

Security System Design Based on Human Face Detection and Recognition on Android Platform

Sheshmal S. Shingne¹, V. Krishnamurthy²

Assistant Professor, ETC Engineering Dept., Lokmanya Tilak College of Engineering, Navi Mumbai, India¹

Director Incharge, Scientist-E, National Institute of Electronics and Information Technology, Chennai, India²

Abstract: There are various electronic security systems available, based on human motion sensing, biometric features sensing, code or password identifications etc. In this system human face detection and recognition is used for opening and closing the doors, by means of switching ON and OFF of relays. By relays the various motors can be operated to door movement. The system is developed on android platform i.e. an android OS is installed on hardware and an android app is running on android OS. The hardware platform is ARM11 based development board. In this system the human face is detected and then recognized with faces present in the database of system. For face detection ‘Haar feature-based cascade classifiers’ algorithm is used. And for face recognition ‘Principal Component Analysis’ (PCA) algorithm is used. For both algorithms OpenCV and JavaCV libraries are used. Application is developed in JAVA in eclipse IDE . For actuation relays the Atmega AVR microcontroller is used. The ARM11 board gives the command to the AVR microcontroller via RS232 comm. Protocol.

Keywords: Face Detection, Face Recognition, Android OS, ARM11, Haar Features based cascade classifier, Principal Component Analysis(PCA), AVR, Atmega8, OpenCV, JavaCV, Eclipse IDE, JAVA, JDK, android plugins, NDK plugins.

I. INTRODUCTION

Security is the major concern of today. The need of security is required everywhere from home to commercial, official and from defense point of view. Over the days the security is provided or accessing system is developed based on password access, mechanical system based access and biometrical features based access. In biometrical based systems fingerprint, iris, and face recognition and detection are widely used. This project is a M.Tech. project work as the standalone hardware implementation for the door access control, by the use of users face detection and recognition methods. The hardware is developed with the ARM 11 microcontroller based development board with android OS platform and AVR microcontroller board for door access DC motors control. For face detection ‘Haar features based Cascade Classifiers’ algorithm is used and for face recognition ‘Principal Component Analysis’ PCA algorithm is used. Both the algorithms are tested in Matlab as well. For hardware implementations coding these algorithms are applied by the use of Open CV library which provides the standard APIs while programming. For hardware realization of the project there are various platforms are available; such as OMAP processor based design, ARM processor based, POWERPC, DSPIC and FPGA based design. FPGA based design is the best solution but considering less complexity in hardware and application, ARM 11 processor based option is chosen.

II. SYSTEM ARCHITECTURE AND EXPLANATION

Following Fig.1 show the system architecture block diagram including the ARM processor based development board, camera, and motor controlling AVR board (called relay/motor driving card), TFT, LCD and related hardware.

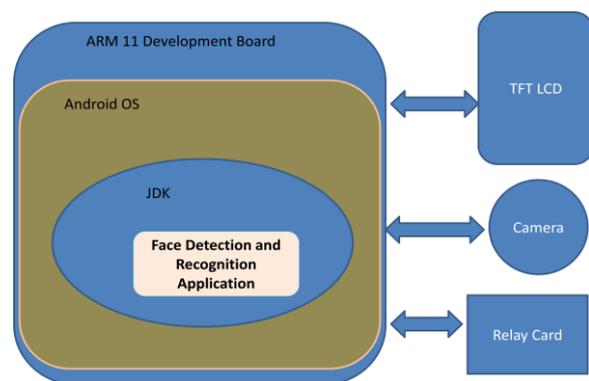


Fig. 1 System architecture block diagram

As shown in Fig.1, The system consist of ARM 11 based development board (Friendly ARM mini 6410) with Samsung’s 533 MHz Samsung S3C6410A ARM1176JZF-S with VFP-Unit and Jazelle (max freq. 667 MHz) ARM11 processor. The processor have lot of features which makes it more suitable for the required application; such as RAM up to 256 MB DDR RAM, 32 bit Bus, Flash up to 1GB NAND Flash, EEPROM 256 Byte (I2C), Ext. Memory with SD-Card socket, Serial Ports- 1x DB9 connector (RS232), total 4x serial port connectors, 1x USB-A Host 1.1, 1x miniUSB Slave/OTG 2.0, Camera of 20 pin (2.0 mm) Camera interface(CMOS), LCD Interface for 4.3”, 5”, 7” LCD interface with 40 pin (2.0 mm) and 41 pin interfacing connector.

