International Journal of Innovative Research in Science and Engineering
Volume 7, Issue 07, 2021
www.ijirse.com
ISSN: 2454-9665

Face Recognition Attendance System Using OpenCV

Jyoti Maurya[1], Shivani Kumari[2], Satya Tiwari[3],

Pooja Maurya[4], Dr. Sudhir Agrawal[5]

[1234]*Student, IT Department, Buddha Institute of Technology,*

[5]*Asst. Professor, IT Department, Buddha Institute of Technology*

ABSTRACT

The face is the important part of the human body that uniquely identifies a person. Using the face characteristics as biometric, the face recognition system can be implemented. The most difficult task in any organization is attendance marking. In a traditional system, the students are called out by the teachers, and their presence or absence is marked accordingly. However, these traditional techniques take more time. In this project, the OpenCV-based face recognition approach has been proposed. This model integrates a camera that captures an input image, an algorithm for detecting a face from an input image, encoding and identifying the face, and marks the attendance. The training database is created by training the system with the faces of the authorized students. The cropped images are then stored as a database with respective labels. The features are extracted using the LBPH algorithm.

Keywords- *OpenCV, camera, face recognition, LBPH, attendance*

INTRODUCTION

Nowadays, technology is increasing and face recognition becomes one of the popular methods for biometrics. Face detection is a computer technology that determines the locations and sizes of human faces in digital images. It detects facial features and ignores anything else, such as tables, trees and buildings. It is a critical application in image analysis yet it is very challenging to create an automated system based on facial recognition. A system with ability to recognize human face accurately. One application of facial recognition is in the field of the attendance management system. The manual attendance system is time-consuming, thus many research has been conducted with the automatic or smart attendance management system to resolve this issue. One solution is the application of biometric attendance management system. However, it is difficult to verify each student in the classroom as there are many students who attend the class, and if the system cannot detect or recognize one student, it will interrupt the learning process.

International Journal of Innovative Research in Science and Engineering
Volume 7, Issue 07, 2021
www.ijirse.com
ISSN: 2454-9665

PROPOSED MODEL

The process involved in face recognition are:

1. Capture

2. Extraction

3. Comparison

4. Matching

The operation in each process is: In step one the capture is the way to snap the picture during the enrolment of the system. Then in the Face Recognition step, extraction is used for finding or extract the specific feature from the face. The third step is comparison, where new input is used for comparison with the database (sample data). Finally, the last step is matching: the system will try to find the matching of the new face with the registered face based on extraction and comparison process.

METHODOLOGY

Below are the methodology and descriptions of the applications used for data gathering, face detection, training and face recognition. The project was coded in Python using a mixture of IDLE and PyCharm IDEs.

a. Face Detection:

First stage was creating a face detection system using Haar-cascades. Although, training is required for creating new Haar-cascades, OpenCV has a robust set of Haar-cascades that was used for the project. Using face-cascades alone caused random objects to be identified and eye cascades were incorporated to obtain stable face detection. The flowchart of the detection system can be seen in figure.

