequate accuracy to advantage later inclusion in a full bundle for computerized facial consciousness”. [3].

A DIRECT LDA ALGORITHM FOR HIGH-DIMENSIONAL DATA WITH APPLICATION TO FACE RECOGNITION

“Linear discriminant analysis (LDA) has been successfully used as a dimensionality reduction technique to many classification problems, such as speech recognition, face recognition, and multimedia information retrieval” The objective is to "ND a projection A that maximizes the ratio of between-class scatter against within-class.

III. OBJECTIVES

This project is created so as to study the various means of recognizing faces with more accuracy and reducing the error rates while recognition. The ideal condition for any recognition project is to reduce the intra class variance of features and increase the inter class variance of features to be detected or recognized. Facial Recognition software is "Capable of uniquely identifying or verifying a person by comparing and analyzing patterns based on the person's facial contours.

IV. RESEARCH METHODOLOGY

The Project utilizes various libraries of Python such as.

A. OpenCV

"OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library". The main purpose of this was to provide a common infrastructure for computer vision applications and it was also built specifically for such purposes not to mention it also accelerated the use of machine perception inside the business product. "Being a BSD-licensed product, OpenCV makes it straightforward for businesses to utilize and modify the code". In total we can say that the library has about 2500 optimized algorithms which is really insane, "These algorithms contain a comprehensive set which comprises of each classic and progressive laptop vision and machine learning algorithms. These algorithms area unit usually accustomed sight and acknowledge faces, determine objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, manufacture 3D purpose clouds from stereo cameras, sew pictures along to produce a high resolution image of a full scene, realize similar pictures from an image info, take away red eyes from pictures taken exploitation flash, follow eye movements, acknowledge scenery and establish markers to overlay it with increased reality, etc". The amazing thing about this library is that it has quite about forty-seven thousand individuals of user community and calculable variety of downloads Olympian eighteen million. The library is utilized extensively in corporations, analysis teams and by governmental bodies. Along with well-established corporations like "Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota" that use the library, there area unit several startups like "Applied Minds, VideoSurf, and Zeitera", that create in depth use of OpenCV. OpenCV's deployed wide array spans vary from sewing street view pictures along, police work intrusions in police work video in Israel, watching mine instrumentality in China, serving to robots navigate and devour objects at "Willow Garage, detection of natatorium drowning accidents in Europe, running interactive art in Espana and New York, checking runways for scrap in Turkey", inspecting labels on product in factories around the world on to fast face detection in Japan.

B. NumPy

"The Python programming language earlier wasn't originally designed for numerical computing as we know it to be, however it also attracted the attention of the scientific and engineering community early" . "In 1995 the interest (SIG) matrix-sig was based with the aim of shaping associate array computing package; among its members was Python designer and supporter Guido van Rossum, WHO extended Python's syntax (in explicit the compartmentalization syntax) to make array computing easier". "An implementation of a matrix package was completed by Jim discoverer, then generalized[further rationalization required by Jim Hugunin and known as Numeric (also diversely observed because the "Numerical Python extensions" or "NumPy").Hugunin, a collegian at the Massachusetts Institute of Technology (MIT),joined the Corporation for National analysis Initiatives (CNRI) in 1997 to work on JPython, leaving Paul Dubois of Lawrence Livermore National Laboratory (LLNL) to need over as supporter. Other early contributors embrace David Ascher, Konrad Hinsen and Travis Oliphant".

"A new package known as Num array was written as a additional versatile replacement for Numeric. Like Numeric, it too is currently deprecated. Numarray had quicker operations for large arrays, however was slower than Numeric on tiny ones, thus for a time each packages were utilized in parallel for varied use cases. The last version of Numeric (v24.2) was discharged on St Martin's Day 2005, whereas the last version of numarray (v1.5.2) was discharged on twenty four August 2006”.

There was a want to urge Numeric into the Python customary library, however Guido van Rossum determined that the code wasn't reparable in its state then. “In early 2005, NumPy developer Travis Oliphant needed to unify the community around one array package and ported Numarray's options to Numeric, cathartic the result as NumPy one.0 in 2006.This new project was a region of SciPy. To avoid putting in the large SciPy package simply to urge associate array object, this new package was separated and known as NumPy. Support for Python three was other in 2011 with NumPy version one.5.0”. In 2011, PyPy started development on associate implementation of the NumPy API for PyPy. It is not nevertheless absolutely compatible with NumPy.

C. FACE RECOGNITION

The “Face recognition” library in python is a library which helps in recognizing and manipulating the faces by using the programming language python or from the command line with the simplest face recognition library after importing the module and accessing the required functions.The “Face recognition” library was built using dlib’s “state-of-the-art face recognition” and was further enhanced and built with deep learning. The model has an accuracy of 99.38%. It is used to find faces in pictures.

V. TRAINING AND TESTING

TRAINING IN OPENCV

In OpenCV, training refers to providing a recognizer algorithm with training data to learn from. The trainer uses the same algorithm (LBPH) to convert the images cells to histograms and then computes the values of all cells and by concatenating the histograms, feature vectors can be obtained. Images can be classified by processing with an ID attached. Input images are classified using the same process and compared with the dataset and distance is obtained. By setting up a threshold, it can be identified if it is a known or unknown face. Eigenface and Fisherface compute the dominant features of the whole training set while LBPH analyses them individually. To do so, firstly, a Dataset is created. You can either create your own dataset or start with one of the available face databases.