As the development board supports USB connectivity, USB based camera is used as image sensor, to display the images and required controlling purposes a 7” TFT LCD is used which comes interfaced along with the development board.

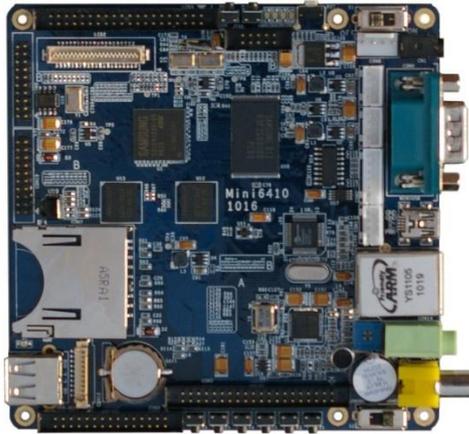


Fig. 2. FriendlyARM 11 Development Board

TFT LCD is connected by 40 pin connector with the board. RS 232 is used to communication between the development board and relay Card board which have AVR Atmega 8 microcontroller, which accepts commands coming from ARM 11 board and controls the relays and DC motors through the L293D IC which controls the movement of the doors that is our application, as shown in fig.3.

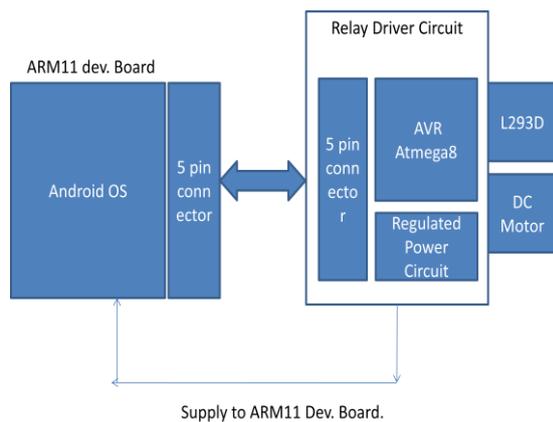


Fig. 3. Block diagram with detailed relay/motor driver circuit

The following fig. 4 is the circuit diagram of the AVR based relay /DC motor driving card. The circuit consists of AVR Atmega 8 microcontroller, L293D motor driver IC with relays which are with the door extended through the connecting wire and Max 232 IC for the communication with the PC and programming and to communicate with the ARM11 board. The circuit also consists of low cost the step down transformer and 7805 regulator based AC/DC 5V DC power supply. The same DC supply is also given to the ARM11 development board. This makes the system a standalone product which accepts the 230VAC supply and works independently.

Development board consists of 1GB NAND flash and android OS is installed in it. The lots of device drivers for android OS such as USB, RS232 are developed by the Friendly ARM and given the Android OS which contains all these device drivers so while application development lot of time can be saved.



Fig.4. AVR based Motor Driving card

The android Application is developed in the Eclipse IDE for which JDK (Java Development kit) is required to pre installed. The application programming is done in Java along with the OpenCV library installed which provides the face detection and recognition APIs for the application. The application controls all the required hardware as per the requirement.

III.SYSTEM OPERATION

The following Fig.5 shows the process flow diagram of the face detection and recognition steps. As shown in the figure, on the LCD screen an ‘welcome’ letter continuously displays on the half portion of the LCD screen. And on the remaining half portion live video stream is continuously displayed and here face detection algorithm (Haar feature based cascade classifier) continuously runs.

Whenever any person comes and sees into the system, then face into the video stream is detected, if proper human face is detected then system stops taking video streams and displays the detected cropped human face into the LCD window. Then system takes the pixel information of the cropped face image and then applies the face recognition algorithm (Principle Component Analysis). If that particular face matches with the face already present into the system database then on LCD screen it displays the face with its name into the database, and ‘welcome’. If that face does not found into the database then it displays message on the screen that ‘Sorry!! You r not authorized to access’. And again new live video stream start to display on LCD screen.

