World Journal of Engineering Research and Technology
WJERT
www.wjert.org
SJIF Impact Factor: 4.326

DESIGN OF A BIOMETRIC DATABASE SYSTEM USING A SECURED NETWORK INTEGRATION

Idigo V. E.[1], Onwujei Augustine I.*[2], Okezie C. C.[3], and Okafor K. C.[4]

[1,2,3]Electronics and Computer Engineering, Nnnamdi Azikiwe University, Awka, Anambra State, Nigeria.

[4]Electrical and Electronics Engineering, Federal University of Technology, Owerri, Imo State, Nigeria.

Article Received on 22/06/2017 **Article Revised on 12/07/2017** **Article Accepted on 02/08/2017**

***Corresponding Author**
Onwujei Augustine I.
Electronics and Computer Engineering, Nnnamdi Azikiwe University, Awka, Anambra State, Nigeria.

ABSTRACT

This work used composite (hybrid) Principal component analysis (C-PCA) as a statistical procedure that uses an orthogonal transformation to convert a set of image observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. In this case, the number of principal components is less than or equal to the number of original variables. This transformation concept was applied in novel facial recognition biometric attendance register for accurate time monitoring and impersonation detection incidences. Emgu camera video dynamic link library (Emgu.CV.dll) and MySQLite-3 were used in deriving the PCA Eigen vectors. After deploying the system, various enrolments were carried out. Eigen verifications were also followed up to check for the image pattern matching. Verified images are certified while unverified images are flagged as illegal image templates. Also, the network deployment context was achieved based on carley graph Ethernet LAN environment. In this case, communication from the IP based cameras are relayed to secured database servers. This work showed the merits and application domain while concluding that with principal components image transformation, facial recognition systems can be used in complex surveillance systems.

KEYWORDS: Principal component analysis (PCA), Biometric System, Network Integration, Database, Emgu. CV. dll) and My SQLite-3.

INTRODUCTION

Interest in various types of biometric recognition systems is growing. In most intelligent applications, a combination of pattern recognition and image analysis could be used to generate a template that is stored in a database. For most security sensitive environments, database integration usually has security concerns. Biometric database security concerns involves the use of a broad range of information security controls to protect databases (potentially including the data, the database applications or stored functions, the database systems, the database servers and the associated network links) against compromises of their confidentiality, integrity and availability. It involves various types or categories of controls, such as technical, procedural/administrative and physical. It is a critical aspect of computer security, information security and risk management.[1] Like all tangible assets that have to be protected by a company, valuable information stored in its computer system database is probably the most precious of assets of the company that must be protected.[2]

Safety measures must be an integral part of any database, right from the start, at the inception and design phase. Modern approaches employed to assure the security of databases address security and protection defences at all levels: physical, network, host, applications and data, essentially, the first measure to be applied must start at the physical level and to then progress right through, reaching the data level at the other end. Initially, companies have had a rather simplistic approach, mainly due to primitive and rudimentary nature of earlier attacks, as well as the simple nature and construction of the then prevalent networks with very limited complexity if any, and did therefore focus on assuring security at the physical level. That then involved basic measures such as limiting access to locations that only authorized personnel may have access to data. More recently, due to the rapidly changing and increased size as well as complexity and expansion of company information systems, the Authentication, Authorization, Access, (AAA) type measures began to be used. Currently, the necessary security measures are now far more complex as these are meant to stop the highly sophisticated attacks from external attackers.[2]

Organizational database classical attacks are motivated by the following factors, viz

a. Databases are the mass of information which the company works with;

b. Databases can reveal private data by processing public data.

Hence, database security must address the following vulnerabilities

a. Theft and fraud;

b. Loss of confidentiality/privacy;
c. Loss of privacy;
d. Loss of integrity;
e. Loss of availability.

The hazards which make these things happen are due in large amount to deliberate human action. Natural type hazards or random events have an impact only on data integrity and availability. To ensure a minimum security of the databases, the following requirements must be satisfied as discussed below, viz: Physical integrity of databases; Logical integrity of databases; the integrity of each element which composes the database; Access control; User identification and Availability. Consequently, conventional systems now employ various methods of authenticating users and protecting communication messages in insecure networks.

Biometric based security systems can recognize a person by physiological characteristics like fingerprint, face, iris, palm, etc. or behavioural characteristics like signature, voice, gait, keystroke dynamics, etc. basically, biometrics can validate genuine user's presence thereby enhancing the authentication reliability.

