*International Journal of Latest Trends in Engineering and Technology*
*Vol.(8)Issue(4-1), pp.204-209*
*DOI: http://dx.doi.org/10.21172/1.841.35*
*e-ISSN:2278-621X*

# EMBEDDED DESIGN AND IMPLEMENTATION OF FACE DETECTION AND FACE RECOGNITION SYSTEM WITH HAAR CASCADE CLASSIFIER AND EIGENFACE ALGORITHM

Taner Arsan [1] & Ebru Günes[2]

**Abstract-In this paper, a fast face detection and face recognition system has been designed and implemented as a part of an embedded smart home system. Haar Cascade Classifier is used for face detection and Principal Component Analysis and Eigenface Algorithm are used for face recognition. Arduino Uno is used for hardware design. C#, and for professional design, DevExpress in Visual Studio are used for software components. Test results show high accuracy in terms of face detection and face recognition.**

**Keywords – Eigenface Algorithm, Face Detection, Face Recognition, Haar Cascade Classifier, Principal Component Analysis**

## 1. INTRODUCTION

The developments in image processing and pattern recognition as well as the development of computing power of computers have turned Computer Vision into reality. At this point it is very important to state that detection of human faces is an important component in human-computer interaction applications. The face detection is to find the position and size of one or more faces in an image or video sequence [1-3]. In recent years, face recognition systems have gained importance in many areas, especially when security is important [4]. The face is analyzed and the necessary permissions such as access to the user are given. This system can be used in public and private areas (factory entries, workplaces, offices, houses, etc.). In many studies, face recognition systems of different types and features have been developed [5-6]. In this article, an embedded system, which could be part of a smart home system, was designed and implemented with both software and hardware components.

The rest of the paper is organized as follows. Methodology and algorithms used in this study, are explained in section II, test results are presented in section III, and concluding remarks are given in section IV.

## 2. METHODOLOGY

### *A. Haar Cascade Classifier*

Haar Cascade Classifier was proposed by Paul Viola and Michael Jones to find and follow objects in videos or pictures [7]. Also known as the Viola and Jones object detection framework or Viola and Jones object detection structure. In the most basic sense, objects that are required to be found by a certain algorithm are first introduced to the computer, and then the images or video frames in which similar shapes are found and scanned to find that object. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images.

The classifier needs positive pictures in which the wanted face is found and negative pictures in which the face is not found. Haar Cascade scans the frames in the positive images with the frames set at certain magnitudes as below and checks the dark luminance values of the sum of the pixel values in the black region and the white region in the frame to generate specific target values. Haar Cascade features are represented in Figure 1.

When these frames (weak classifiers) are considered for each image size, hundreds of thousands of kernels will form. Negative images will be scanned that's why the majority of the frames that will not be used will be eliminated as there are no objects or faces inside. In positive images, objects will be selected and the frames to be used in the object will be determined. The first feature selected is relies on the focus on the region of the eyes is often darker than the region of the nose and cheeks. The nose may be selected with the ratio of the brightness of the cheeks being lower than the brightness of the nose zone. In the Haar Cascade Classifier, the integrals of the pictures are taken first. Thus, instead of calculating individual sums of pixel values, it is calculated with an integral. Thus, a great deal of processing power will be removed from the computer. Furthermore, in the process of finding objects, instead of being scanned repeatedly by each magnitude frame, only the parts that match the previous steps are scanned and process load is reduced [7].

[1] *Department of Computer Engineering, Kadir Has University, Istanbul, Turkey*
[2] *Department of Computer Engineering, Kadir Has University, Istanbul, Turkey*

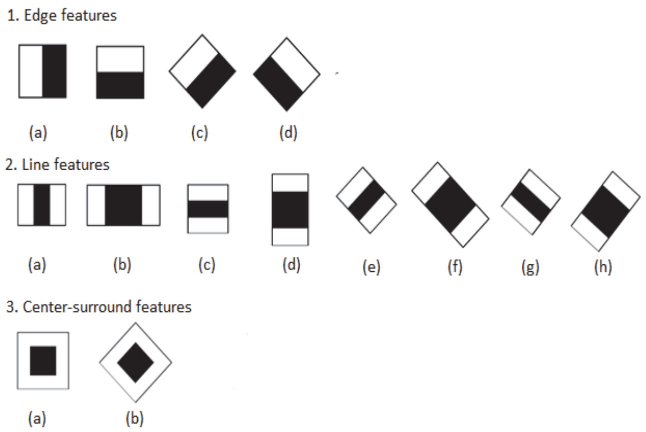


Figure 1. Haar Cascade Features.

