

Face Recognition Using Line Edge Mapping Approach

Ibikunle F¹, Agbetuyi F.², Ukpere G^{2,*}

¹Department of CIT Engineering, Botswana Int'l University of Science & Technology, Botswana

²Department of Electrical and Information Engineering, Covenant University, Ota, Nigeria

*Corresponding author: godson.ukpere@gmail.com

Received August 03, 2013; Revised November 02, 2013; Accepted November 14, 2013

Abstract The research is based on development of an authentication system. Similar to these is the facial features authentication method which is a very new and unpopular method of authentication. The method is very unique with its operation as it doesn't require contact between the individual and the authentication device. The palm and retinal scanners have motivated the invention of the authentication system. Retinal scanners are contactless authentication methods which scans the venation in the retinal of the individual which is of course unique to human beings. The technology employed in this work uses picture frames from videos, detects facial features, and or matches the face to the respective individual's face features in the database. Authentication systems are used to identify or verify an individual as well as to distinguish the individual so far identified. This work develops an authentication system that operates with similar accuracy and speed of the human identification.

Keywords: *biometric, recognition system, authentication system*

Cite This Article: Ibikunle F, Agbetuyi F., and Ukpere G, "Face Recognition Using Line Edge Mapping Approach." *American Journal of Electrical and Electronic Engineering* 1, no. 3 (2013): 52-59. doi: 10.12691/ajeec-1-3-4.

1. Introduction

Technology advances have taken huge steps in our modern economy for authentication, validation and distinction between individuals. Some of these unique features possessed by individuals include finger-print, signatures, retinal scanning, as well as facial features. The technology employed uses picture frames from videos, detects facial features, and or matches the face to the respective individual's face features in the database. Authentication systems are used to identify or verify an individual as well as to distinguish the individual so far identified. Most authentication methods are based on Biometric information and this method however is partly biometric as it uses the facial features of an individual by picking of the light absorption properties of the individuals' face. One of the main aspects of face identification is its robustness. A face recognition system would allow a user to be identified by simply walking past a surveillance camera. Robust face recognition scheme require both low dimensional feature representation for data compression purposes and enhanced discrimination abilities for subsequent image retrieval. The representation methods usually start with a dimensionality reduction procedure, since the high dimensionality of the original visual space makes the statistical estimation very difficult and time consuming.

Similar to these is the facial features authentication method which is a very new and unpopular method of authentication. The method is very unique with its operation as it doesn't require contact between the

individual and the authentication device. The palm and retinal scanners have motivated the invention of the authentication system. Retinal scanners are contactless authentication methods which scans the venation in the retinal of the individual which is of course unique to human beings. Palm authentication also known as the Hand Geometry authentication method uses the size of the palm, the shape and sizes of the fingers. The Face Recognition method however operates like a human being, it simply sees a face, processes it, and tries to identify the individual just like how the brain operates.

Face recognition takes images of people, and returns the possible identity of that person. Face recognition systems are intended for use as a security system to find people in a crowd or deny access to a particular person from a sensitive area. Face authentication typically has a user position themselves in front of a camera, and then they enter their username and have the camera take an image from them. The image is compared to other images of the person. Based on this comparison the user is either granted access or denied.

This paper is organized as follows. Section I examined the background which includes introduction that gave insight and helpful hints on the subject matter. Section II gives detail review of technical and academic literature on previous work on the subject of facial recognitions methods and approaches. Section III is on the system design and implementation. Section IV gives the work simulation results and analysis. Section V carries out the proposed system performance evaluation. While section VI ends with the conclusion and recommendations for other possible investigations and improvements that could be made to the work in future.

2. Literature Review

Automated face recognition is a relatively new concept. Developed in the 1960s, the first semi-automated system for face recognition required the administrator to locate features (such as eyes, ears, nose, and mouth) on the photographs before it calculated distances and ratios to a common reference point which were then compared to reference data. In the 1970s, Goldstein, Harmon, and Lesk used 21 specific subjective markers such as hair color and lip thickness to automate the recognition. The problem with both of these early solutions was that the measurements and locations were manually computed. In 1988, Kirby and Sirovich applied principle component analysis, a standard linear algebra technique, to the face recognition problem [2]. This was considered somewhat of a milestone as it showed that less than one hundred values were required to accurately code a suitably aligned and normalized face image. In 1991, Turk and Pentland discovered that while using the eigenfaces techniques, the residual error could be used to detect faces in images— a discovery that enabled reliable real-time automated face recognition systems. Although the approach was somewhat constrained by environmental factors, it nonetheless created significant interest in furthering development of automated face recognition technologies. The technology first captured the public's attention from the media reaction to a trial implementation at the January 2001 Super Bowl, which captured surveillance images and compared them to a database of digital mugshots. This demonstration initiated much-needed analysis on how to use the technology to support national needs. The following are the methods and approaches previously used in facial recognition. These approaches produced results based on the years they were developed and deployed.