Biometric traits offer three main benefits: (1) universality---every person possesses the biometric features, (2) uniqueness---it is unique from person to person, (3) performance stability---its properties remain stable during one's lifetime. These characteristics enable biometric-based authentication and identification systems to provide higher level protection than conventional knowledge based and token based system.[12] If the biometric authentication fails, an 'authentication failed' message will be returned.

In this research, a biometric scheme is proposed in the authentication process, the facial biometric are employed to work with a communication network for remote interactions. The basic idea is to transfer the locally matched Linear Discriminant Analysis **(**LDA) facial images with fused key binded template to the central server for biometric authentication. The computation demand on the central server is addressed using carley server interconnection procedure. Biometric cancellable keys are generated while minutia details are never exposed externally. Establishment of symmetric fused session keys further reduces vulnerabilities on the database server.

This research, proposes an efficient security firewall model designed for database server in secured computing environment. In this scheme, anomaly tracking for secure image authentication from the server will be considered. The multilevel security scheme proposed for the database will provide a secure and trustworthy authentication of remote database communication over insecure network. This will make it difficult for any invalid access communication with the image database of a mission critical system. This can be applied in data captures, repositories for forensic studies.

AIM

The main aim of this research is to develop a biometric recognition system based on Principal Component Analysis for image verification from a secured database network server clusters.

LITERATURE REVIEW

Overview on Database Security

Database security issues have been more complex due to widespread use and use of distributed client/server architecture as opposed to mainframes system. Databases are a firm main resource and therefore, policies and procedure must be put into place to safeguard its security and the integrity of the data it contains. Besides, access to the database has become more rampant due to the internet and intranets therefore, increasing the risks of unauthorized access.[23] The objective of database security is to protect database from accidental or intentional losses. These threats pose a risk on the integrity of the data and its reliability. Database managers in an organization identify threats and make policies that take action to mitigate any risks. Such actions include controls using passwords and username to control users who access the databases. The system created is called database management security system which keeps user details and allows access when provided with passwords and usernames.[23]

There are different threats to the database systems. Loss of availability means that data or systems cannot be accessed by any user. This most often arise from sabotage of the hardware, applications or networks system. This may halt the activities of the organization as well impede on the operation in the day to day activities of the organization.[23] Excessive privilege abuse is another method through which data can loss its integrity. When users are given too much privilege in the system database they abuse them for malicious purposes.
Another threat to database security is that of privileges elevation. This is when some user can convert extra privileges from ordinary user to administrator through taking database platform

software vulnerability. Denial of service is another problem in database security. This is a kind of an attack where data or network applications are targeted to avoid access to users. Often, the intention is to extort money. Other techniques that can be used in denial of service include data corruption, network flooding and resource overload.

Another threat to the problem of database insecurity is weak system and procedures for performing authentication. Weak authentication can result to attackers getting legitimate rights of user and then steal or change credentials. Some of the ways in which an attacker can hack in include use of social engineering, where passwords are requested through phone calls for maintenance purposes. Other includes brute force where the attacker does guess the passwords. Strong authentication is therefore required to address these challenges. Besides that, there is backup data exposure, where the storage media is left exposed leading to attacks. For example, tape and hard disks need to be secured well.[23]

Loss of data integrity can cause the data to be corrupted and invalid. This can result to delay in operations of the company as well as making wrong decisions which can affect the performance of the company.[23] This can only be restored through backup and recovery procedures.

Another issue is the loss of confidentiality. This is where the secrecy of crucial data in an organization is breached resulting to loss of confidentiality and eventual loss of competitiveness.[23]

Another threat to database system is the loss of privacy, theft or fraud is also common in firms such as banks. This occurs when personnel enter protected areas where databases are hosted and interfere with the systems. To prevent this threat, the firms should have controls on restricted areas as well install firewall to prevent people gaining unauthorized access to the database systems.[23] Other threats that can be detected are accidental losses which could result from malfunctioning systems and operating procedures.

Other forms of threats to databases could include inference theft. This is the process of sending queries deducing unauthorized information from legitimate sources. Identity theft is another form of threat to database. This is the situation where a person poses as another person and uses social security number to wipe out the details of the holders.[24]

There are several goals that are often targeted for database security issues. The first one is confidentiality. This relates to secrecy or privacy in terms of access by authorized subjects or processes. The second goal is to ensure that integrity is maintained and that means that data can only be changed by authorized subjects. Another goal is the availability of data. This is the need to maintain access to only authorized persons.