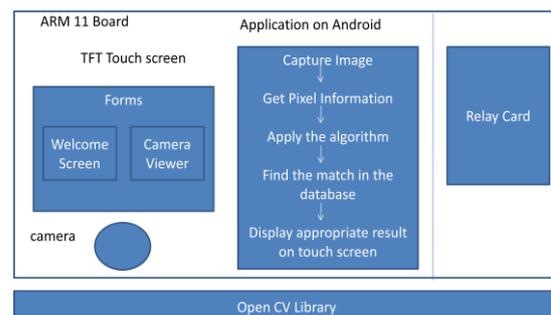


Fig. 5 System process diagram

Such a detailed face detection and recognition process and message display is shown in following fig.6.

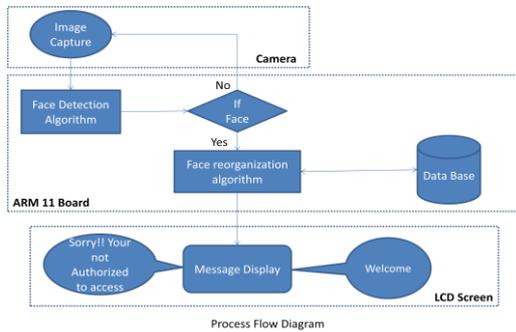


Fig.6 Process flow diagram

Whenever any human face is detected and recognized and found in database, then ARM 11 board sends commands to the relay card (AVR uC) via RS232, and which drives the DC motor for 15 seconds in clockwise direction i.e. to open the door driven by DC motor , then waits for next 15 seconds and then again drives the DC motor for 15 seconds in anticlockwise direction, i.e. to close the door. The DC motor is driven through through the motor driver IC L293D.

IV. ALGORITHM DEVELOPMENT AND TESTING

A. Face detection explanation

Face detection is achieved by using the “Haar like feature based Cascade Classifier” algorithm. It’s an effective face/object detection method proposed by Paul Viola and Michael Jones in their paper “Rapid Object Detection using a Boosted Cascade of Simple Features” in 2001 [1]. It’s a machine learning based approach where a cascade function is trained from lot of positive and negative images. It is then used to detect the faces in other images. This algorithm works broadly into two parts. First training phase and then second detection phase. Initially the algorithm needs lots of positive (images with faces) and negative (images without face) to train the classifier. Classifier is the group of features that is applied to the image in which face is to be detected. Feature is the single value obtained by subtracting the sum of pixels under white rectangle from sum of pixels under black rectangle from the group of pixels in rectangle form taken from the face region.

Following fig.7 shows the various types of rectangles that can be taken from the face to get particular features. Fig.8 shows the face regions from which such features are being collected. To compute the feature rapidly the integral image is formed. And large no. of features are formed from selecting all possible rectangle group of pixels called kernels. Then applying all the features on all the positive and negative images. For each feature it finds the best threshold which will classify the faces to positive and negative. We select the features with minimum error rate, which means they are the features called classifiers that best classifies the face and non face images.

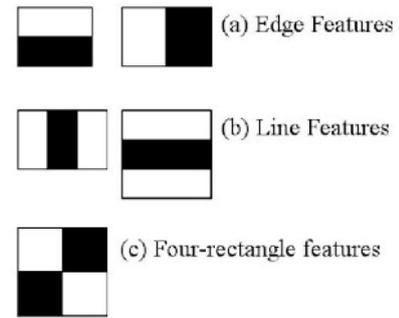


Fig. 7 Various types of features

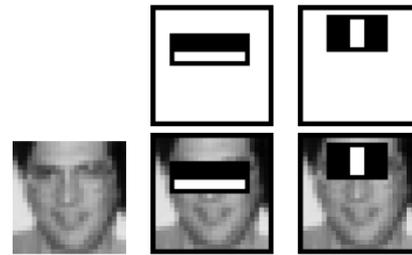


Fig. 8. Face regions for feature selection

To get the best result the group of weak classifiers form a final classifier; and group of such a classifiers forms a strong classifier. In an image of non face region so to detect it in first there is a concept of cascade of classifiers. Which is based on the principle the instead of applying all the features on the window, grouping the features into different stages of classifier and apply one by one as shown in fig. 9. If a window fails the first stage then discard it and no need to consider the remaining features on it. If it passes then apply second stage of features and continue the process. The window which passes all the classifier stages is the face detected image.

