2020 International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC)
GLA University, Mathura, UP, India. Feb 28-29, 2020

2020 International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC) 978-1-7281-6575-2/20/$31.00 ©2020 IEEE 10.1109/PARC49193.2020.236628

Real Time Attendance System Using Face Recognition Technique

Mayank Srivastava
Department of CEA,
GLA University
Mathura (UP), India
mayank.srivastava@gla.ac.in

Amit Kumar
Department of CEA,
GLA University
Mathura (UP), India
amit.kumar_cs16@gla.ac.in

Aditya Dixit
Department of CEA,
GLA University
Mathura (UP), India
aditya.dixit_cs16@gla.ac.in

Aman Kumar
Department of CEA,
GLA University
Mathura (UP), India
aman.kumar_cs16@gla.ac.in

***Abstract*—Authentication plays a very important role in computer based communication for having system control. Face recognition has become a very crucial part of biometric verification and finds uses in many applications, e.g. video monitor system, human-computer interaction, network security. This paper is about integrating face recognition technology with open source computer vision (OpenCV) algorithm and develop a Attendance System. This software will facilitate the attendance automaton process and enable faculties to enquire student's data by just maintaining a log for clock-in and clock-out time.**

Index Term — Authentication, OpenCv Haar Cascades Algorithm, Face recognition, Attendance automaton.

I. INTRODUCTION

Face recognition has been in practice since computer vision emerged due to both, practical significance and theoretical likes of cognitive scientists. Even though Identification other methods such as biometrics have higher accuracy, researchers always took a great interest in face recognition due to its non-invasiveness nature and primary person identification methods of humans. This method is becoming very popular globally as a biometric solution because there involves zero user effort in this method compared to other technologies. The three most important applications of face recognition are: visitor management system; employee management and time attendance systems; authorization systems and access control systems.

The traditional method of taking attendance is to manually distribute attendance sheet to the students in the class taking a lot of time. Furthermore, it is very difficult to get perfectly accurate attendance by this technique in large classrooms.

This paper describes how face recognition technology can be used as a better method to record attendance by automatically recognizing the person. The Proposed system makes a log file too for keeping record of every entrance synced with a universal system time.

Scientists first used facial recognition in the semi-automatic system in the 1960s. Key features were located by marking the photograph; In this the eyes, ears, nose and mouth were used as features. Ratios and distances were then calculated from the points at a common reference point and compared to the reference data. A system was developed by Goldstein, Hermann, and Lesak [2] in the early 1970s which contained 21 subjective markers, e.g. Hair color. This system did not make the process of automaton easier but proved more problematic because measurements were performed by hand. Fischer and Elsleherb [3] did this differently. Face's different pieces were measured and then compared with respect to one global template, producing a data not unique enough for representing an adult face. The connectionist approach [4] employs the combination of gestures and markers to classify human face and the implementation of this requires neural net principle and 2-d pattern recognition. Mostly, this approach requires a large number of training faces to achieve respectable accuracy; Therefore, it has not received widespread acceptance. Irawan et al. [11] also focus on novel IOT based monitoring system. The first fully automated system [5] used very general pattern recognition. It explicitly compared a general facial model of expected features and developed an order of image comparison pattern for this model. This approach is a statistical approach and relies on histogram and gray scale values.

The authors are with the Computer Science and Engineering Department, GLA University Mathura, India (e-mail: Mayank.srivastava@gla.ac.in, amit.kumar_cs16@gla.ac.in, aditya.dixit_cs16@gla.in, aman.kumar_cs16@gla.ac.in).

II. SYSTEM OVERVIEW

The current authors used the antigen approach to facial recognition developed by Kirby and Syrovich at Brown University in 1988. This method analyzes facial images and calculates eigenface [8] which are faces consisting of eigenvectors. A comparison of the eigenface is used to detect the presence of a face and its identity. Turk and Pentland [1] developed a method of face recognition which performed a five sequential procedures to recognize. The first step is to load some training images in the system. This step helps in defining face space which means to have a set of images that appear like face. Then the next step requires calculation of eigenface for a given face. When compared with known faces and using a bit of statistical analysis it can be known whether the image presented is exactly one face. Now, if system declares an image as face, it checks its identity. If an unknown face keeps coming to the system, the system learns to recognize its identity and gives result.