2.1. Eigen Faces

This is one of the most thoroughly investigated approaches to face recognition. It is also known as Karhunen-Loève expansion, eigen picture, eigenvector, and principal component. [1,2] used principal component analysis to efficiently represent pictures of faces. They argued that any face images could be approximately reconstructed by a small collection of weights for each face and a standard face picture (eigen picture). The weights describing each face are obtained by projecting the face image onto the eigen picture. In mathematical terms, eigen faces are the principal components of the distribution of faces, or the eigenvectors of the covariance matrix of the set of face images. The eigenvectors are ordered to represent different amounts of the variation, respectively, among the faces. Each face can be represented exactly by a linear combination of the Eigen faces. It can also be approximated using only the “best” Eigen vectors with the largest Eigen values. The best M Eigen faces construct an M dimensional space, i.e., the “face space”. The authors reported 96 percent, 85 percent, and 64 percent correct classifications averaged over lighting, orientation, and size variations, respectively. Their database contained 2,500 images of 16 individuals. As the images include a large quantity of background area, the above results are influenced by background. The authors explained the robust performance of the system

under different lighting conditions by significant correlation between images with changes in illumination.

2.2. Neural Networks

The attractiveness of using neural networks could be due to its non-linearity in the network. Hence, the feature extraction step may be more efficient than the linear Karhunen-Loève methods. One of the first Artificial Neural Networks (ANN) techniques used for face recognition is a single layer adaptive network called WISARD which contains a separate network for each stored individual. The way in constructing a neural network structure is crucial for successful recognition. It is very much dependent on the intended application. For face detection, multilayer perception and convolution neural network have been applied. Reference [3] proposed a hybrid neural network which combines local image sampling, a self-organizing map (SOM) neural network, and a convolutional neural network. The SOM provides a quantization of the image samples into a topological space where inputs that are nearby in the original space are also nearby in the output space, thereby providing dimension reduction and invariance to minor changes in the image sample. The convolutional network extracts successively larger features in a hierarchical set of layers and provides partial invariance to translation, rotation, scale, and deformation. The authors reported 96.2% correct recognition on ORL database of 400 images of 40 individuals. The classification time is less than 0.5 second, but the training time is as long as 4 hours. The research work in [4] used Probabilistic Decision-Based Neural Network (PDBNN) which inherited the modular structure from its predecessor- a Decision Based Neural Network (DBNN). The PDBNN can be applied effectively to: find the location of a human face in a cluttered image (Face detector); to determine the positions of both eyes in order to generate meaningful feature vectors (Eye localizer: and for Face recognizer. PDNN does not have a fully connected network topology. Instead, it divides the network into K subsets. Each subset is dedicated to recognize one person in the database. PDNN uses the Gaussian activation function for its neurons, and the output of each “face subset” is the weighted summation of the neuron outputs. In other words, the face subset estimates the likelihood density using the popular *mixture-of-Gaussian* model. The learning scheme of the PDNN consists of two phases, in the first phase; each subset is trained by its own face images. In the second phase, called the decision-based learning, the subset parameters may be trained by some particular samples from other face classes. PDBNN-based biometric identification system has the merits of both neural networks and statistical approaches, and its distributed computing principle is relatively easy to implement on parallel computer. It was reported that PDBNN face recognizer had the capability of recognizing up to 200 people and could achieve up to 96% correct recognition rate in approximately 1 second. However, when the number of persons increases, the computing expense will become more demanding.

In general, neural network approaches encounter problems when the number of classes (i.e., individuals) increases. Moreover, they are not suitable for a single model image recognition test because multiple model

images per person are necessary in order for training the systems to “optimal” parameter setting. Also long training time is required (4 hours) for Multilayer perception convolution neural network.

2.3. Graph Matching

Graph matching is another approach to face recognition. In [5], a dynamic link structure for distortion invariant object recognition which employed elastic graph matching to find the closest stored graph is presented. Dynamic link architecture is an extension to classical artificial neural networks. Memorized objects are represented by sparse graphs, whose vertices are labeled with a multi-resolution description in terms of a local power spectrum and whose edges are labeled with geometrical distance vectors. Object recognition can be formulated as elastic graph matching which is performed by stochastic optimization of a matching cost function. They reported good results on a database of 87 people and a small set of office items comprising different expressions with a rotation of 15 degrees.