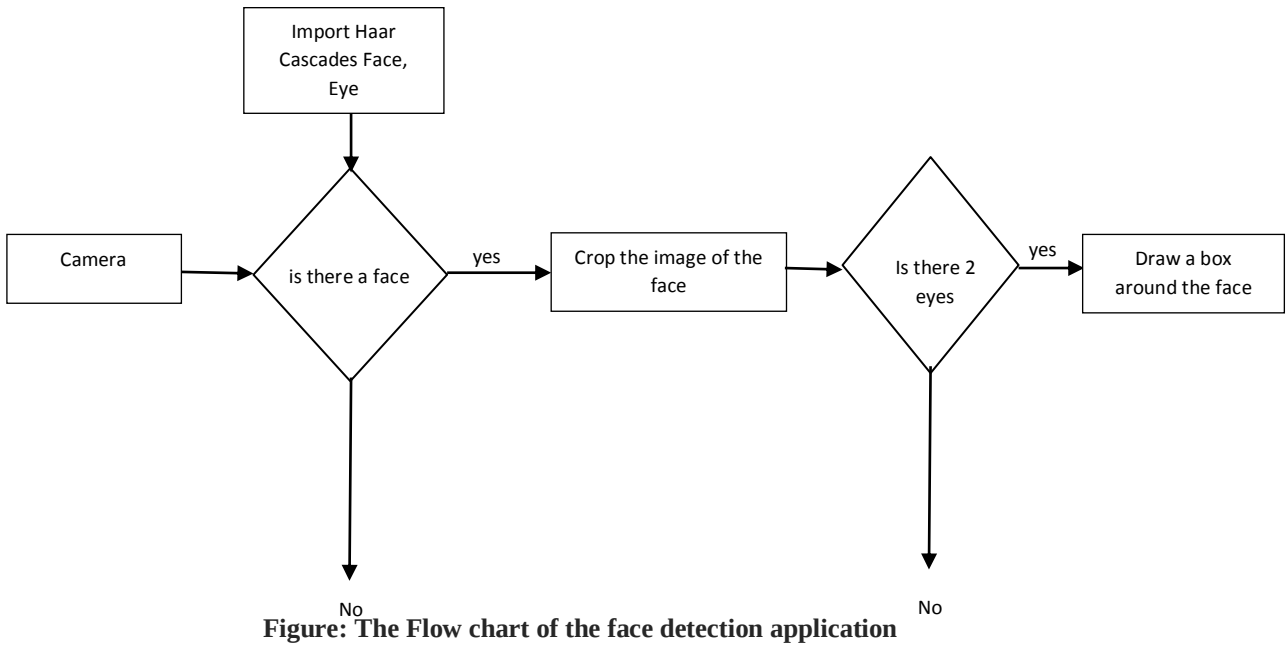
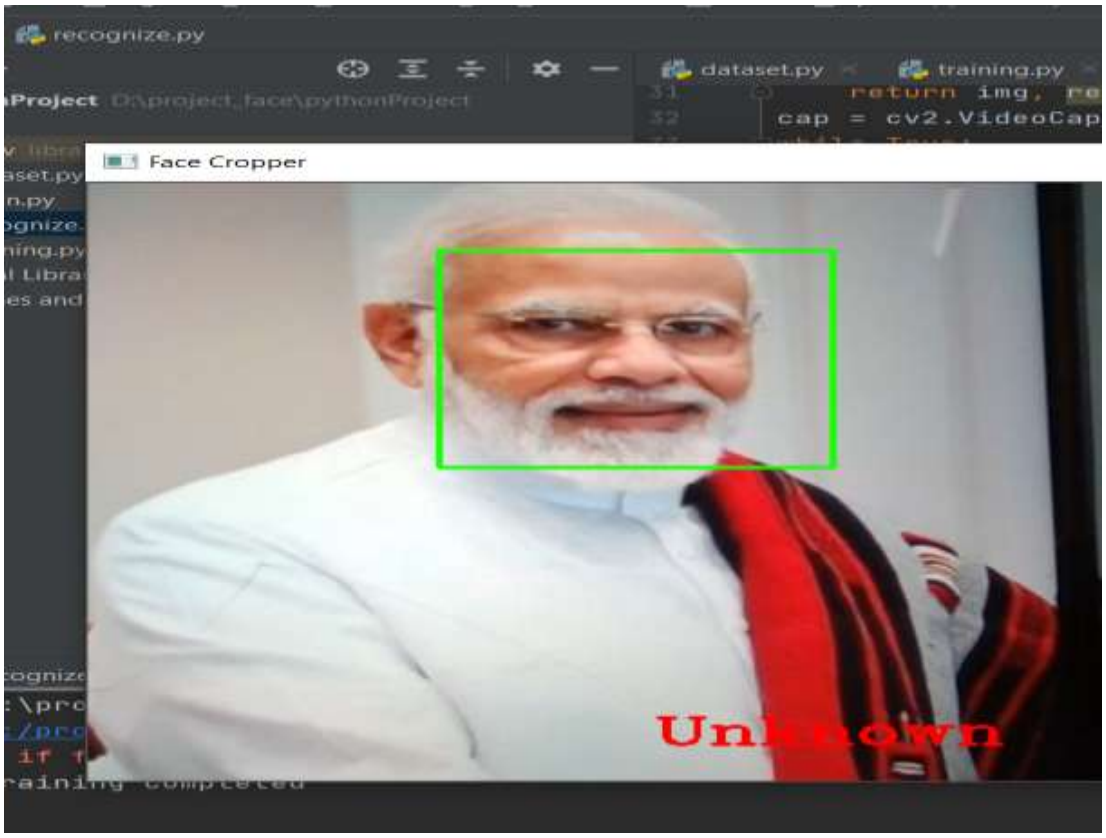


Figure: The Flow chart of the face detection application

International Journal of Innovative Research in Science and Engineering
Volume 7, Issue 07, 2021
www.ijirse.com
ISSN: 2454-9665