### *B. Principal Component Analysis (PCA)*

PCA (Principal Component Analysis) reduces the size of the dataset during data analysis. PCA is a method used to reduce the working dimension and to reveal differences between objects [8]. Principal Component Analysis can reduce data in N-dimensional space to lower spaces. There is no significant loss of information in this dimension reduction. It is also realized more efficiently when performing this size reduction process. The aim of PCA can be expressed as:

1-) PCA has some jobs such as estimate, redundancy removal, feature subtraction, data compression, because applications having linear models are appropriate, such as signal processing, image processing, system and control theory, communications, etc.

2-) Face recognition can be applied to many areas. Also, face recognition can be classified into 3 parts. These sections are face identification, face sorting, or gender specification. The most notable of these applications include crowd surveillance technology, video content indexing, individual identification such as driver's license cup shots matching, entrance security, etc. The principal idea of using PCA for face recognition can be named eigenspace projection which is to state the large 1-D vector of pixels built from 2-D obverse image into the compact main components of the feature space. We must do recognizing the eigenvectors of the covariance matrix derived from a series facial images (vectors) for calculate to eigenspace.

### *C. Face Recognition Algorithm - Eigenface Algorithm*

Eigenface Algorithm is one of the face recognition techniques. This technique is the name given to a set of eigenvectors when they are used in the computer vision problem of human face recognition. A set of eigenfaces can be generated by performing a mathematical process with PCA (Principal Component Analysis) on a wide set of images showing various human faces. The eigenfaces method is based on the maximization of the scattering of facial images with the Karhunen-Loeve transformation and the linear projection of the small dimension to the longest. The Eigenface method was first applied by Sirovich and Kirby in 1987 to show the face effectively [9]. And also the Eigenface method was developed by Turk and Pentland for face recognition [10]. The imaginary image obtained from the differentiating components of the face is called eigenface. The Eigenface algorithm aims to find the covariance matrix representing the distribution of faces. These eigenvectors can be thought of as a group of features representing the differences between faces [11]. Eigenfaces are the eigenvectors of the covariance matrix of the distribution that the images in the training set generate. The images in the training set can be symbolized as $\mathbf{\Gamma_1, \Gamma_2, ,} \ldots , \mathbf{\Gamma_M}$.
The averages of the images in this cluster are obtained as

$$\mathbf{\Psi} = \frac{1}{M}\sum_{n=1}^{M}\mathbf{\Gamma_n} \tag{1}$$

The mean difference vector of each image can be expressed by $\mathbf{\Phi_j} = \mathbf{\Gamma_j} - \mathbf{\Psi}$. From these obtained vectors, it is desired to find the vector M which gives the best distribution of the domain, the vector of the orthonormal $\mathbf{u_n}$. Each of these $\mathbf{u_n}$ vectors is selected as to maximize the $\mathbf{u_k}$ coefficients defined below.

$$\boldsymbol{\lambda_k} = \frac{1}{M}\sum_{n=1}^{M}(\boldsymbol{u_k^T\Phi_n})^2 \tag{2}$$

The $\mathbf{u_n}$ vectors should provide in equation (3) with orthonormal condition.

$$\boldsymbol{u_l^T u_k = \delta_k = \begin{Bmatrix} 1, & l = k \\ 0, & l \neq k \end{Bmatrix}} \tag{3}$$

**u$_k$** vectors and **λ$_k$** constants are the eigenvectors and eigenvalues respectively of the covariance matrix **C** defined below.

$$\mathrm{C} = \boldsymbol{\frac{1}{M}\sum_{n=1}^{M} \Phi_n \, \Phi_n^T = AA^T} \tag{4}$$

$$\boldsymbol{A = [\Phi_1 \quad \Phi_2 \quad \dots \quad \Phi_M]} \tag{5}$$

The eigenvectors of the given **C** matrix form the eigenvectors of face space. For face images of **N**x**N** size, **C** matrix is **N$^2$**x**N$^2$** dimension. The original feature vector **Ω$^T$** of any face image can be found as follows, using the original face.

$$\boldsymbol{w_k = u_k^T(\Gamma - \Psi)} \tag{6}$$

$$\boldsymbol{\Omega^T = [w_1 \quad w_2 \quad \dots \quad w_M]} \tag{7}$$

It can be approximated again with the projection of any face image to face space.

$$\boldsymbol{\Gamma = \Psi + \sum_{i=1}^{M} w_i u_i} \tag{8}$$

The eigenfaces will be formed as dark and light areas that are compose in a particular model. This model is how the different features of a face are singled out to be interpreted and scored. Face image samples of this algorithm are given in Figure 2.