The matching process is computationally expensive, taking about 25 seconds to compare with 87 stored objects on a parallel machine with 23 transputers. The technique was extended in [6] and matched human faces against a gallery of 112 neutral frontal view faces. Probe images were distorted due to rotation in depth and changing facial expression. Encouraging results on faces with large rotation angles were obtained. They reported recognition rates of 86.5% and 66.4% for the matching tests of 111 faces of 15 degree rotation and 110 faces of 30 degree rotation to a gallery of 112 neutral frontal views. In general, dynamic link architecture is superior to other face recognition techniques in terms of rotation invariance; however, the matching process is computationally expensive.

2.4. Geometrical Feature Matching Techniques

Geometrical feature matching techniques are based on the computation of a set of geometrical features from the picture of a face. The fact that face recognition is possible even at coarse resolution as low as 8x6 pixels when the single facial features are hardly revealed in detail implies that the overall geometrical configuration of the face features is sufficient for recognition. The overall configuration can be described by a vector representing the position and size of the main facial features, such as eyes and eyebrows, nose, mouth, and the shape of face outline. One of the pioneering works on automated face recognition using geometrical features was presented in [7]. Their system achieved a peak performance of 75% recognition rate on a database of 20 people using two images per person, one as the model and the other as the test image. The work in [8] automatically extracted a set of geometrical features from the picture of a face, such as nose width and length, mouth position, and chin shape. There were 35 features extracted from a 35 dimensional vector. The recognition was then performed with a Bayes classifier. They reported a recognition rate of 90% on a database of 47 people. The matching process utilized the information presented in a topological graphic representation of the feature points. After compensating

for different centroid location, two cost values, the topological cost, and similarity cost, were evaluated. The recognition accuracy in terms of the best match to the right person was 86% and 94% of the correct person's faces were in the top three candidate matches. In summary, geometrical feature matching based on precisely measured distances between features may be most useful for finding possible matches in a large database such as a Mug shot album. However, it will be dependent on the accuracy of the feature location algorithms. Current automated face feature location algorithms do not provide a high degree of accuracy and require considerable computational time.

2.5. Morphable Face Model

Morphable face model is based on a vector space representation of faces that is constructed such that any convex combination of shape and texture vectors of a set of examples describes a realistic human face. Fitting the 3D morphable model to images can be used in two ways for recognition across different viewing conditions:

Paradigm 1: After fitting the model, recognition can be based on model coefficients, which represent intrinsic shape and texture of faces, and are independent of the imaging conditions:

Paradigm 2: Three-dimension face reconstruction can also be employed to generate synthetic views from gallery probe images. The synthetic views are then Internationally transferred to a second, viewpoint-dependent recognition system.

More recently, the work in [8] combines deformable 3D models with a computer graphics simulation of projection and illumination given a single image of a person; the algorithm automatically estimates 3D shape, texture, and all relevant 3D scene parameters. In this framework, rotations in depth or changes of illumination are very simple operations, and all poses and illuminations are covered by a single model. Illumination is not restricted to Lambertian reflection, but takes into account specular reflections and cast shadows, which have considerable influence on the appearance of human skin. This approach is based on a morphable model of 3D faces that captures the class-specific properties of faces. These properties are learned automatically from a data set of 3Dscans. The morphable model represents shapes and textures of faces as vectors in a high-dimensional face space, and involves a probability density function of natural faces within face space. The algorithm presented in [9] estimates all 3Dscene parameters automatically, including head position and orientation, focal length of the camera, and illumination direction. This is achieved by a new initialization procedure that also increases robustness and reliability of the system considerably. The new initialization uses image coordinates of between six and eight feature points. The percentage of correct identification on CMU-PIE database, based on side-view gallery, was 95% and the corresponding percentage on the FERET set, based on frontal view gallery images, along with the estimated head poses obtained from fitting was 95.9%.

2.6. Line Edge Map Edge Information (LEM)

Line Edge Map Edge Information (LEM) is a useful object representation feature that is insensitive to

illumination changes to certain extent. Though the edge map is widely used in various pattern recognition fields, it has been neglected in face recognition except in recent work reported in [10]. Edge images of objects could be used for object recognition and to achieve similar accuracy as gray-level pictures. The above mentioned report made use of edge maps to measure the similarity of face images. 92% accuracy was achieved. Takács argued that process of face recognition might start at a much earlier stage and edge images can be used for the recognition of faces without the involvement of high-level cognitive functions. A Line Edge Map approach, proposed in [11] extracts lines from a face edge map as features. This approach can be considered as a combination of template matching and geometrical feature matching. The LEM approach not only possesses the advantages of feature-based approaches, such as invariance to illumination and low memory requirement, but also has the advantage of high recognition performance of template matching. Line Edge Map integrates the structural information with spatial information of a face image by grouping pixels of face edge map to line segments. After thinning the edge map, a polygonal line fitting process is applied to generate the LEM of a face. An example of a human frontal face LEM is illustrated in Figure 1.