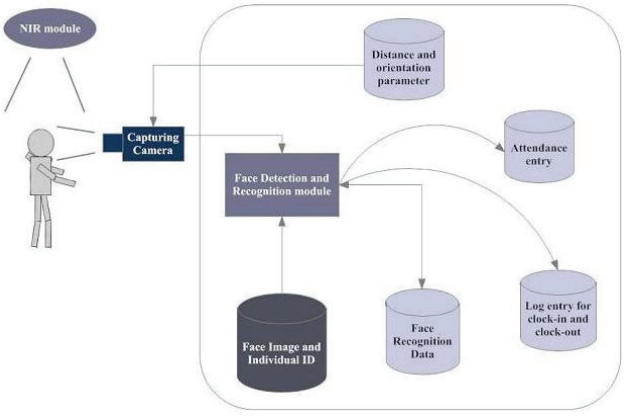

Fig. 1. Architecture of the system

A. *Open Source (Computer Vision Library)*

OpenCV and the Light Tool Kit (FLRK) play most important role in the implementation of the system. With the help of OpenCV we can develop simple-to-use computer vision which facilitates development of effective vision applications. Vision related 500 different functions are part of OpenCV library. OpenCV is the most crucial technological player behind the concept of face recognition. FLTK facilitates development of interface. The user places a minimum distance of 50 cm in front of the camera and its image is captured as input. Frontal faces are extracted from the image and then transformed into gray scale and stored. An xml file contains eigen values which are obtained by applying the Principal Component Analysis (PCA) algorithm [7] on the diagrams. Now, when a frontal face is obtained from a video, the system calculates eigen values and matches with nearest neighbor.Haar Cascades

Paul Viola and Michael Jones developed a method HaarCascade that identifies objects in a video and a picture. They published this in "Rapid Object Detection in Boosted Cascades of Simple Features".

The following steps summarize the process.

1) The training data sets are {D1, D2, D3,..DM } .The average Avg can be calculated by the formulae:

$$Avg = \frac{1}{M}\sum_{j=1}^{M} D_j$$

2) The element of training data sets differs from Avg by the vector Yj=Dj-Avg. For computing (covariance matrix) Cov given formula is used:

$$Cov = \frac{1}{M}\sum_{j=1}^{M} Y_j . Y_j^T$$

3) M important eigen vectors of Cov are chosen as EK's, and for every element in the training data set weight vector.

$$W_{jk} = E_K^T . (D_j - Avg), \forall j, k$$

Wjk is calculated where k varies from 1 to M by the formulae.

III. SYSTEM IMPLEMENTATION

The three steps that are involved in the system:

1) The first step performs the task of detection and extraction of face image which is stored in an xml file and can be used in future.

2) The second step includes the learning and train ing of face image and thus computing eigen vector and eigen value of image.

3) The final step is to recognize by comparing face images stored in xml file.