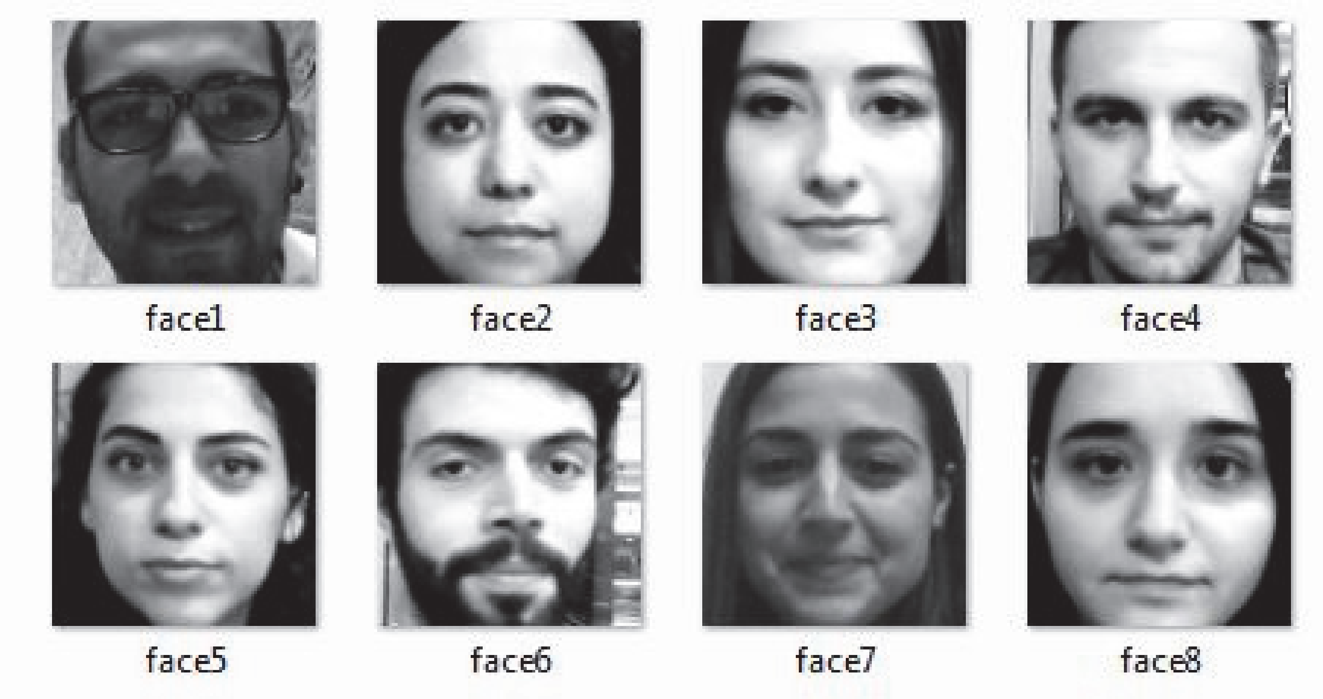


Figure 2. Face Image Samples of Eigenvalues Algorithm.

*1) Reason for using Eigenface Algorithm*

Eigenface recognition algorithm has several advantages: Eigenface provides a cheap implementation, completely automatic and easy to code, having a high recognition speed, eigenface adequately reduces statistical complexity in face image representation, possibility to use in real time applications, eigenface can handle large databases, no knowledge of geometry and reflectance of faces is required and eigenface provides high recognition rate. Eigenface allows reconstruction, so once the eigenfaces are calculated, each face image in the image space can be transformed to face space. These advantages reflect the power of appearance-based approach in ease of implementation.

### *D. Hardware Components*

Arduino UNO has been used to inform the users and to open door with the help of sensors. It works by getting commands “1” or “0” from C# according to the codes written in Arduino IDE. C# sends data to device that it is intended to be wireless with the help of Bluetooth communication. Apart from the main modules like Keypad, LCD, Solenoid, Bluetooth; additional components like relay, battery have been used to operate these modules. In additional, LEDs, resistors and cables were used. For these components, Password.h, Keypad.h, Wire.h, LiquidCrystal_I2C.h and SoftwareSerial.h libraries have been used by Arduino IDE. When the device receives the "1" command, the user is informed, the solenoid is activated, the green light is on; when it receives "0" command, it has been aimed to perform functions such as user information message and red led lighting. In addition, a password has been set for emergencies and if this password was entered from the keypad, it has been expected to execute a related operation. There are several options such as Arduino, Intel Edison/Galileo, Beaglebone or Raspberry Pi available to implement such a system. Followings reasons became important to choose Arduino for implementation. Arduino is relatively cheaper than other microcontroller platforms, Arduino can work with cross platforms such as Linux, Windows and MacOS. Most microcontroller system is limited to Windows, Arduino provides simple programming environment, and it is open source and easily software and hardware expandable.

