Improving Speed and Security of EVM's Using Finger Print and Face Recognition

Ashwin Chettri [1], Prof. D. G. Chougule [2]

[1] M.Tech Student, [2]Professor
[1,] Department of Technology, Shivaji University Kolhapur
[2,] TKIET, Warananagar

***Abstract*—Electronic voting system as adapted by many nations as a mode of electing contestants has been proved to be susceptible to tampering and rigging. The common voting machine lags proper way of authentication of voters. In the present scenario, authentication of voters is done manually in the polling station. A system which makes use of two biometric for authentication purpose ensure only eligible candidates vote and just once. The fact that every individual has unique fingerprint is exploited and used as one of the authentication tool. Face recognition is another tool used for authentication purpose. The overall system comprises of a fingerprint module along with the camera using ARM processor. Face recognition makes use of MATLAB for developing Eigen faces during initial training phase after which the system is completely stand-alone.**

I. INTRODUCTION

Biometrics or biometric recognition refers to the identification of humans based on physical or behavioural traits or characteristics such as face, fingerprints, iris, gait and voice. Biometrics is used in computer science as a form of identification and access control. It is also used to identify individuals in groups that are under surveillance. A biometric authentication system operates by acquiring raw biometric data from a subject (e.g., face image), extracting a feature set from the data (e.g., Eigen-coefficients), and comparing the feature set against the templates stored in a database in order to identify the subject or to verify a claimed identity[1].

Biometric identifiers are the distinctive, measurable characteristics used to label and describe individuals. Biometric identifiers are often categorized as physiological versus behavioural characteristics. A physiological biometric would identify by one's voice, DNA, hand print or behaviour. Behavioural biometrics is related to the behaviour of a person, including but not limited to: typing rhythm, gait, and voice. Some researchers have coined the term behaviometrics to describe the latter class of biometrics.

Many different aspects of human physiology, chemistry or behaviour can be used for biometric authentication. The selection of a particular biometric for use in a specific application involves a weighting of several factors. It is identified that seven such factors to be used when assessing the suitability of any trait for use in biometric authentication. 1) Universality: means that every person using a system should possess the trait. 2) Uniqueness: means the trait should be sufficiently different for individuals in the relevant population such that they can be distinguished from one another. 3) Permanence: relates to the manner in which a trait varies over time. More specifically, a trait with 'good' permanence will be reasonably invariant over time with respect to the specific matching algorithm. 4) Measurability (collectability): relates to the ease of acquisition or measurement of the trait. In addition, acquired data should be in a form that permits subsequent processing and extraction of the relevant feature sets. 5) Performance: relates to the accuracy, speed, and robustness of technology used (see performance section for more details). 6) Acceptability: relates to how well individuals in the relevant population accept the technology such that they are willing to have their biometric trait captured and assessed. 7) Circumvention: relates to the ease with which a trait might be imitated using an artifact or substitute.

In this paper, unique features of two biometric such as face and fingerprint are explored and used for authentication. A fingerprint module with its in built scanner using fingerprint and a camera along with ARM processor is used for authentication. The images from the camera are initially sent to MATLAB for calculation of Eigen faces, Eigen faces are then stored in the stand-alone system's Flash memory. Once the Eigen faces are calculated and stored in the embedded system Flash, the embedded system is completely stand-alone.

II. PORTABLE FACE RECOGNITION SYSTEM

The main aim of the project is to make a portable face recognition system using Arm Lpc 2148. In this project face recognition is meant to enhance the security of EVM's along with finger print as it is used for authentication. When one thinks of face recognition, one immediately thinks of finding features of a face: eyes, nose, ears, and cheek bones. But who's to say that these are the most distinguishing characteristics of a face, and that they are the best features by which a face should be described? And what if these features are correlated? Instead of hard-coding features for detection, I decided to find the orthogonal features that most optimally describe our large training set by using Principal Component Analysis. This creates an orthogonal basis of principal components for our training set. The basis vectors are known as eigenfaces, and can be thought of as characteristic features of a face. All new faces will be described as a linear combination of these eigenfaces. This is equivalent to projecting the new face onto the subspace spanned by our eigenfaces. By using only the eigenfaces with the highest eigenvalues, Principal component analysis (PCA), or Karhunen-Loeve transformation, is a data-reduction method that finds an alternative set of parameters for a set of raw data (or features) such that most of the variability in the data is compressed down to the first few parameters. Principal component analysis mathematically is an orthogonal linear transformation that changes the data into a new coordinate system such that the variance is put in order from the greatest to the least. Because of the decreasing variance property, much of the variance (information in the original set of p variables) tends to be concentrated in the first few PCs. This implies that we can drop the last few PCs without losing much information. PCA is therefore considered as a dimension-reduction technique.

