Implementation of Automated Attendance System using Face Recognition.

Mathana Gopala Krishnan, Balaji, Shyam Babu
Guided by: Mr.K.Rajesh AP-II CSE & Supported by: Dr.A.Uma makeswari AD, CSE

Abstract— Authentication is an issue in computer based communication.Face recognition is widely used in many applications such as system security and door control system.The paper describes how to take student's attendance using face recognition.The face recognition is implemented with the help of Principal Component Analysis (PCA) algorithm.The system will recognize the face of the student and saves the response in database automatically.The system also includes the feature of retrieving the list of students who are absent in a particular day.

Index Terms— Absent list, Authentication, Automatic, Database, Face Recognition, PCA

1 INTRODUCTION

Face recognition is one of the few biometric methods that possess the merits of both accuracy and low intrusiveness.For this reason since the early 70's, face recognition has drawn the attention of researchers in fields from security and image processing to computer vision.Face recognition has also proven useful in multimedia information processing areas.

Traditionally,attendances are taken manually in the class room using attendance registers given to the faculty members.But it is a time consuming event.Also,it is very difficult to verify students one by one in a large classroom environment,whether they are present or not.

The proposed system demonstrates how face recognition is used for taking attendance of a student automatically using PCA algorithm, how to store the faces in the database and how to retrieve the absent list.

1.1 Evolution

The first attempt to use face recognition was made in 1960's with semi-automated system.The photographs contained marks to locate major features.It used features such as mouth, eyes and nose.But the problem was to select from the database a small set of records such that one of the image records matched the photograph.

Another approach which seeks to classify the human face using combination of gestures and identifying markers [10]. But the problem is that this approach requires a huge number of training faces to achieve decent accuracy.

Fisher's approach [6] was to measure different pieces of the face which are then mapped onto a global template.But the problem is that those features do not contain enough unique data to represent an adult face.

The first fully automated system [12] developed uses very general pattern recognition.It compared faces to a generic face model and created a patterns.But this approach is statistical and relies on histograms and gray scale values.

2 SYSTEM OVERVIEW

The system uses the eigenface approach for face recognition. The method analyzes and computes eigenfaces [4] which are faces composed of eigenvectors.The method also compares the eigenfaces to identify the presence of a person(face) and its identity.The method involves the following steps [1].As a first step the system should be initialized with a set of training faces.Next,when a face is detected the eigenface is calculated for that face.Then, the system compares the eigenvectors of the current face and the stored face image and determines whether the face is identified or not.The final step(optional) is that if the unknown face is detected repeatedly the system may learn to recognize it.

2.1 Components

The main component used in the system are open source computer vision library (OpenCV) and Microsoft Visual Studio 2010 Professional Edition.OpenCV is a library of programming functions aimed mainly at real-time computer vision developed by intel Russia research center.The library is cross platform.The OpenCV's application areas includes 2D and 3D feature toolkits,facial recognition system,object identification and motion tracking.OpenCV library contains more than 550 functions.The interface for OpenCV used here is Microsoft Visual Studio 2010 Professional Edition.

2.2 Process

The student needs to be in front of a camera at a minimum distance of 60cm.The system will detect the image of the student according to PCA [13], converts it into a gray scale and stores it in an xml file.When the student reappears before the camera, faces are recognized by comparing the eigenfaces of current and strored images.Then the names of the detected faces are stored in Microsoft Access Database.

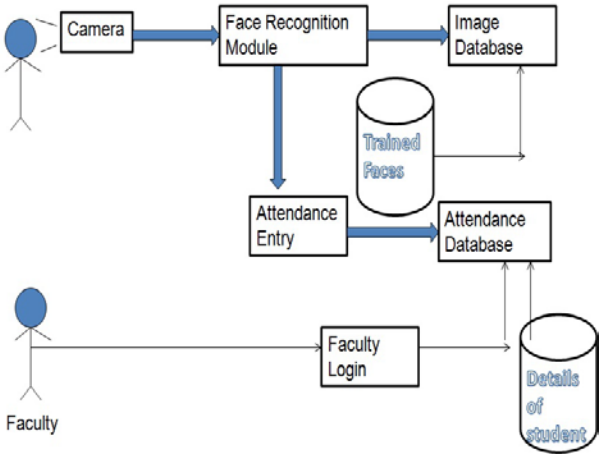

Fig. 1. Overview of the system.

2.3 Algorithm

The algorithm used in the system is Principal Component Analysis (PCA). Applications such as face recognition and image compression uses PCA algorithm.It is also used for finding data patterns [5].The following steps shows the process of PCA.

1. Mean center the data(optional)
2. Compute the covariance matrix of the dimensions.
   $C = A^T A$ (1)
3. Find eigenvectors (V, E) of covariance matrix C.
4. Sort eigenvectors in decreasing order of eigenvalues.
5. Project onto eigenvectors in order.