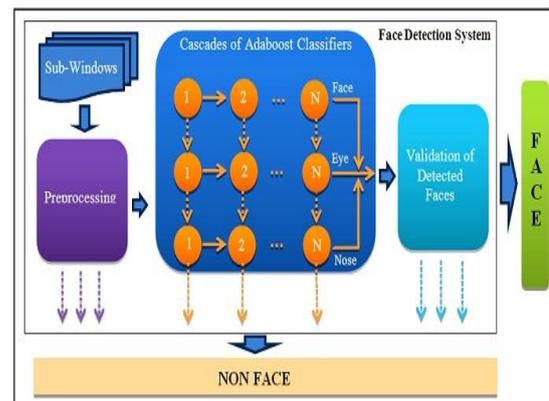


Fig. 9 Cascade of classifier

B. MATLAB testing of face detection

As our final aim was hardware implementation , so the Matlab testing is done with the face detection given in computer vision system toolbox. It comes with already trained classifiers. This was to test the required efficiency of face detection. The fig.10 shows the result of face detection in MATLAB.

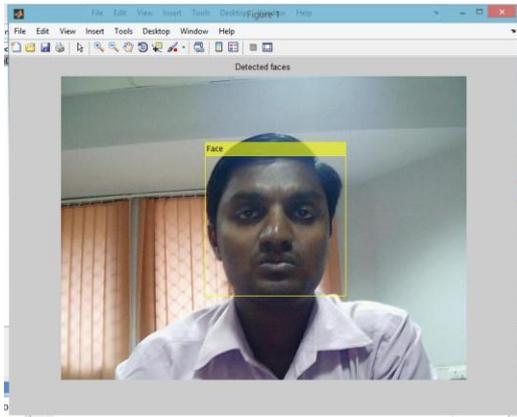


Fig.10 Face detection result in MATLAB

C. *Face recognition explanation*

Principle Component Analysis (PCA) is one of the basic and still most efficient method used for face recognition. PCA involves a mathematical procedure that transforms a number of possibly correlated variables into a number of uncorrelated variables called Principle Components, related to the original variables by an orthogonal transformation. The major advantage of PCA is the eigenface approach helps reducing the size of the database required for recognition of the test image. The trained images are not stored as the row images rather they are stored as their weights which are found out projecting each and every trained image to a set of eigenfaces obtained. The explanation of PCA is as follows:

Firstly Taking the detected N no. of images then forming the Images Matrix. For explanation considering the example image as follows of 128 x 128 pixels having individual pixels values given in decimal values equivalent of the 8 bit character value.

$$\begin{matrix} \text{Image} \\ \text{[128x128]} \end{matrix} = \begin{bmatrix} 150 & 152 & \dots & 151 \\ 131 & 133 & \dots & 72 \\ \dots & \dots & \dots & \dots \\ 144 & 171 & \dots & 67 \end{bmatrix} \quad 128 \times 128$$

And then convert this image into a single image row of dimension of 1 x 16K as below:
[150 152 .. 151 131 133..72 ...67]1 x 16K

And then forming a image matrix of all the images for example as below where each row refers to one image.

$$D = \begin{bmatrix} 150 & 152 & \dots & 254 & 255 & \dots & 252 \\ 131 & 133 & \dots & 221 & 223 & \dots & 241 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 144 & 171 & \dots & 244 & 245 & \dots & 223 \end{bmatrix} \quad N \times n$$

where N refers to the total number of images and n refers to number of pixels into single image. Then calculating the mean image from the image matrix D. Next is subtracting the mean(μ_x) from the individual image row from image matrix which is the mean centered data matrix U. Then calculate the covariance of the image matrix.

$$\Sigma = U'U / (N-1)$$

Next is to calculate the eigenvalues and eigenvectors of the covariance matrix Σ .

$$\Sigma = \phi \lambda \phi'$$

Where ϕ is eigenvector matrix, ϕ' is its transpose and λ represents eigenvalues matrix.