Classifier objects are created using classifier class in OpenCV through the cv2.CascadeClassifier() and loading the respective XML files. A camera object is created using the cv2.VideoCapture() to capture images. By using the CascadeClassifier.detectMultiScale() object of various sizes are matched and location is returned. Using the location data, the face is cropped for further verification. Eye cascade is used to verify there are two eyes in the cropped face. If satisfied a marker is placed around the face to illustrate a face is detected in the location.



b. Collecting the image data:

Collecting classification images is usually done manually using a photo editing software to crop and resize photos. Furthermore, PCA and LDA requires the same number of pixels in all the images for the correct operation. This time consuming and a laborious task is automated through an application to collect 50 images with different expressions. The application detects suitable expressions between 300ms, straightens any existing tilt and save them. The Flow chart for the application is shown in figure below.

International Journal of Innovative Research in Science and Engineering
Volume 7, Issue 07, 2021
www.ijirse.com
ISSN: 2454-9665

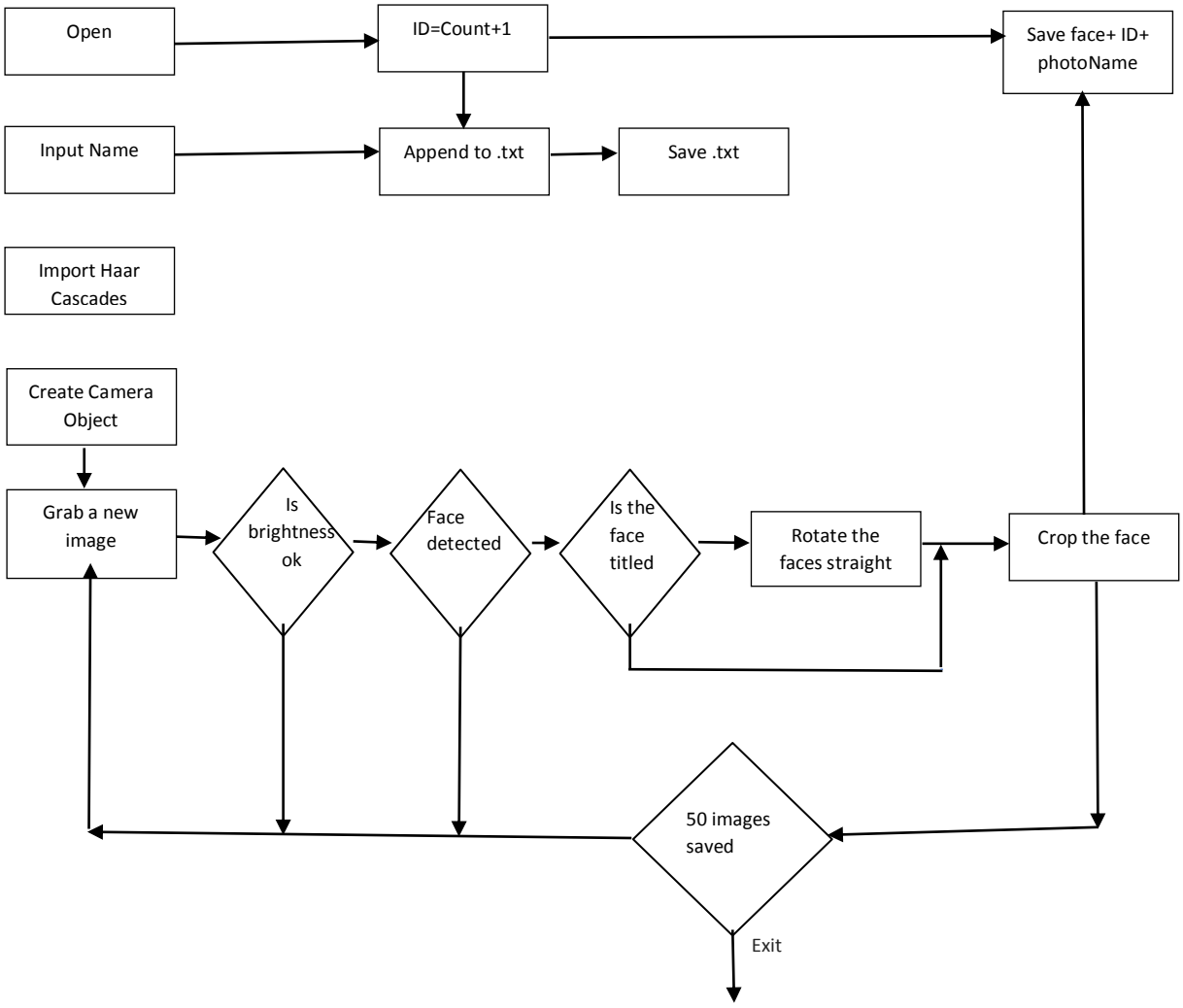


Figure: The Flowchart for the image collection

Application starts with a request for a name to be entered to be stored with the ID in a text file. The face detection system starts the first half. However, before the capturing begins, the application check for the brightness levels and will capture only if the face is well illuminated. Furthermore, after the face is detected, the positions of the eyes are analysed. If the head is tilted, the application automatically corrects the orientation. These two additions were made considering the requirements for Eigenface algorithm. The Image is then cropped and saved using the ID as a filename to be identified later. A loop runs this program until 50 viable images are collected from the person. This application made data collection efficient.

c. Training the Classifiers:

OpenCV enables the creation of XML files to store features extracted from datasets using the Face Recognizer class. The stored images are imported, converted to gray scale and saved with IDs in two lists with same

indexes. Face Recognizer objects are created using face recogniser class. Each recogniser can take in parameters that are described below:

cv2.face.createLBPHFaceRecognizer()

1. The radius from the centre pixel to build the local binary pattern.
2. The Number of sample points to build the pattern. Having a considerable number will slow down the computer.
3. The Number of Cells to be created in X axis.
4. The number of cells to be created in Y axis.
5. A threshold value similar to Eigenface and Fisherface. if the threshold is passed the object will return -1Recogniser objects are created and images are imported, resized, converted into numpy arrays and stored in a vector. The ID of the image is gathered from splitting the file name, and stored in another vector.