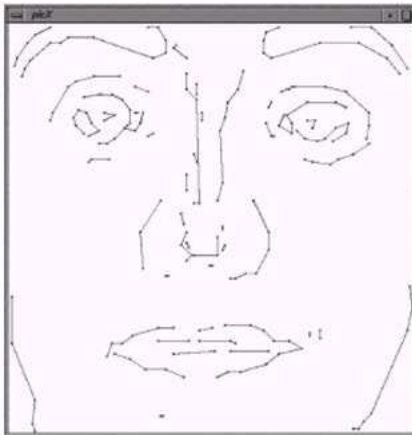


Figure 1. Line Edge Mapping (LEM)

The LEM representation reduces the storage requirement since it records only the end points of line segments on curves. Also, LEM is expected to be less sensitive to illumination changes due to the fact that it is an intermediate-level image representation derived from low-level edge map representation. The basic unit of LEM is the line segment grouped from pixels of edge map. A face pre-filtering algorithm is proposed that can be used as a pre-process of LEM matching in face identification application. The pre-filtering operation can speed up the search by reducing the number of candidates and the actual face LEM matching is only carried out on a subset of remaining models. The only limitation that existed in this approach has been countered by the specifications of modern computer systems and is no longer regarded as a limitation in any sense. Earlier systems had problems with storage space which made the size of each individuals face template (16 Kilobytes) bulky for obsolete computer systems. The parallel mutli-threaded processor operations of the application also posed a threat to old machines.

3. System Design and Development

The facial recognition approach used in developing this application is based on Line Edge Mapping method.

3.1. Line Edge Mapping

Line edge mapping works with the outline of the facial features, maps out the important points as a vector line, and saves the template. Line edge map has advantage over all other methods of face recognition, because it identifies the most facial features, it has a higher accuracy than others due to this effect as in [11]. LEM (Line Edge Mapping) consists of a series of line segments, it records only the endpoints of lines which further reduces it storage requirements. LEM matches two different images using LHD (Line Segment Hausdorff Distance). This is used in calculating the distance between lines using angular projection, parallelism, and perpendicularity of the two different lines to be matched and check if they meet the threshold for similarity.

The LHD is calculated as follows:

Let an array of LEMs be $A^L = [a_1^L, a_2^L, a_3^L, \dots, a_p^L]$

And another LEM be $B^L = [b_1^L, b_2^L, b_3^L, \dots, b_q^L]$

Then the LHD Vector be represented by $\bar{d}(a_i^L, b_j^L)$

$$\bar{d}(a_i^L, b_j^L) = \begin{bmatrix} \bar{d}_\square(a_i^L, b_j^L) \\ \bar{d}_\parallel(a_i^L, b_j^L) \\ \bar{d}_\perp(a_i^L, b_j^L) \end{bmatrix}$$

$\bar{d}_\square(a_i^L, b_j^L) = f(\square(a_i^L, b_j^L)) \rightarrow$ **angular line**

matching with tolerance: This matches different lines between two images if they are at a slight angle with each other with a tolerance marking the threshold of the similarity. $\Theta(a_i^L, b_j^L)$ represents smallest intersection angle between lines a_i^L and b_j^L . Function ‘f’ is the penalty factor that ignores the smaller angles and penalizes the greater ones.

$$f(\square) = \square^2 / W$$

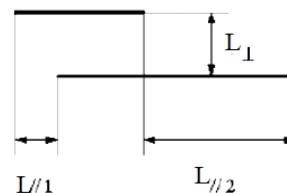
$\square =$ angle, W is determined during training

$\bar{d}_\parallel(a_i^L, b_j^L) = \min(L_{\parallel 1}, L_{\parallel 2}) \rightarrow$ **parallel line matching:**

This matches the parallelism of different lines and compares it in two images. $L_{\parallel 1}, L_{\parallel 2}$ are the two parallel lines, the ‘min’ function is the minimum distance between the edges of the lines.