### *E. Software Design Components and Implementation*

A database was created for the admin login. SQL Server 2014 Management Studio was used for this studio. Firstly, “facelib” was created for this program. A table is implemented and it is called Admin_Login. This table contain five columns, these are Admin_Id, Admin_username, Admin_password, Admin_image, Admin_mail. Admin_Id is primary key for Admin_Login. This table is executed by login button, change password button and signup button. Finally, a store procedure is coded in order to add a new admin. This procedure takes information of admin from application. It has like Admin_username, Admin_password, Admin_Image and Admin_mail paramaters.

Admin page is shown in Figure 3, and admin name and password is checked on database. User information class is also generated. Controlling name and password is made due to this class when someone clicks the login button. If Admin_Login table has these information then, face recognition page is opened, and send the mail at admin’s e-mail address. However, if information is not correct, lbl_message shows “There is no such password or username” message.

If admin wants to change password, user writes a name and new password to textboxes and clicks change password button. As mentioned above, user information class has function to change password. While changing password, Admin_username column is the significant criterion. When the admin entry is done, the admin username uploaded from database. The communication between the device and the application is provided via Bluetooth. C# to uses serial port to communicate. COM9 is selected from the combo box for this Arduino Bluetooth connection. When the connection occurs, “connected” message is shown in the label.

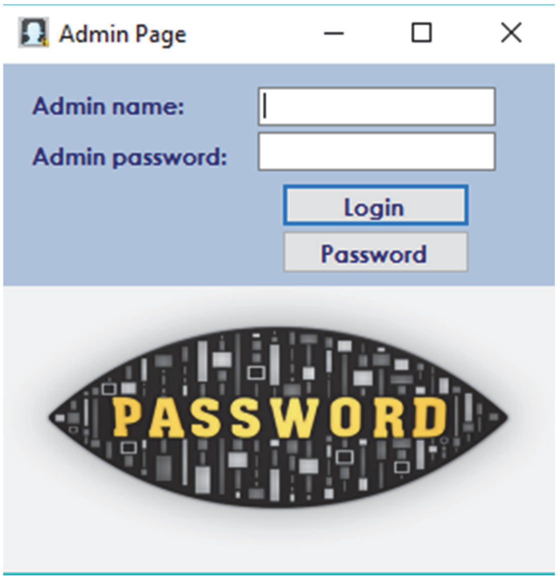
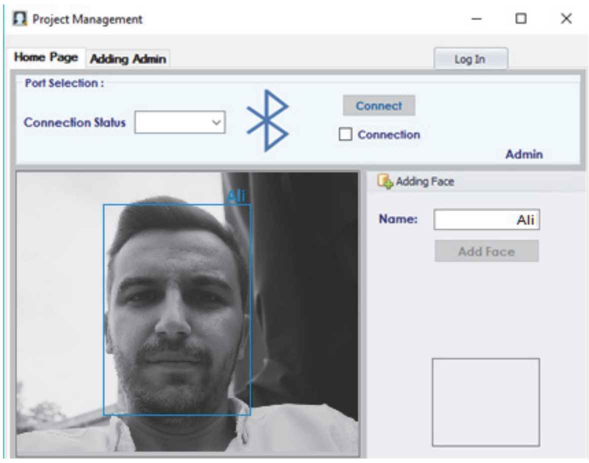


Figure 3. Admin Page.

When camera is initialized in home page as seen in Figure 4 with form load, each_face is initialized. This function starts detection, gets face and gets results from eigenobjectrecognizer class. Finally, if the person is recognized, face concatanates with name. When recognition is done, blue frame is appeared. Name is written up of frame. SendOne function is initialized and it sends one to Arduino via Bluetooth. Add Face Button is used if the detected face introduced newly. Here, for getting an image to user a timer is used in every 3 seconds. Label says “look left, right and camera” because of many profile takes good recognition. Name is kept from textbox, and it is saved. These images appear gray type on picture box because, Haar Cascade method allows gray image. Program database is in program stored at RecognizedFaces folder. This folder has recognized faces and added_name.txt document.



Name: Ali

Figure 4. Home Page.