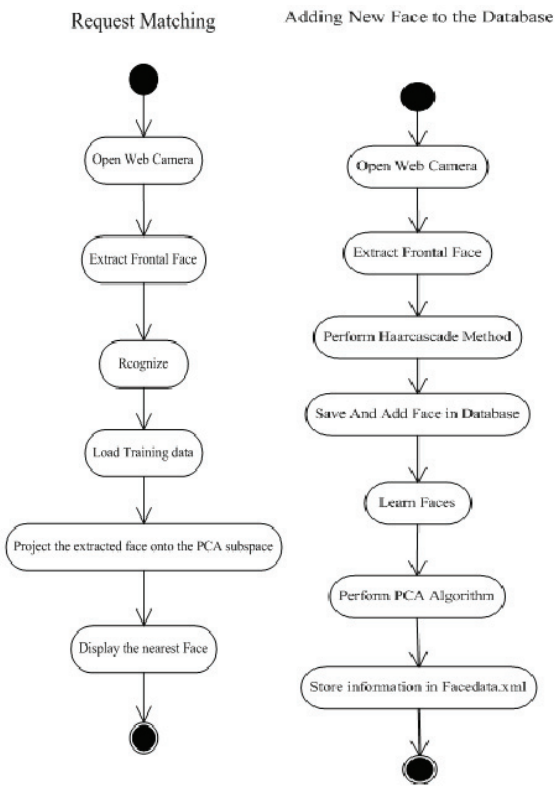

Fig. 2. Flowchart of the Proposed Model

A. Face Detection and Extract

First of all, openCAM_CB() function call is done which captures photo by turning on camera. ExtractFace() fiunction call obtains frontal face from videos by using OpenCv HaarCascade technique and loads the haarcascade_ frontalface_alt_tree.xml as the classifier for the process. The output of "1" and "0" are given by classifier. "1" if the region shows the object (i.e., face), and "0" if it does not. If someone wants to search an object out of image, one needs to perform checking of every location with the help of classifier. This requires moving of search window. The classifier is very flexible and can easily be used for different sizes by "resizing" when possibility of getting objects at other sizes exist. By scanning at different scales a previously unknown size object in the image can be found out. a gray scale image of 50x50 pixels is obtained after face detection process.

B. Learn and Train Face Images

Learn() function is the function that performs the task of applying algorithms on the training set. The implementation of above function requires following four steps:

1 Load the training data.

2 Do Haar Cascade on it to find a subspace.

3 Project the training faces.

4 Store all the training information.

   a. Eigenvalues

   b. Eigenvectors

   c. The average training face image

   d. Projected face image

   e. Person ID numbers

C. The built-in Open CV function is called to perform the Hara Cascade, cvCalcEigen object (), which calculates subspace. Then an output variable is created by rest doPCA() that stores outcomes on returning of cvCalcEigenObjects () [5].

D. If we have to perform Haar Cascade, then we must make sure that the dataset is "centered". The meaning for facial images that dataset should be "centered" means that the average image – a type of image where every pixel contains the average value of that pixel of all facial images that are part of training set. Then, the process of dataset centering is done by deducting the values of pixels of average face form every training image. This process is performed under cvCalcEigenObjects ().

E. But we require us to get a grip on the average image, because later it will be necessary to project the data for that motive, and therefore assignment of memory for average image, which is a floating point image is compulsory. This step leads us to getting a subspace using algorithm of HaaRCascade and this subspace will contain training images converted into points. This process is named as "projecting" of training image. And the function of OpenCv that is used for this task is cvEigenDecomposite (). Finally, the built-in persistence functions of the OpenCV store learned face representation data using an XML file. Recognize and Identification

Recognize() function: The function which is implemented for the purpose of validation of the program by the Eigenface. The two stages of total three stages of the process – facial images loading and projecting theses to subspace have already been discussed. When called LoadFaceImgArray(), face images are loaded, which are listed on the training data set. The storage of ground truth of user identification number is done in IMGARR and in personNumTruthMat. The local variable N Testing faces stores the total count of face images. And the next thing that we are required to do is loading of global variable N trainedfaces, pAvgTraining, EigenVectArray and nEigens. FunctionloadTrainingDataSet () is the function which does this task for us. Finally, location and loading of all of data value is done in an XML file by OpenCV. Then comes the last step of the validation phase and which is projecting of all of test image on the Hc subspace and then locating the closest of the training images. Projection of test image by CvEigenDecomposite () can be considered as the same thing as the learning() function's face-projection code.

The like we did previously, we again need to perform the passing of these through a lot of eigen values. (nEigens), and array of eigenvectors (eigenVectArray). This time, however, as the first argument, the test image is passed and not an image that has been trained. The storage of output from the function of CvEigenDecomposite() is done by a local variable ProjectedTestes. Since there is no need to store the approximate test image, we used the C array for ProjectedTestFace instead of using the OpenDrive matrix.

The FindNearestNeighbour() function calculate the distance for each estimated training instance from the projected project image. Here the basis of distance is "squared Euclidean distance." To calculate the Euclidean distance of two points, we must add the square distance to each dimension, and then find root of that sum. Here, we do sum, but out of finding of square root phase. The end outcome is perfectly identical, since the least spaced neighbour also has least squares distance, we can reduce the time of calculations of these steps by going for comparison of these squared values.We can also reduce the complexity of the problem of least square.

IV. EXPERIMENT AND RESULT

The following steps describe the how experiments are performed:

A. *Face recognition:*

Start capturing images via a camera module placed in the bus: Get Started:

// Pre-capture the captured image and extract the face image.

// the first step is to perform calculation of eigen values of the captured face image and compare it with the eigen values of the existing faces in the training data Sets of college database.

// If the eigen value does not match the existing one, the storage of the information of new face image in the face database (xml file) should be done.