$\bar{d}_\perp(a_i^L, b_j^L) = L_\perp \rightarrow$ **perpendicular line matching:**

This matches the perpendicularity of different lines and compares them between two images. L_\perp is the distance between perpendicular points. The representations of the above are represented in the figure below:



The distance between the two segments A,B can be calculated as follows:

$$d^2_{\square}(a_i^L, b_j^L) + d^2_{\parallel}(a_i^L, b_j^L) + d^2_{\perp}(a_i^L, b_j^L)$$

$$d(a_i^L, b_j^L) = \sqrt{d^2_{\square}(a_i^L, b_j^L) + d^2_{\parallel}(a_i^L, b_j^L) + d^2_{\perp}(a_i^L, b_j^L)}$$

3.2. Programming and Compilation

The application is developed using C#.NET. It is a Microsoft based programming language based on the Microsoft .NET Framework. It is a high level, fully object oriented programming language. The following are the packages required for the development of the coding:

- **Microsoft C#.NET Compiler:** This is the source code compiler for C#.NET. It is integrated into VS

2010 IDE in order to make building, compilation, debugging and publishing faster.

- **Visual Studio 2010 IDE** (Integrated Development Environment): An application software that compiles, debugs and build .NET related programming languages in a single package.
- **FaceSDK.NET DLL** (Face Software Development Kit Dynamic Link Library): This is a DLL that comprises of functions, delegates, classes, objects etc. that consists of Line Edge Map calculations and template extraction.
- **SQLite3.NET DLL** (Sequential Query Language Lite 3 Dynamic Link Library): This is a DLL that handles the creation, execution and data reading of SQL queries to a mobile database.

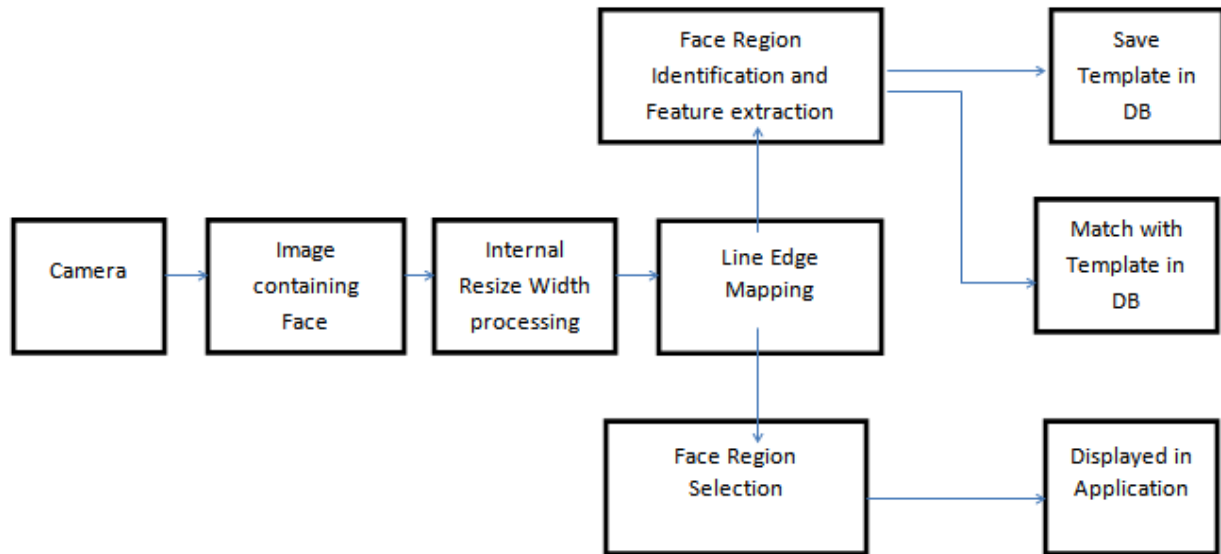


Figure 2. Block Diagram of Face Recognition Operation

3.3. Programming Series

3.3.1. Camera Image Streaming

The code block below shows how the camera images are captured and rendered into the **PictureBox** class. NET Control.

Each image frame from the camera is captured in a loop, and it sets the pictureBox's image to the camera image. This loop continues as long as the program is on and gives the pictureBox a video stream feel.

3.3.2. Database Connection String

SQLite database is used to store the users' information. The connection parameters below are used to create a new instance of an SQLite connection to DB.db (local database file in application root directory).

```
SQLiteConnection con = new SQLiteConnection
("Data Source=DB.db;Version=3;New=True;");
```

3.3.3. Face Template Extraction

The image from the camera is process in order to identify the presence of a face and dynamically create a template which can be stored or used for various purposes (recognition / authentication).

```
byte[] tempd = new byte[FSDK.TemplateSize];
```

```
FSDK.GetFaceTemplateInRegion(imageHandle, ref facePosition, out tempd);
```

The face template consists of a byte array into which the template parameters are stored with a length of 16384, which means it consists of 16384 bytes equivalent to 16KB (Kilobytes).

3.3.4. Face Template Storage

The information carried by the face template contains raw byte data precisely 16KB in length. Each face template of an individual to be added to the system is 16KB, and 10 Face Templates are taken in-order to increase the flexibility of recognition.