The face recognition technique for a portable system consists of three stages namely; Training, Logging and enrolling mode as described below.

A) Training

The training process is the only time we use a computer; once this step is complete, the system is completely standalone. If we were to sell this system as a consumer product, we would ship the system pre-trained. The training process consists of teaching the system to key in on the most important features of a face. To do this, a large number of facial images are taken and sent to Matlab to help the system determine the distinguishing features of a face. Matlab will be used to create the Eigen faces, which are the principle components of the training set .The image will be send through the microcontroller to Matlab over the serial port. Once all of the training images are in Matlab the Eigen faces and average face are created. Finally, the Eigen faces and average face are sent to flash memory through the microcontroller. Once the Eigen faces and average face are in flash, the system is completely standalone.

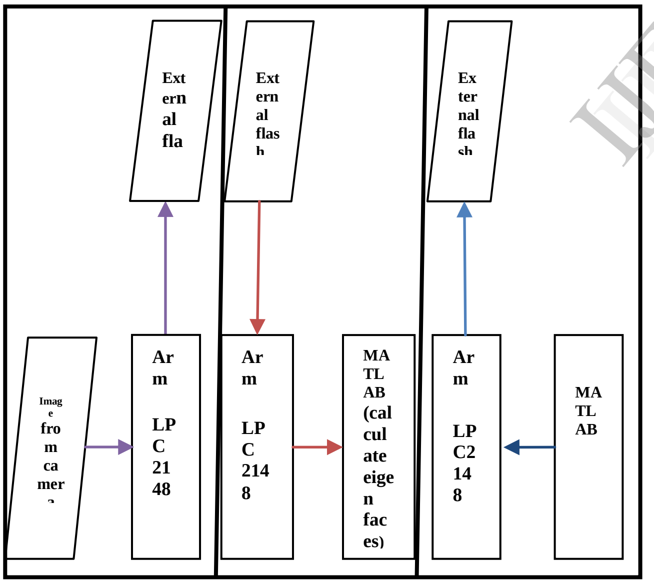


Fig .1 Showing different stages of training.

B) Logging

The Logging In process is initially very similar to enrolling. The user presses the login button to take his picture, store it in flash memory and begin the logging in process. Again the newly captured image and Eigen faces are pulled from flash back to the microcontroller to calculate the user's template. This template is compared with all of the previous templates. For the user to be logged in, their template needs to "match" (be close enough to) only one saved template; otherwise they will be "denied access". Again, the cosine of the angle between the two templates is used to determine template match. Whether or not the user was logged in, the top three matches are displayed on the LCD.

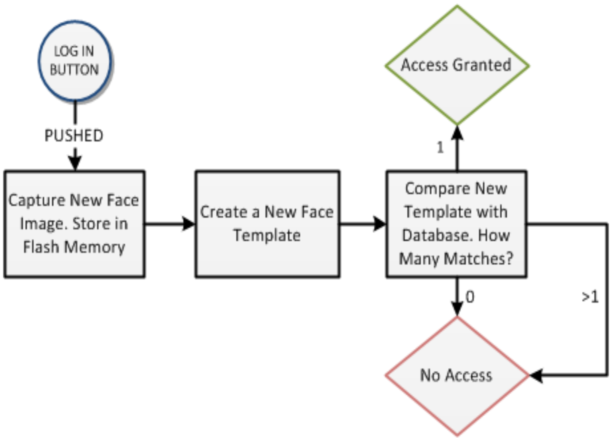

LOG IN BUTTON
PUSHED
Capture New Face Image. Store in Flash Memory
Create a New Face Template
Compare New Template with Database. How Many Matches?
1
Access Granted
0
>1
No Access

Fig 2. Logging in method.