3 System Implementation

Three basic steps are used for implementing the proposed system.

1. Detect and extract the face image and save the details in an xml file.
2. Calculate eigenvalue and eigenvector for that image.
3. Recognize the face and match it according to eigenvalues and eigenvectors stored in xml file [1].
4. Store the name of the face displayed in Microsoft Access Database.

3.1 Face detection and extraction

The function openCAM_CB() is called for starting the camera to capture the image.Next, ExtractFace() is used to extract the frontal face [2] in a vedio frame.The ExtractFace() usesOpenCV haarcascade method to load the face.xml(haarcascade file) as the classifier.The output of the classifier is in binary form and outputs "1" if face is found and "0" otherwise.After the face is detected it is clipped into a grayscale image of 50x50 pixels which is done by "Add Face" button in the face recognition module.

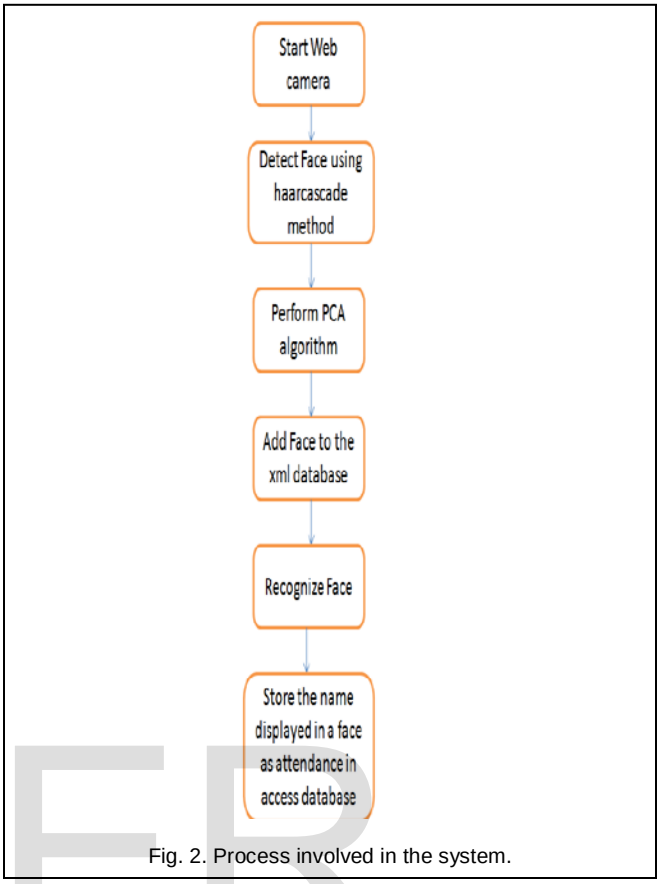

Fig. 2. Process involved in the system.

3.2 Learning and Training Face Images

The function Learn() performs the PCA algorithm on training datasets. The Learn() implementation involves four steps.

1. Load the training data.
2. Find a subspace by doing PCA on training data.
3. Project the training faces onto the PCA subspace.
4. Save the training information such as
   a. Eigenvalues
   b. Eigenvectors
   c. Average training face image
   d. Projected face image
   e. Person ID numbers

The function cvCalcEigen Objects() is an inbuilt function in OpenCV and it is called for performing PCA subspace calculation.To do PCA the dataset must be centred.Interms of face images this means finding the average image which is nothing but an image in which each pixel contains the average value for that pixel across all face images in the training set.The data set can be centred by subtracting the average pixel values of faces from each training image.This is what is happening inside cvCalcEigen Objects() [12].

Now we have calculated subspace for PCA using cvCalcEigen Objects(). We can convert the training images into points in this subspace.This process is called "projecting" the image.

This is done with the help of OpenCV function called cvEigenDecomposite().Then the data for learned faces is stored in an xml file using OpenCV's built in function.

3.3 Recognization and Identification

In OpenCV,we have a function called recognize() [12],which will implement the recognition of the eigenfaces.It has three steps in which two of them are already done that is loading the face image and projecting onto the subspace.The function loadFaceImgArray(),loads the face image into faceImgArray,as listed in xml file.Here,the number of face images is stored in a separate textbox named"No.of faces in the scene"and the number of faces, is automatically counted according to the number of faces detected.

The global variables such as noEigens, trainingimageAvg and EigenVectorArray should be loaded.OpenCV locates and loads each data value in xml file by the name given.

The final step in the recognition face is projectiong each test image onto PCA subspace and locating the closest projected image. cvEigenDecomposite() is used to project the test image.