This information is saved as a BLOB data type in the database, against the individual's name, email and database primary key ID. The block of code below is used to create a new user record in the database.

```

publicstaticboolinsert_vals(stringname,stringemail,byte[] face)
{
    SQLiteConnection con = newSQLiteConnection("Data Source=DB.db;Version=3;");
    SQLiteCommand com = newSQLiteCommand("INSERT INTO 'users' ('fname','email','face','date') VALUES
(@name,@email,@face,@date)",con);
    com.Parameters.Add("@name", DbType.String, name.Length).Value = name;
    com.Parameters.Add("@email", DbType.String, email.Length).Value = email;
    com.Parameters.Add("@face", DbType.Binary, face.Length).Value = face;
    System.Windows.Forms.MessageBox.Show(sql.Message + "\n" + sql.StackTrace);
returnfalse;
}
}

```

3.3.5. Face Template Data Declaration

When the program loads as well as when a new user is added, public statically declared variable (An object array) is made to carry the data from the database. The object array contains four different arrays which are the “names string array”, “email string array”, “dateUTC long array”, “userid integer array” and the “face byte array”. Each of these arrays have the same array content count and are commonly match with the same array index i.e. names [i] should give a user’s name and equivalent value for his email in email [i] and face [i] for his/her saved face template, dateUTC [i] for the date of registration, userid [i] the database primary key ID for the user (i is an integer for the array index).

3.3.6. Facial Recognition and Authentication

After the variables have been populated, the program listens for the presence of a face in each of the image frames captured from the camera in real time. When it detects a face it then extracts the facial template, and tries to match it with any of the templates in the array of face templates it received from the database.

Below is the code block that actively extracts the face template from the image from the camera.

```

byte[] tempd = newbyte[FSDK.TemplateSize];//16384 in length
FSDK.GetFaceTemplateInRegion(imageHandle, ref facePosition, out tempd);

```

Below is the code block that attempts to match the face template with the existing templates held in the variables, with a False Acceptable Rate (FAR) set to 0.5% (0.005) which performs a very strict matching.

```

publicboolcheck_reg(byte[] ttp)
{
    float similarity = 0.0f;
    byte[] t1 = t;
    FSDK.MatchFaces(ref ttp, ref t1, ref similarity);
    float threshold = 0.0f;
returnfalse;
}

```

4. Simulation Results

The Face Detection System is integrated in the programmed software application. The major process that encompasses the Face Detection System is as follows:

- **Camera Selection:** The Application generates a list of all the USB (Universal Serial Bus) cameras connected to the computer system, and gives you the option of the camera to use for the operation.
- **Real Time Image Capturing (Virtual Video Stream):** The application virtualizes a video stream by simultaneously capturing picture frames from the camera, and displaying it in a picture box after it has been processed.
- **Image Processing:** Before the image is displayed in the picture box, it is processed by the software, by appending a rectangular skeleton, encapsulating a detected face, as well as the information of the individual underneath the rectangle, if the individual has already been registered.
- **Face angular and arbitrary rotations:** Enabling Arbitrary rotations processing extends default in-plane face rotation angle from the default value -15..15 degrees to -30..30 degrees. Enabling Angular rotations enable the software to detect a face even during in-plane rotation. Enabling any of these two parameters will reduce the application performance and increase CPU usage of the computer system.

Figure 3 shows the application main window. It is the first window form that is displays when the application is ran.



Figure 3. Main Application Window

Figure 4 below is the setting panel where parameters can be modified to suit the applications performance.

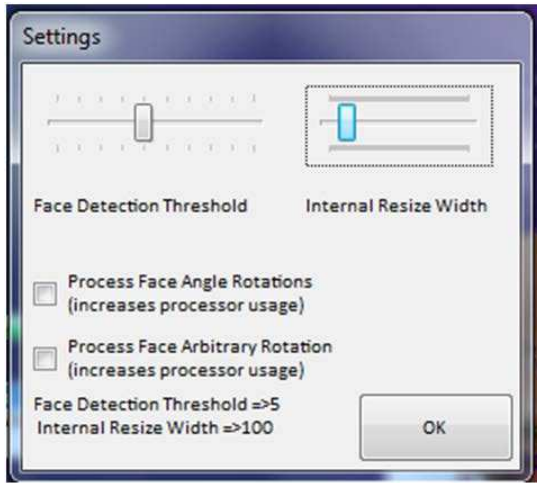


Figure 4. Facial Authenticator Pro Setting Window

Table 1. Facial Feature Detection Tests varying internal resize width of Face Detected

| Internal Resize Width (pixels) | Facial Feature Identification Probability (Avg. %) | Recognition Rate (Avg. %) | Permissible Distance from Camera (cm) |
|--------------------------------|--|---------------------------|---------------------------------------|
| 50 | 75 | 90 | 30 |
| 75 | 80 | 88 | 45 |
| 100 | 90 | 84 | 70 |
| 150 | 98 | 78 | 80 |
| 200 | 98 | 70 | 100 |
| 300 | 99 | 50 | 150 |
| 400 | 99 | 10 | 300 |