Then it is required to choosing the highest valued eigenvalues to represent the image and forming the feature vector of the eigenvalues corresponding to the chosen eigenvalues as follows:

$$\text{Feature Vector}(\phi)_{pca} = [\phi_1, \phi_2, \phi_3, \dots, \phi_n]$$

Its result is deriving the new data set which is much reduced in dimension much lower than the original image and form a new test database as follows:

$$\text{Final data}(P\phi) = (Px - \mu_x) \cdot (\phi)_{pca}$$

Then next is calculate the test image's new data set and compare it with the data in the data set of the database and declaring the image recognition based on the lowest difference between the test new data to the image data into the database and finally retrieving the original image by adjusting the mean into the dataset and display the result image as follows:

$$Px = P\phi \cdot (\phi)_{pca} + \mu_x$$

Where, Px - the original image
 $P\phi$ - final database
 $(\phi)_{pca}$ - transpose of the feature vector
 μ_x - the difference image mean vector

D. *MATLAB testing of face recognition*

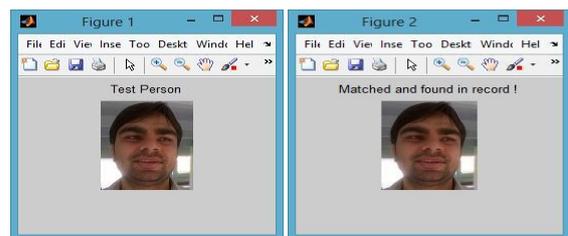


Fig.11 Face recognition result in MATLAB

As the first step is to create the database of images in terms of 'Eigenfaces', here the own database of 20 images is created that is called here as Traindatabase. Each image of 112 x 92 pixels. And the PCA algorithm is applied to the traindatabase images.

Then for every face to be recognized its face is cropped and converted into 112 x 92 pixels format and then placed into the test image folder. And when program is run by giving the test persons image; if the particular persons image is present into the database them it displays the result with the test person image as well as database image with label 'Matched and found in record !' as shown in fig.11.

V. HARDWARE TESTING AND RESULTS

The following fig.12 shows the ARM11 development board along with TFT LCD with android OS installed. When board is powered up then the android OS with various inbuilt apps appears on the TFT LCD. When the face detection and recognition program successfully build and downloaded into the board then its app icon appears on the LCD screen along with the different pre-existing icons.



Fig.12 ARM11 dev. Board with TFT LCD

Following consequent figures shows the testing and operating results of the project. Whenever project app is being run then USB camera starts working and live video stream appears into the LCD screen as shown in the following fig.13.

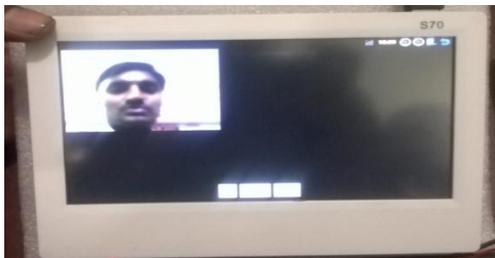


Fig.13 Initial LCD window when project run

If in the video stream, there exist a human face then it crops it and displays on the screen as shown in the fig.14.

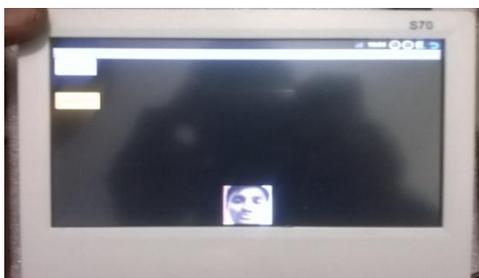


Fig.14 Face detection phase

When face is detected then camera stops working and it then starts face recognition stage. If that particular face is not being found into the database then there is a provision to store that face into the database or add that particular person into the database. So to add that person into the database following window appears as shown in fig.15.