C) *Enrolling*

Before a user can login to the system, he first needs to register his face and enroll it into the database. To enroll into the system, the user presses the enroll button, which will capture the image. Before the image is captured, the current number of system users is checked; if the maximum number of users is met the new user cannot be enrolled. If the maximum hasn't been met, the user's face image is captured and is once again sent to flash memory and is stored there temporarily for calculation. The image and Eigen faces are all pulled back to the microcontroller to calculate the new user's "template", which is a short vector describing the user's correlation with the Eigen faces. The template is then compared with the previously stored templates; if the new template is too close to a previous template, the user cannot be enrolled. The "closeness" between two templates is defined as the cosine of the angle between them. If there are no matches, the new template is added to the database in flash memory to save it in case of a system reset.

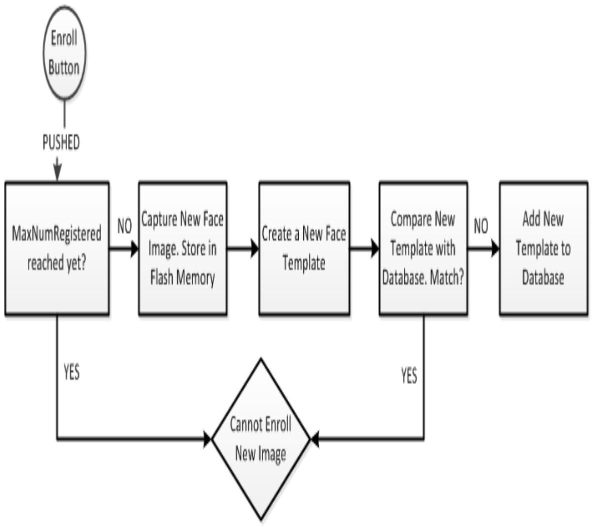

Enroll Button
PUSHED
MaxNumRegistered reached yet?
NO
Capture New Face Image. Store in Flash Memory
Create a New Face Template
Compare New Template with Database. Match?
NO
Add New Template to Database
YES
YES
Cannot Enroll New Image

Fig .3 Enrolling method.

III. Related Works

Now that reliable, accurate, and efficient face detection algorithms are available coupled with advances in embedded technologies; low-cost implementations of robust real-time face detectors can be explored. The most common target technologies are: pure hardware, embedded microprocessors, and configurable hardware.

Abbas Bigdeli *et al.*[5] explains how face recognition can be implemented in an embedded system using FPGA. His work focuses on the use of FPGAs as the embedded prototyping technology where the thread of execution is carried out on an embedded soft-core processor. Custom instructions have been utilized as a means of applying software/hardware partitioning through which the computational bottlenecks are moved to hardware

Matthew Turk and Alex Pentland [6] developed a near-real-time computer system that can locate and read subject's head and then recognize the person by comparing characteristics of the face to those known individuals. Their approach treats the face recognition problem as an intrinsically two-dimension recognition problem rather than requiring geometry of three dimensional geometry, taking advantage of the fact that faces are normally upright and thus may be described by a small set of 2-D characteristics views. Their method was based upon Principle Components Analysis (PCA).

Vincenzo Contiapproach *et al.[7]* showed how two biometrics can be fused and thus improving system accuracy and dependability.

Jain A *et al.[1]* described various features of finger print, he mentioned that fingerprint is composed of many ridges and furrows. These ridges and furrows present good similarities in each small local window, like parallelism and average width. However, shown by intensive research on fingerprint recognition, fingerprints are not distinguished by their ridges and furrows, but by Minutia, which are some abnormal points on the ridges. Among the variety of minutia types reported in literatures, two are mostly significant and in heavy usage: one is called termination, which is the immediate ending of a ridge, the other is called bifurcation, which is the point on the ridge from which two branches derive.

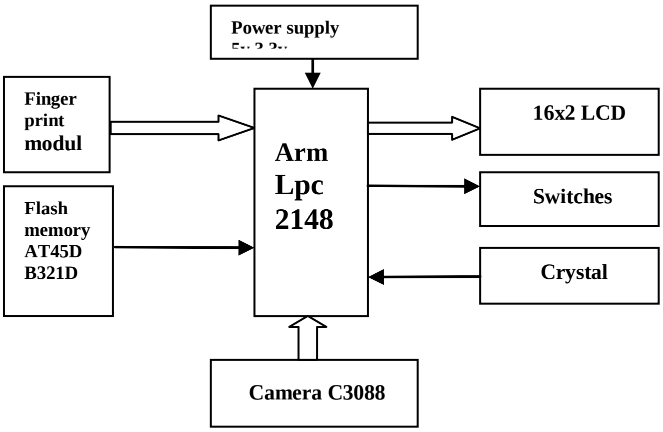


Fig. 4 Main components of the proposed system.