•Yale Face Database

•AT & T Face Database

The .xml or. Yml configuration file is made from the several features extracted from your dataset with the help of the FaceRecognizer Class and stored in the form of feature vectors.

TRAINING THE CLASSIFIERS

OpenCV enables the creation of XML files to store features extracted from datasets using the FaceRecognizer Class. The stored images are imported, converted to Grayscale and saved with IDs in two lists with same indexes. Face Recognizer obj ects are created using FaceRecognizer class. Each recognizer can take in parameters described below. cv2.face.createEigenFaceRecognizer()

train() FUNCTION
Trains a FaceRecognizer with given data and associated labels.

Parameters:

The training images, that means the faces you want to learn. The data has to be

given as a vector<Mat >. labels The labels corresponding to the images have to be

given either as a vector<int> or any other data type.

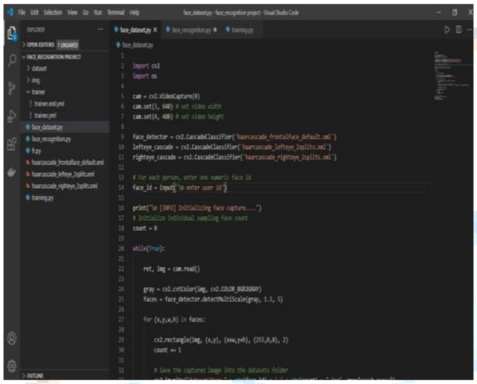
CODE
Given below is the code for creating a .yml file, that is the configuration model that stores features extracted from datasets using the FaceRecognizer Class. It is stored in

a folder named ‘recognizer’ under the name ‘training Data.yml’.

DATASET:

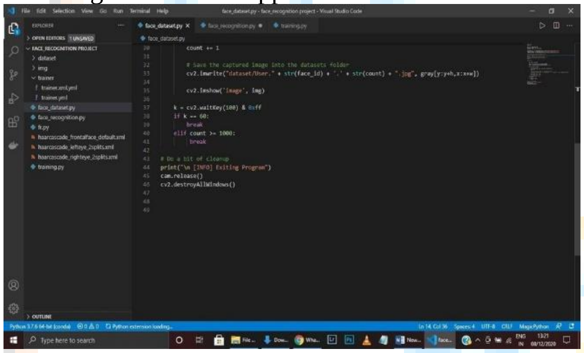
This is the code that will be used to create a dataset. It will turn the camera and take

number of pictures for few seconds. Given below is the code for face_dataset.py



```
import cv2
import os

cam = cv2.VideoCapture(0)
cam.set(3, 640) # set video width
cam.set(4, 480) # set video height

face_detector = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
lefteye_cascade = cv2.CascadeClassifier('haarcascade_lefteye_2splits.xml')
righteye_cascade = cv2.CascadeClassifier('haarcascade_righteye_2splits.xml')

# For each person, enter one numeric face id
face_id = input('\n enter user id')

print("\n [INFO] Initializing face capture....")
# Initialize individual sampling face count
count = 0

while(True):

    ret, img = cam.read()

    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    faces = face_detector.detectMultiScale(gray, 1.3, 5)

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

        cv2.rectangle(img, (x,y), (x+w,y+h), (255,0,0), 2)
        count += 1

        # Save the captured image into the datasets folder
```

Figure 1: Code snippet for the dataset

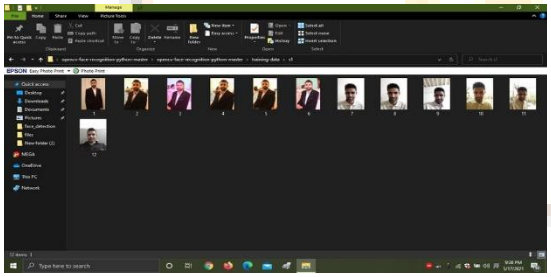


```
        count += 1

        # 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])

        cv2.imshow('image', img)

    k = cv2.waitKey(100) & 0xff
    if k == 60:
        break
    elif count >= 1000:
        break

# Do a bit of cleanup
print("\n [INFO] Exiting Program")
cam.release()
cv2.destroyAllWindows()
```

Figure 2: Face_dataset.py

OUTPUT

After running the dataset code we will get number of pictures in a folder named dataset. Now these photos will be used to train. The more the pics the greater the accuracy of the trainer.



VI. CONCLUSION

Facial Detection and Recognition systems are gaining a lot of popularity these days. Most of the flagship smartphones of major mobile phone manufacturing companies use face recognition as the means to provide access to the user.

This project report explains the implementation of face detection and face recognition using OpenCV with Python and also lays out the basic information that is needed to develop a face detection and face recognition software. The goal of increasing the accuracy of this project will always remain constant and new configurations and different algorithms will be tested to obtain better results. In this project, the approach we used was that of Local Binary Pattern Histograms that are a part of the FaceRecognizer Class of OpenCV

REFERENCES

[1] Schneiderman.United States of America Patent U.S. Patent No. 8,457,367, 2013.AL-Hussami, M. (2008).

[2] R. J. Baron, “Mechanisms of human facial recognition,” International Journal of Man-Machine Studies.

[3] M. Nixon, “"Eye Spacing Measurement for Facial Recognition",” International Society for Optics and Photonics., vol. (Vol. 575), (19 December 1985).

[4] H. &. Y. J. Yu, “A direct LDA algorithm for high-dimensional data—with application to face recognition,” 2001.