Here is the access to the keypad where the particular name can be given to that person and then saved it.



Fig.15 Creating face database phase

When the particular person or its face already exist into the face database then that person face is displayed along with the name assigned to it, as shown in the fig.16. And next to that if the door is closed already then the DC motors get driven for 15 sec. to open the door and Welcome message appears on the screen for that particular person.

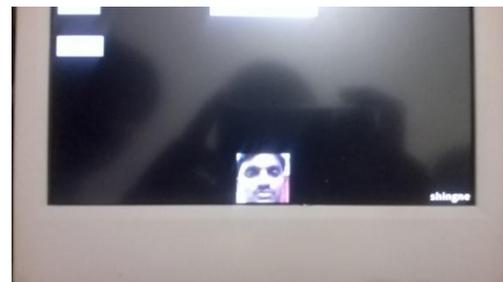


Fig. 16 Face recognition stage

VI.FUTURE SCOPE

The project can be merged with the existing biometric systems such as fingerprint recognition and iris based recognition to enhance the identity of particular person. While issuing citizenship card , voter ID to a particular human ; his/her face is already being kept into the database such as the India is adopting “Adhaar Card” in many applications and public services. So from Adhaar database we can get face data and so project can be implemented in security and persons online identification and investigation systems in many ways. And as it is into android platform, so the advantage is that now a day’s android based Smartphone’s and tablets are going to use with internet connectivity so the project can be implemented with lot of advancements in a large scale.

ACKNOWLEDGMENT

As it the M.Tech project work, lot of help is got from the remaining teaching faculties of the NIELIT, Aurangabad, especially from the **Dr. R. Maheshwari** (Director, NIELIT Aurangabad), **S.V.Valunekar** (Scientist E), **Mrs. W. Kandlikar** (Scientist D) in the successful completion of the work. Lot of assistance is got from my colleagues, especially from **Basant Sundesha** and **Vikas Solanki**.

REFERENCES

- [1] Paul Viola and Michael J. Jones, “*Rapid Object Detection using a Boosted Cascade of Simple Features.*” IEEE CVPR 2001
- [2] Rainer Lienhart and Jochen Maydt. “*An Extended Set of Haar-like Features for Rapid Object Detection.*” Submitted to ICIIP2002.
- [3] DOC493: “*Intelligent Data Analysis and Probabilistic Inference*” Lecture 15
- [4] “*A tutorial on Principal Components Analysis*” Lindsay I Smith, February 26, 2002
- [5] “*A tutorial on Principal Components Analysis*” Javier R. Movellan.
- [6] Yoav Freund and Robert E. Schapire. “*A decision-theoretic generalization of on-line learning and an application to boosting.*” In Computational Learning Theory: Eurocolt '95, pages 23–37. Springer-Verlag, 1995.
- [7] “*Android™ Application Development For Dummies®*”, 2nd Edition, by Michael Burton and Donn Felker, John Wiley & Sons, Inc.
- [8] C. Papageorgiou, M. Oren, and T. Poggio. “*A general framework for object detection*”. In International Conference on Computer Vision, 1998.
- [9] M. Turk, A. Pentland, “*Eigenfaces for Recognition*”, Journal of Cognitive Neuroscience, Vol. 3, No. 1, Win. 1991, pp. 71-86
- [10] “*Digital Image Processing*” Second Edition, Prentice Hall, by Rafael C. Gonzalez and Richard E. Woods.
- [11] “*Digital Image Processing using Matlab*” Prentice Hall, by Rafael C. Gonzalez and Richard E. Woods and Steven L. Eddins.
- [12] “*MATLAB Image Processing*” By Bruce Tannenbaum IEEE Boston Meeting, Dec 3, 2012
- [13] “*A Guide to MATLAB for Beginners and Experienced Users*” by Brian R. Hunt, Ronald L. Lipsman, Jonathan M. Rosenberg, Kevin R. Coombes, John E. Osborn, and Garrett J. Stuck; Cambridge University Press.
- [14] www.mathworks.com
- [15] www.developer.android.com