IV. Proposed work

Electronic voting machine has already been implemented and is in use in various countries including India, but these EVM's (electronic voting machine) does not provide necessary security like rigging etc. So to provide some security this project aims at using fingerprint module to eliminate rigging. Rigging is eliminated by use of finger-print, in which one candidate casts the votes of all the members or few amounts of members in the electoral list illegally. The fact that each individual in this planet has a unique finger-print is exploited and used in this voting machine so that rigging is eliminated. The main aim of this project is to implement face recognition in an embedded system using ARM. Face recognition has been implemented earlier in computer's which has a very powerful processor in it. So the main challenge in this project is to design an embedded system which can detect faces and match with the already stored images and give the result. In this project face recognition is for those who either don't have fingers or for person having deteriorated fingerprint. The project is expected to do finger print and face recognition and thus authenticate a person to vote. This project uses two biometrics i.e. fingerprint and face recognition using finger-print module and camera interfaced to ARM. The project makes use of MATLAB for converting image to Eigen faces once and later the system becomes stand alone and recognizes face on its own. The project demands the user to submit his Finger print at the polling booth. The project uses the Finger print technology and Embedded Systems to design this application. The main objective of this project is to design a system that asks the user to show his Finger print as an identity proof. The system reads the data from the Finger print and verifies this data with the already stored data in its database. If the details present in the database matches with the stored data, the system allows the person to enter into and poll his vote. If the details of the Finger do not match with the stored data, the system immediately activates the display and the security authorities can come and take the further action.

Background Math

[1] Training

The training portion of the system consists of creating the eigenfaces based on a set of training faces. The goal was to create as many eigenfaces as possible and keep the M eigenfaces that had the highest eigenvalues.

Given N training images, create a matrix where each column is a face vector of length 176 * 143 = 25168.

$$S = [\vec{F}_1\ \vec{F}_2\ \vec{F}_3\ \ldots \vec{F}_M] \qquad F = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{25167} \end{bmatrix}$$

A total of N=40 training faces was used to create the eigenfaces. Below are the raw images captured (before any processing) of some of the volunteers; not all of these faces were used to create the eigenfaces.

To create the eigenfaces we use the following algorithm:

1. Normalize each face

- Calculate the mean and standard deviation of each face. This is like brightness and contrast.

$$\mu_i = \frac{1}{25168}\sum_{j=0}^{25167} x_j \qquad \sigma_i = \sqrt{\sum_{j=0}^{25167}\left(x_j - \mu_i\right)^2}$$

- Normalize each image so that each face is closer to some desired mean and standard deviation

$$F_{norm_i} = (F_i - \mu_{des})\frac{\sigma_i}{\sigma_{des}} + \mu_i$$

- Calculate the mean image based on the M normalized images.

$$\Psi = \frac{1}{M}\sum_{i=1}^{M} F_{norm_i}$$

- Calculate the difference matrix using the new mean face.

$$\Phi_i = F_i - \Psi \qquad D = [\Phi_1\ \Phi_2 \ldots \Phi_M]$$

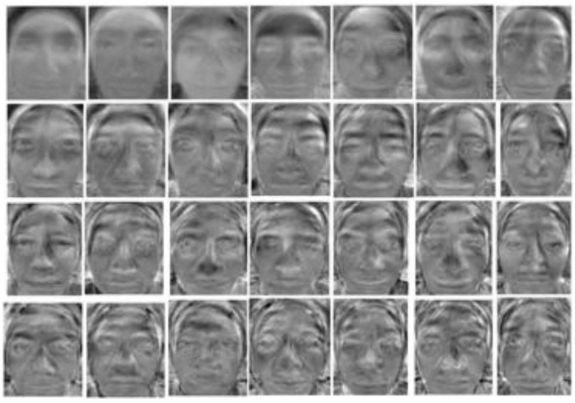
- Find the eigenvectors of D * D'. However, this is too large of a matrix so it cannot be done directly.
  - Find the eigenvectors v of D' * D.

$$(D^T D)v_i = \lambda_i v_i$$

  - Multiply the resulting eigenvector by D to get the eigenvectors of D * D'.

$$D(D^T D)v_i = D(\lambda_i v_i) \rightarrow (DD^T)(Dv_i) = (D\lambda_i)v_i \rightarrow (DD^T)\tilde{v}_i = \lambda_i \tilde{v}_i$$

3. Take the eigenfaces that correspond to the M eigenvectors with the highest value.
   Below are 28 of the 30 eigenfaces that was created. For enrolling purposes we use only the top 25 eigenfaces.