This is the Single face detection with respect to the preset **Internal Resize Width**. The internal resize width is the resolution the application uses to process the image from the camera. The lower it is, the higher the application’s performance, but a higher value will increase precision.

Table 2. Face Detection by varying Threshold Probability

| Threshold Value | Permissible Distance from Camera (cm) | Accuracy (%) |
|-----------------|---------------------------------------|--------------|
| 1 | 120 | 10 |
| 2 | 90 | 30 |
| 3 | 80 | 60 |
| 4 | 70 | 80 |
| 5 | 60 | 93 |
| 6 | 50 | 95 |
| 7 | 45 | 95 |
| 8 | 45 | 95 |
| 9 | 45 | 95 |
| 10 | 30 | 98 |

- **Threshold Value** controls the minimum falsifiable probability that a face exists in the current image captured from the camera.

After the face has been detected in the image frame from, the application extracts the facial features from the image, converts it into a template with format discussed earlier, and goes through a loop to try to match the template with an existing template in the database. The index used to detect the threshold for the similarity between two templates (of each step in the loop) is called the False Rejection Rate (FRR) or False Acceptance Rate (FAR) respectively [12].

The FRR is approximately inversely proportional to FAR but might tend to be exponential at the extreme values. FAR is the acceptable error rate, to which two faces are allowed to have similar features. FAR being an error should be kept as low as possible to increase accuracy and also taking the Rate of recognition into consideration. The FRR being reliability should be kept as high as possible to increase matching accuracy, and taking the rate of recognition into consideration. The FAR and FRR can be used alternatively without any form of setback or uncertainty, depending on representation preference. Table 3 and Table 4 shows the tests carried out by varying the values of the FAR and FRR of the application.

Table 3. Face Recognition by varying False Acceptance Rate

| False Acceptance Rate (FAR) (float [f]) | False Acceptance Rate (FAR) (%) | Matching Accuracy (approx. %) | Recognition Rate (%) |
|---|---------------------------------|-------------------------------|----------------------|
| 0.01 | 1 | 60 | 99 |
| 0.005 | 0.5 | 70 | 90 |
| 0.001 | 0.1 | 85 | 85 |
| 0.0005 | 0.05 | 95 | 70 |
| 0.00001 | 0.001 | 98 | 40 |
| 0.00005 | 0.005 | 99 | 10 |

NOTE that the FAR or FRR values are not changed in the application settings, but is hardcoded into the application source code before compilation.

Table 4. Face Recognition by varying FRR (False Rejection Rate)

| False Rejection Rate (FRR) (float [f]) | False Rejection Rate (FRR) (%) | Matching Accuracy (approx. %) | Recognition Rate (%) |
|--|--------------------------------|-------------------------------|----------------------|
| 1 | 100 | 99 | 30 |
| 0.5 | 50 | 75 | 65 |
| 0.4 | 40 | 70 | 70 |
| 0.2 | 20 | 50 | 85 |
| 0.1 | 10 | 45 | 88 |
| 0.01 | 1 | 20 | 99 |

5. Performance Evaluation

The performance evaluation of the system is carried out with a few variables and constants. The constant parameters in this context are: Illumination and Face Posture. While the varying parameters are: Internal Resize Width of the Image processing engine and False Acceptable Rate (maximum error rate) in face template matching.

5.1. Face Identification Probability

While testing for optimum face identification probability by varying the internal resize with, higher internal resize width gives a higher identification probability. These values are exponential proportion until resize width reaches about 300 pixels. This can be seen to produce optimum result for the face identification parameter. Although increasing the resize width increases the probability of identification, it also has an adverse effect on the performance of the system, creating unnecessary time lags in image processing. Figure 5 shows the relationship between Face Identification Probability and internal resize width.

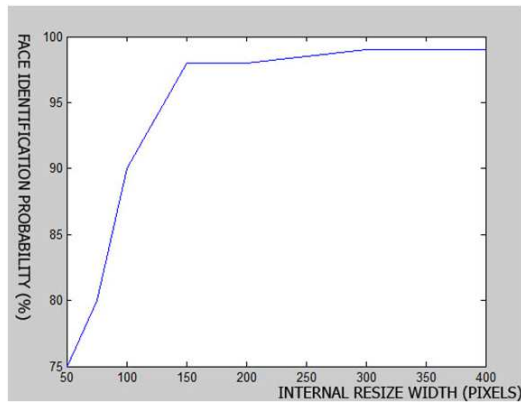


Figure 5. Relationship between Face Identification Probability and Internal Resize width.