By using FaceRecognizer.train(NumpyImage, ID) all three of the objects are trained. It must be noted that resizing the images were required only for Eigenface and Fisherface, not for LBPH. Next, the configuration model is saved as a XML file using FaceRecognizer.save(FileName). In this project, all three are trained and saved through one application for convenience.

d. The Face Recognition:

Face recogniser object is created using the desired parameters. Face detector is used to detect faces in the image, cropped and transferred to be recognised. This is done using the same technique used for the image capture application. For each face detected, a prediction is made using FaceRecognizer.predict() which return the ID of the class and confidence. The process is same for all algorithms and if the confidence his higher than the set threshold, ID is -1. Finally, names from the text le with IDs are used to display the name and confidence on the screen. If the ID is -1, the application will print unknown face without the confidence level.

RESULTS:

The collected images are shown below. Each face has 50 images. Three applications were written to iterate through the parameters of each algorithm. On each iteration, the algorithm is trained using different parameters and tested against a photo. The resulting data is plotted at the after finishing the tests. The applications are :

TestDataCollector LBPH.py.

CONCLUSION:

This paper describes the project for visual perception and autonomy module. Next, it explains the technologies used in the project and the methodology used. Finally, it shows the results, discuss the challenges and how they were resolved followed by a discussion. 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.

ISSN: 2454-9665

7. REFERENCES:

[1] Takeo Kanade. Computer recognition of human faces, volume 47. Birkhauser Basel, 1977.

[2] Lawrence Sirovich and Michael Kirby. Low-dimensional procedure for the characterization of human faces. Josa a, 4(3):519-524, 1987.

[3] M. Turk and A. Pentland. Eigenfaces for recognition. Journal of Cognitive Neuroscience, 3(1):71-86, Jan 1991.

[4] Dong chen He and LiWang. Texture unit, texture spectrum, and texture analysis. IEEE Transactions on Geoscience and Remote Sensing, 28(4):509{512, Jul 1990.

[5] X. Wang, T. X. Han, and S. Yan. An hog-lbp human detector with partial occlusion handling. In 2009 IEEE 12th International Conference on Computer Vision, pages 32-39, Sept 2009.

[6]. S. Chintalapati and M.V.Raghunandh, "Automated attendance management system based on face recognition algorithms."IEEE Int. Conference and Computational Intelligence and Computing Research, 2013.

[7]. Kavita , Ms. Manjeet Kaur," A Survey paper for Face Recognition Technologies", International Journal of Scientific and Research Publications, Volume 6, Issue 7, July 2016

[8]. K Goyal, K Agarwal, R Kumar, "Face Detection and Tracking", International Conferences on Information Technology, Information Systems and Electrical Engineering(ICITISEE), 2017.

[9]. T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with Local Binary Patterns." IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 12,pp

[10]. Refik Samet and Muhammad Tanrivedi. "Face Recognition-Based Mobile Automatic Classroom Attendance Management System", IEEE Int. International Conference on Cyberwords, 2017.