[2] Enrolling

The enrollment process consists of creating a user "template" for comparison when he tries to log in again later. The template is an M-length vector (M is the number of eigenfaces used to create the face space) that represents the correlation of the user's face with each eigenface. We use M = 25.

To create a user template from the 176x143 face image, we use the following algorithm:

- Follow the same steps as above to normalize the new face image to the desired mean and standard deviation used in the eigenface calculation.
- Create the "difference" face by subtracting the average face from the normalized new face
- Dot the "difference" face with each eigenface. The value of the ith dot product is the ith element of the template. T is the template, F is the difference face, and E is the eigenface.

$$T = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_M \end{bmatrix} \qquad d_i = F \cdot E_i$$

[3] Logging In

When a user attempts to log in, a new face template is created from the newly captured image as described above. This template is then compared with every other saved template. The measure of correlation we use is the cosine of the angle between the different templates. If the correlation between two templates is above the desired threshold then a "match" has been found. After testing, we decided to use 0.85 as a threshold. This was small enough to reduce false negatives while high enough to eliminate false positives.

To log in a user with their new 176x143 image, we used the following algorithm:

- Follow the same steps as the enrollment process to create a new template, T_new
- Calculate the correlation of T_new with all of the stored templates to find the number of matches.

$$\text{corr}_{new,i} = \frac{(T_{new} \cdot T_i)}{\|T_{new}\|_2 \|T_i\|_2}$$

[4] Results vs. Expectations

Our results were better than it was expected. I was able to show a reasonable login success rate (88%) with reasonable run time (15 seconds). For this I'm extremely satisfied. It was a little disappointing that the system was so sensitive to hair modifications, head tilt, and other sometimes unidentifiable variables. Some of these are limitations of the eigenface method.

If I was to repeat this project from scratch, I would do a few things differently. First, I would improve mechanical structure to include a forehead rest or some other head position normalizer. Second, I would use something other than serial dataflash. This could improve run time and would have prevented the data corruptions that plagued my last week. If I improved run time, I could use more eigenfaces, further improving our successful login rate and making the system more robust. I would have also liked to use more training faces, but it was difficult to find more than 50 people in a few days to train the system with.

V. Conclusion

This paper explored the possibility of making a portable face recognition system as used in the proposed electronic voting machine along with fingerprint identification to increase the security of voting machines.

References

[1] A. Jain, P. Flynn, and A. Ross, Handbook of Biometrics. New York: Springer, 2007.

[2] D. Ashok Kumar, T. Ummal Sariba Begum, International journal of innovative technology &creative engineering (ISSN: 2045-8711) vol.1 no.1 January 2011.

[3] Shaharam Mohammadi, Ali Frajzadeh A. Matching Algorithm of Minutiae for Real Time Fingerprint Identification System ,World Academy of Science, Engineering and Technology 60,2009

[4] R. Chellappa, C.L. Wilson, and S.Sirohey, "Human and machine recognition of faces: A survey," Proc. IEEE, vol. 83, pp. 705–740, 1995

[5] Abbas Bigdeli, Colin Sim, Morteza Biglari-Abhari and Brian C. Lovell, Australian Government Department of the Prime Minister and Cabinet funded project.

[6] Matthew Turk and Alex Pentland., the media laboratory Massachusetts Institute of Technology.

[7]

[8] Vincenzo Conti, Carmelo Militello, Filippo Sorbello, Member, IEEE, and Salvatore Vitabile, Member,

[9] H. Wechsler, P. Phillips, V. Bruce, F. Soulie, and T. Huang, Face Recognition: From Theory to Applications, Springer-Verlag, 1996.

[10] Y. M. Mustafah, A. Bigdeli, A. W. Azman, B. C. Lovell, Smart Cameras Enabling Automated Face Recognition in the Crowd for Intelligent Surveillance System.

[11] Shaharam Mohammadi, Ali Frajzadeh A. Matching Algorithm of Minutiae for Real Time Fingerprint Identification System ,World Academy of Science, Engineering and Technology 60,2009