## 3. TEST RESULTS

The software was tested on different conditions and in different ways to see how efficient it is. In these tests, four different criteria have been used, namely, recording the face from a single angle and from different angles, recognizing the previously added face in the same light and in different light. Face recognition process can be described as follows:

**Case 1:** Randomly selected 30 face in previously added 104 face, when trying to recognize a face in a different light compared to photographed in the database, it has not recognize 13 of them. This means that it has approximately **56.67%** recognition rate. Randomly selected 30 face in previously added 104 face, when trying to recognize a face in a same light compared to photographed in the database, , it has not recognize 6 of them. This shows that it can recognize **80%**.
**Case 2:** Randomly selected 30 face in previously added 104 face, when trying to recognize a face in a different light compared to photographed in the database, it has not recognize 6 of them. This means that it has 80% recognition rate. Randomly selected 30 face in previously added 104 face, when trying to recognize a face in a same light compared to photographed in the database, it has not recognize 1 of them. This means that it has **96.67%** recognition rate.
**Result:** Based on the results obtained from the tests, when we added a face with a few photographs from different angles, we observed that the recognition rate increased. We also observed that the program is very sensitive to light and when it saw a face which is in the database in same light although recognized in the same light, when it saw the same person in different light, it has difficulty to recognize. A test example is given in Figure 5.

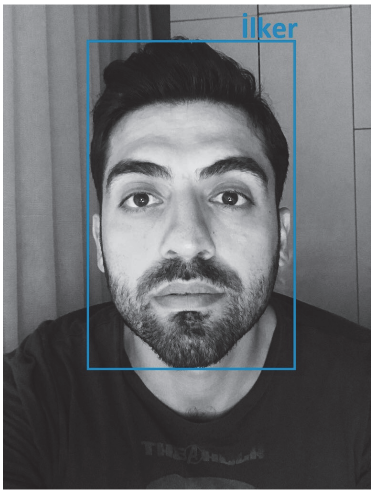


Figure 5. Face Recognition Test Example.

## 4. CONCLUSION

In this paper, embedded design and implementation of face detection and face recognition system has been introduced. This system includes hardware and software components and it is planned as a part of an embedded smart home system. Face detection is achieved by Haar Cascade Classifier, then Principal Component Analysis and Eigenface Algorithm are used for face recognition. Arduino Uno is an effective solution for these type smart systems applications. Tests results show **96.67%** recognition accuracy rate and it is a very good results for this type embedded systems.

## 5. ACKNOWLEDGMENT

Authors thank Merve Durkaya, Selin Sancaklı and Taha Berk Aksu for their valuable contribution and support.

## 6. REFERENCES

[1] M. Yang and D. J. Kriegman and N. Ahuja, “Detecting Faces in Images: A Survey,“ IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 1, pp.34–58, 2002.

[2] E. Hjelmas, B. K. Low, “Face Detection: A Survey,“ Computer Vision and Image Understanding, vol. 83, no.39 pp.236–274, September 2001.

[3] T. Sauquet, Y. Rodriguez and S. Marcel, “Multiview Face Detection,“ IDIAP, IDIAP-RR, no:49, 2005.

[4] Z. Lei, C. Wang, Q. Wang, and Y. Huang, “Real-Time Face Detection and Recognition for Video Surveillance Applications”, pp. 168–172, 2009.

[5] F. Sayeed and M. Hanmandlu, “Properties of information sets and information processing with an application to face recognition,” Knowledge and Information Systems, vol. 52, no. 2, pp. 485–507, Aug. 2017.

[6] K. Nasrollahi and T. B. Moeslund, “Haar-like features for robust real-time face recognition”, pp. 3073–3077, 2013.

[7] P. Viola and Michael J. Jones. "Robust realtime face detection", International Journal of Computer Vision, 57(2):137-154, 2004.

[8] Y.L. Hsu, P.Y. Huang, and D.T. Chen, “Sparse principal component analysis in cancer research,” Translational Cancer Research; Vol 3, No 3, Translational Cancer Research (Statistical and Bioinformatics Applications in Biomedical Omics Research), June 2014.

[9] L. Sirovich, M. Kirby, “Low-dimensional procedure for the characterization of human faces.”, J Opt Soc Am A., March 4(3):519-24, 1987.

[10] M. Turk, A. Pentland, “Eigenfaces for Recognition”, Journal of Cognitive Neuroscience, Volume 3, No:1, MIT, 1991.

[11] C. Sreejith, K. Sreeshma, C. H. Suhanna, N. Sabina, P.K. Shabanu, “An Experimental Study on Object Recognition Using Eigen Algorithm”, Journal of Computer Applications (JCA), ISSN: 0974-1925, Volume IV, Issue 3, 2011.