Method of Systems Analysis

System Methodology is the process of studying the existing system and identifying the basic information requirement. A set of cohesive, orderly and related techniques that influences how a system is developed is known as Methodology. In this work, two design methodologies were considered. They include waterfall methodology and evolutionary development. These methodologies were adopted because this meets with the iterative feedback mechanism required to achieve FRBAR so that as prototypes are developed, they can be either further improved on, or if any aspect of the new technology is proved to be unworkable, then they are replaced.

Waterfall Methodology

In Waterfall methodology, a rigid, linear and sequential schedule is followed and the goals for each phase are clearly defined and must be completed before moving onto the next one as shown in Figure 3.1.While the waterfall methodology has a number of advantages around the implementation discipline, it also has some disadvantages. The waterfall methodology is typically intended for large projects which this project is clearly not and more importantly the methodology does not have the ability to easily handle major change or upgrades late in the process which are envisaged in this system. Using waterfall model, the system requirement to achieve FRBAR is first analyzed, the functional, non-functional and hardware requirements are listed. Then the software required in developing FRBAR such as SQLite, .NET framework, and the C++ developing language and their requirements are analyzed. The outline of the proposed face recognition system is analyzed. Next, the program design stage comes in. In this stage, the block diagram and flowchart are drawn to illustrate the process of the program design. Then the program enters into its coding stage where the necessary codes needed to develop FRBAR are written. After FRBAR is created, it was tested for errors and other factors that could influence it. When all the necessary tests are completed, it is then deployed into full used and operation. The entire process is illustrated in Figure 1.

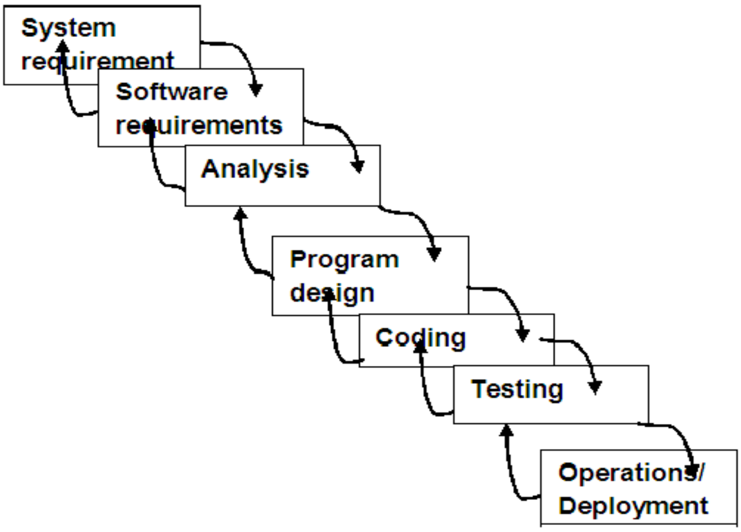


Figure 1: Traditional Waterfall Methodology.

Euclidian Distance

In mathematics, the Euclidean distance or Euclidean metric is the ordinary distance between two points that one would measure with a ruler, and is given by the Pythagorean formula. By using this formula as distance, Euclidean space (or even any inner product space) becomes a metric space. The associated norm is called the Euclidean norm. The Euclidean distance between point's p and q is the length of the line segment connecting them. Here we use the Euclidian distance to compare the training faces and input faces. The algorithm calculates the Euclidian distance between the input image and training face.

Known face is with minimum Euclidian distance and unknown face with largest distance. The input face is considered to belong to a class if Euclidean distance (εk) is below a threshold θε. Then the face image is considered to be a known face. If the input image is above the threshold, the face is determined as unknown.

The Euclidian distance between two instances X_i and X_j are given by

$$d(X_i , X_j) = \sqrt{\textstyle\sum_{r=1}^{n} (ar(Xi) - ar(Xj)^{2}} \qquad (1)$$

3.3 PCA Eigenface Algorithm

In developing the facial recognition algorithm, the first step was to obtain a set S with K face images. Each image is converted into a vector of size N and the training set is formed with K faces is given by

S_0= $(X_{1,} y_1, Z_1, \dots \dots \dots \dots \dots X_n, y_n, Z_n)^T$ = $\{\Gamma_1 + \Gamma_2 + \Gamma_3 \dots \dots \dots \dots \dots + \Gamma_n\}$ (2)

T_0= $(R_{1,} G_1, B_1, \dots \dots \dots \dots \dots R_n, G_n, B_n)^T$ (3)

This yields the shape and texture combined independently.

ii. Obtain the mean of training faces as

$\psi = \frac{1}{M} \sum_{n=1}^{m} \Gamma_n$ (4)

