CHANDIGARH UNIVERSITY
Discover. Learn. Empower.

Facial Recognition using python

Domain: Internet of Things

Project Synopsis

Team Members :

1.	Ram Ashish Yadav	(21BCS 1508)
2.	Arpit Kumar	(21BCS 1815)
3.	Chaitanya Vatsyayan	(21BCS 2414)
4.	Ayushi	(21BCS 2849)
5.	Shreya Pandey	(21BCS 8286)

Guided by : Ms. Samli

Teacher Name : Mr. Lokesh Pawar

Title of the project:

Face Recognition Using IOT and Python.

Introduction:

The system is based on security that combines the functions of smart phone and home network system. It enables the users to monitor visitors in real-time, remotely via the IoT based doorbell installed near the entrance door to a house. Face Recognition is a technology in computer vision. In Face recognition / detection we locate and visualize the human faces in any digital image.

It is a subdomain of Object Detection, where we try to observe the instance of semantic objects. These objects are of particular class such as animals, cars, humans, etc. Face Detection technology has importance in many fields like marketing and security.

This system makes security as further autonomous by capturing the images automatically and processing the image for facial matching and uses mail communication to the server to confirm the intruder is known or unknown.

Literature Review :

We first review the development of face recognition approaches, followed by a review of face modelling and model compression methods. Finally, we will present one major application of face recognition technology, namely, face retrieval.

We mainly pay attention to the methods that are employed in the task-specific cognition or whose behaviour is specified by humans (i.e., artificial intelligence pursuits), although there are developmental approaches for facial processing (e.g., autonomous mental development and incremental learning methods) that have emerged recently.

Open CV :

Open Source Computer Vision Library is an open-source computer vision and machine learning software library. which is built to provide a common infrastructure for machine learning algorithms and computer vision.

It has thousands of optimized algorithms which can be used different purposes like detecting and recognizing faces, identifying objects and many more. We need it to take pictures using our webcam and some manipulation needed to be done in the image.

NumPy:

NumPy is the fundamental package for scientific computing in Python which provides a multidimensional array object other mathematical operations can be performed using this but simply speaking we just need it to convert our images into some form of an array so that we can store the model that has been trained.

Haar Cascade:

Haar Cascade is basically a classifier which is used to detect the objects for which it has been trained for, from the source. The result is an XML file which stores the trained result. If said simply the Haar Cascade is trained by superimposing the positive image over a set of negative images.

The training requires a high spec system and a good internet connection and thousands of training images that is why it is carried out in the server. For increasing the efficiency of the results they use high-quality images and increase the number of stages for which the classifier is trained. We need Haar cascade frontal face recognizer to detect the face from our webcam.

OBJECTIVE:

The objectives of Face Recognition for Real-Time Applications are given below :

1. To enhance the Frame/sec for Face Recognition System, such that Recognition is done in Real Time.
2. Presently, work on 30frames/sec Our motto is to achieve higher frames/sec or high-Resolution frames/sec.
3. Face Detection: Locate faces and draw bounding boxes around faces and keep the coordinates of bounding boxes.
4. Face Alignments: Normalize the faces to be consistent with the training database.
5. Feature Extraction: Extract features of faces that will be used for training and recognition tasks.
6. Face Recognition: Matching of the face against one or more known faces in a prepared database.

Software Used:

Software Used :	Open-CV
Version :	3.1.0
Tool :	Eclipse C
Module :	Image Processing
Language :	Python 3.3+

Face Recognition System:

Face recognition has always been an interesting research area over the last few years. Scientists and researchers from different areas of psychophysical sciences and from other areas such as computer sciences have done their research works on it.

Psychologists and neuroscientists mainly deal with the human perception part of the topic, whereas engineers studying on machine recognition of human faces deal with the computational aspects of face recognition.

Although building facial recognition seems easy it is not as easy in the real world images that are being taken without any constraint. There are several challenges that are faced by the

Problem Statement:

Illumination: It changes the face appearance drastically, it is observed that the slight changes in lighting conditions cause a significant impact on its results.

Pose: Facial Recognition systems are highly sensitive to the pose, Which may result in faulty recognition or no recognition if the database is only trained on frontal face view.

Facial Expressions: Different expressions of the same individual are another significant factor that needs to be taken into account. Modern Recognizers can easily deal with it though.

Low Resolution: Training of recognizer must be done on a good resolution picture, otherwise the model will fail to extract features.

Aging: With increasing age, the human face features shape, lines, texture changes which are yet another challenge.

Future Scope :

Face Recognition is a way where we make a machine learn and recognize things and humans with you of programing and mathematical functions. Our aim is to develop a device in future that would be able to be installed in homes of people and they would be able to safeguard their things with this device/software.

It can be implemented anywhere and by creating device based on IOT and AIML this would be a breakthrough for us.

Conclusion:

In this deep learning project, we would develop a model for real-time human face recognition with python and Open CV. The way we design and code and would develop Face detection model, Using Haar-cascades for face detection worked extremely well even when subjects wore spectacles. Real time video speed was satisfactory as well devoid of noticeable frame lag. Considering all factors, LBPH combined with Haar-cascades can be implemented as a cost effective face recognition platform.

An example is a system to identify known troublemakers in a mall or a supermarket to provide the owner a warning to keep him alert or for automatic attendance taking in a class.

We would develop a software that would be useful on a basic level and to implement it in real world more accuracy would be needed and that would be attained by training the model over the course.

Reference:

1. Geeks for Geeks
2. Great Learning
3. Towards data science
4. Ahonen, T., Hadid, A., and Pietikainen, M. Face Recognition with Local Binary Patterns. Computer Vision - ECCV 2004 (2004), 469–481.
5. A.K. Jain, S. J. R. Small sample size effects in statistical pattern recognition: Recommendations for practitioners. IEEE Transactions on Pattern Analysis and Machine Intelligence 13, 3 (1991), 252–264.
6. Belhumeur, P. N., Hespanha, J., and Kriegman, D. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. IEEE Transactions on Pattern Analysis and Machine Intelligence 19, 7 (1997), 711–720.
7. Brunelli, R., and Poggio, T. Face recognition through geometrical features. In European Conference on Computer Vision (ECCV) (1992), pp. 792–800.
8. Cardinaux, F., Sanderson, C., and Bengio, S. User authentication via adapted statistical models of face images. IEEE Transactions on Signal Processing 54 (January 2006), 361–373.
9. Chiara Turati, Viola Macchi Cassia, F. S., and Leo, I. Newborns face recognition: Role of inner and outer facial features. Child Development 77, 2 (2006), 297–311.
10. http://www.opencv.org
11. http://www.bytefish.de/blog
12. http://www.github.com/bytefish