Before doing final step we should pass noEigens(number of eigenvalues) and EigenVectorArray(array of eigenvectors).This time instead of training image we pass a test image as the first parameter.The output from cvEigenDecomposite() is stored in a local variable (testprojectface).The system uses OpenCV matrix for storing the projected test face.

4 ANALYSIS AND RESULTS

4.1 Analysis

The analysis process involves the following steps:

Step 1: Face Detection and Extraction:

Images can be captured with the help of webcam on the user side.

Start:

The captured image should be processed and extracted.

The eigenvalue of the captured image should be calculated and should be compared with eigenvalues of existing face images in the database

If the eigenvalues matches recognition step will be done otherwise, save the new face image information in the face database (xml file).

End

Step 2: Face Recognition:

The basis for the face recognition is PCA algorithm and using PCA the following steps would be followed for face recognition:

Start:

The information about the matched face image can be found from the database.

The name field in the face recognition module is added to the MS Access Database along with the date to make the completion of attendance for each student.

End

4.2 Results

The result of the analysis process is presented here in the form of grayscale images.

Fig. 3. Images stored in xml database (Trained Faces)

The team performed a set of experiments to demonstrate the feasibility of the system.45 different images of different persons are used in training dataset and the result is binary image and is shown in Figure 3 using fisher discriminant framework method.

TABLE 1
INFERENCE ABOUT FACE RECOGNITION

Face Angles	Percentage of Detection rate	Percentage of recognition rate
0° (Frontal Face)	96.5 %	94 %
20°	78 %	75 %
60°	57 %	55 %
75°	1.2 %	0.65 %
90°	0 %	0 %

5 Conclusion

In order to reduce the faculty effort and to manage the time effectively the authors proposed automated attendance system base on face recognition in schools/colleges.The system takes attendance for particular amount of time and after the time expires the system automatically closes the attendance.The result of the experiment shows improved performance in the estimation of attendance compared to traditional pen and paper type attendance system.The current work is mainly focussed on face detection and extraction by PCA algorithm in video frames or images.

In further work authors are intended to improve face recognition by comparing 3D face images with 2D face images(Real time).Also the authors are intended to improve on multiple face recognition at the same time so that the effectiveness of time can still be managed and try to improve on the portability of the system.

References

[1] M. A. Turk and A. P. Pentland, "Face Recognition Using Eigenfaces," in Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. 586–591. 1991.

[2] A. J. Goldstein, L. D. Harmon, and A. B. Lesk, "Identification of Human Faces," in Proc. IEEE Conference on Computer Vision and Pattern Recognition, vol. 59, pp 748 – 760, May 1971.

[3] T. Ahonen, A. Hadid, M. Peitikainen, Face recognition with local binary patterns. "In Proc. of European Conference of Computer Vision", 2004.

[4] M. H. Yang, N. Ahuja, and D. Kriegmao, "Face recognition using kernel eigenfaces," IEEE International Conference on Image Processing, vol. 1, pp. 10-13, Sept. 2000.

[5] Y. Zhang and C. Liu, "Face recognition using kemel principal component analysis and genetic algorithms," IEEE Workshop on Neural Networks for Signal Processing, pp. 4-6 Sept. 2002.

[6] M. A. Fischler and R. A. Elschlager, "The Representation and Matching of Pictorial Structures," IEEE Transaction on Computer, vol. C-22, pp. 67-92, 1973.

[7] P. Sinha, B. Balas, Y. Ostrovsky, and R. Russell, "Face Recognition by Humans: Nineteen Results All Computer Vision Researchers Should Know About," in Proceedings of the IEEE, vol. 94, Issue 11, 2006.

[8] A. T. Acharya and A. Ray, Image Processing: Principles and Applications, New York: Wiley, 2005.

[9] T. D. Russ, M. W. Koch, and C. Q. Little, "3D Facial Recognition: A Quantitative Analysis," 38th Annual 2004 International Carnahan Conference on Security Technology, 2004.

[10] S. S. R. Abibi, "Simulating evolution: connectionist metaphors for studying human cognitive behaviour," in Proceedings TENCON 2000, vol. 1 pp 167-173, 2000.

[11] B. Tacacs and H. Wechsler. Face recognition using binary image metrics. In Proceedings, Third International Conference on Automatic Face and Gesture Recognition, pages 294–299, April 1998.

[12] Y. Cui, J. S. Jin, S. Luo, M. Park, and S. S. L. Au, "Automated Pattern Recognition and Defect Inspection System," in proc. 5 th International Conference on Computer Vision and Graphical Image, vol. 59, pp. 768 – 773, May 1992.

[13] J. Zhu and Y. L. Yu, "Face Recognition with Eigenfaces," IEEE International Conference on Industrial Technology, pp. 434 -438, Dec. 1994.