5.2. False Acceptable Rate (FAR)

The False Acceptable Rate (FAR) is the error value (in %) to which two different face templates can be said to match. FAR and FRR are inversely proportional to each other and are used interchangeably in the design of the system. In-order words, when working with FAR, a low value will improve the matching accuracy, while a high FRR will improve the matching accuracy. In this scenario, FAR is to be used to derive the corresponding matching accuracy by varying the FAR value as a percentage. Figure 6 shows the graph of the relationship between the FAR and the matching accuracy.

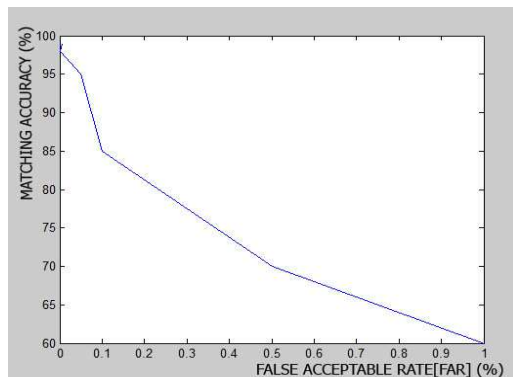


Figure 6. Graph of the relationship of FAR to the matching accuracy

Also by reducing the FAR to get a better matching accuracy, the system performance, with respect to the speed/rate of face recognition is reduced, and this creates a time lag in image processing. Optimum values of both FAR and internal resize width can be chosen based on the specification of the system that the application runs.

6. Conclusion

The major aim of this work is to design and construct a facial authentication system which can be used for the secondary or primary authentication of individuals with a much faster, modern and efficient authentication technique. Future work could be done to further improve efficiency, lighting condition limitation as well as the requirements for the installation. With these additional improvements, the standard could be raised for future facial authentication system. The overall efficiency of this authentication system is approximately 85% which can be improved by developing a more complex algorithm or by increasing the facial template features without adversely affecting the speed of operation. It can also be integrated with infrared cameras in order to increase its efficiency in poor illumination situations. Using the application on more compact devices such as a mobile device that are handy will increase the usage diversity of the application, replace common mobile user access methods like finger printing and manual PIN input.

References

- [1] L.Sirovich and M. Kirby, "Low-Dimensional procedure for the characterization of human faces" *Journal of Optical Soc. of Am.*, vol. 4, pp. 519-524, 1987.
- [2] M. Kirby and L. Sirovich, "Application of the Karhunen-Loève procedure for the characterisation of human faces" *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, pp. 831-835, Dec. 1990.
- [3] S. Lawrence, C.L. Giles, A.C. Tsoi, and A.D. Back, "Face recognition: A convolutional neural-network approach" *IEEE Trans. Neural Networks*, vol. 8, pp. 98-113, 1997.
- [4] S.H. Lin, S.Y. Kung, and L.J. Lin, "Face recognition/detection by probabilistic decision-based neural network," *IEEE Trans. Neural Networks*, vol. 8, pp. 114-132, 1997.
- [5] M. Lades, J.C. Vorbruggen, J. Buhmann, J. Lange, and M. Konen, "Distortion Invariant objectrecognition in the dynamic link architecture" *IEEE Trans. Computers*, vol. 42, pp. 300-311, 1993.
- [6] Wiskott and C. von der Malsburg, "Recognizing faces by dynamic link matching" *Neuroimage*, vol. 4, pp. 514-518, 1996.
- [7] T. Kanade, "Picture processing by computer complex and recognition of human faces," technical report, Dept. Information Science, Kyoto Univ., 1973.
- [8] R. Bruneli and T. Poggio, "Face recognition: features versus templates," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, pp. 1042-1052, 1993.
- [9] V. Blanz and T. Vetter, "Face recognition based on fitting a 3D morphable model," *IEEE Trans. On Pattern Analysis and Machine Intelligence*, vol. 25, no. 9, September 2003.
- [10] B. Takács, "Comparing face images using the modified hausdorff distance," *Pattern Recognition*, vol. 31, pp. 1873-1881, 1998.
- [11] Y. Gao and K.H. Leung, "Face recognition using line edge map," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 6, June 2002.
- [12] LuxandFaceSDK "Face Detection and Recognition Library" Developer's Guide 2011.