// If the eigen value matches with the existing one then the validation step will occur and it will give the notification in app.

End;

B. *Face recognition:*

These steps will be performed for face recognition using the defeat cascade algorithm:

Start:

// Get the facial information of the matching face image from the database.

// Then updating of log table with the corresponding facial image and system time that completes attendance for an individual student should be done.

End;

This section present the results of experiments conducted to capture faces in a gray scale image of 50x50 pixels.

TABLE I. DESCRIBE THE OPENCV FUNCTION USE IN THE PROPOSED SYSTEM AND ITS EXECUTION RESULTS.

Test data	Expected Result	Observed Result	Pass/ Fail
OpenCAM_CB()	Connects with the installed camera and starts playing.	Camera started.	pass
LoadHaar Classifier()	Loads the HaarClassifier Cascade files for frontal face	Gets ready for Extraction.	Pass
ExtractFace()	Initiates the Paul-Viola Face extracting Frame work.	Face extracted	Pass
Learn()	Start the Haar Cascade Algorithm	Updates the facedata. xml	Pass
Recognize()	It compares the input face with the trained faces.	Nearest face	Pass

Fig. 3. Training Images

TABLE II. FACE DETECTION AND RECOGNITION RATE

Face Orientations	Detection Rate	Recognition Rate
O^0 (Frontal face)	98.7 %	95%
18º	80.0 %	78%
54º	58.6 %	68%
72º	0.00 %	0.00%
90º(Profile face)	0.00 %	0.00%

We experimented with 30 faces as a training set of 7 people for measurement of accuracy of the system. The Extract () function shows a sample binary image in fig. III obtained with the help of face extracting frame work detection method by Paul – Viola.

The results in Table 2 clearly shows that with respect to face detection and recognition rate, on increasing the face angle, camera decreases.

Introducing entry and exit times, the authors intend to develop an attendance management system for colleges which is based on facial recognition technology. Every student's attendance is collected by the system through constant observation at the entry and exit points. The results of our initial experiment performed better in performance assessment than traditional black and white display systems. This system is mainly developed for face recognition from images or video frames.

We always intend to make our systems better and accurate and thus we hope that with the interaction between users and system, the accuracy of system improves. The best part about our system is its profound importance in mobile.Face Recognition application and that can be used by general public for enquiring the photo of any person captured by a mobile phone camera by the help of proper authorization facilitated by database.

Our System ensures that only permitted persons can travel. And functionalities such as unknown person notification on the app, mark the valid person attendance are also provided.

REFERENCES

[1] Face Detection and Recognition using OpenCV, Article, http://shervinemami.info/faceRecognitio n.html, Published by Shervin Emami, 2010

[2] Seeing with OpenCV, Article, http://www.cognotics.com/opencv/servo2007_series/part_1/index.html, Published by Robin Hewitt, 2010

[3] OpenCV Homepage, http://opencv.willowgarage.com

[4] FacebRecognition Homepage, http://www.face-rec.org/algorithms/

[5] Wikipedia, Three-dimensional face recognition, http://en.wikipedia.org /wiki /Threedimensional_face_recognition

[6] Wikipedia, Active appearance model, http://en.wikipedia.org/wiki/Active_appe arance_model

[7] Computer Vision Papers, http://www.cvpapers.com/

[8] Kamesh V, Karthick M, Kavin K, Velusamy M and Vidhya R, "Real-Time Fraud Anomaly Detection in E- banking Using Data Mining Algorithm", South Asian Journal of Engineering and Technology,Vol.8,supplementary issue.1,PP.144-148, April 6, 2019.

[9] Vijayakumar M and Prabhakar E, "A Hybrid Combined Under - Over Sampling Method for Class Imbalanced Datasets," International Journal of Research and Advanced Development (IJRAD), Volume 02, Issue 05, pp. 27 - 33, December 2018.

[10] V S. Sureshkumar, D. Joseph Paul, N.Arunagiri, T.Bhuvaneshwaran ,S,Gopalakrishnan "Optimal Perfor mance And Security Of Data Through FS- Drops Methodology," ,International Journal of Innovative Research In Engineering Science and Technology , pp:1-7, Issue 3,volume5, 2017

[11] J.D. Irawan, E. Adriantantri and A. Farid, "RFID and IOT for attendance management system," MATEC Web of Conferences (ICESTI), volume 164, 01020, 2018.