And it is subtracted from the original face as

φ_i=Γ_i- ψ (5)

Eigen vectors are finding such that

$\lambda_k = \frac{1}{M} \sum_{n=1}^{m} [U_k . \varphi_n]^2$ (6)

Where

U_k eigen vector and λ_k eigen vector

Covariance matrix is calculated as

C= $\frac{1}{M} \sum_{n=1}^{m} [\varphi_n . \varphi_n{}^T]$ (7)

The new face is transformed into its Eigenface components as

ω_k= $\mu_k{}^T$ $(\Gamma - \psi)$ (8)

Then the weight vector is formed by

$W\lambda^T$ = $[\varphi_1, \varphi_2, \varphi_3, \dots\dots. \varphi_n]$ (9)

The Euclidian distance between two instances X_i and X_j are given by

d(X_i , X_j) = $\sqrt{\sum_{r=1}^{n} (ar(Xi) - ar(Xj)^2}$ (10)

Then calculate the Euclidian distance between the input face and training faces, this is given by

$D = \varepsilon_K = \varphi_i$- ψ (11)

If D (ε_K) is below a threshold ϑ_K, then the face is concluded to be known and concludes the pattern matching. If the input is above the threshold, then it's not a valid face

Model Construction

Generalizing the facial modelling process on an individual person offers pairs of dimensional objects. The morphable face model is based on a vector space representation of faces. The database server for the facial templates can house as many captured or enroller faces as possible. Now, facial scans are stored in cylindrical coordinates relative to a vertical axis of the IP/web camera. The coordinates and texture values of all the n vertices of the reference face (n = 75; 972) are concatenated to form shape and texture vectors given by Equ 2 and 3. This yields the shape and texture combined independently:

$$S = \sum_{i=1}^{m} a_i S_i, T = \sum_{i=1}^{m} b_i T_i.$$

S and T are further represented as:

$$S = \bar{s} + \sum_{i=1}^{m-1} \alpha_i s_i, T = \bar{t} + \sum_{i=1}^{m} \beta_i T_i, \bar{s} = \frac{1}{m} \sum_{i=1}^{m} S_i, \bar{t} = \frac{1}{m} \sum_{i=1}^{m} T_i,$$

Where $\bar{S}$ is the mean shape and $\bar{t}$ is the mean texture.

3.5. Model Fitting

After the images are captured as shown in Figure 2, the image synthesis renders the new projected positions of vertices of Fisher component analysis 3D model, along with illumination and colour.

During the process of fitting the captured image both the shape, texture and their respective coefficients τ_i and β_i are optimized in the rendering process, which are concatenated into a vector ρ: the head orientation angles $\emptyset$, θ and γ, the head position (P_x; P_y) in the image plane, size s, color and intensity of the light sources L, as well as color constant, and gain and offset of colors. The primary goal in analysing the face vector is to minimize the sum of square differences over all facial variations in the input image and the symmetric reconstruction. In Figure 3.4, the goal of the fitting process is to find shape and texture coefficients τ_i and β such that rendering R_ω produces an image I_{model} that is as similar as possible to I_{input}. Figure 3.4 depicts a training image with automatically marked feature points from the database server. The marked feature points have been converted to triangles to create a face mask from which texture information can be gathered. Points line only on the eyebrows, around the eyes, lips and chin.

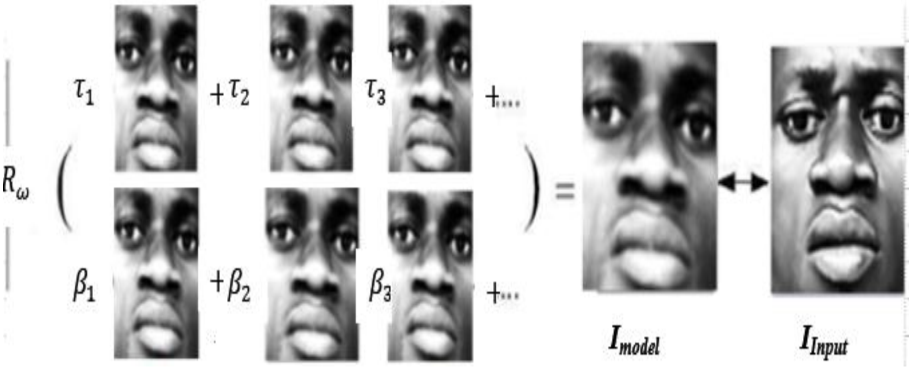


Figure 2: Facial Process fitting for Eigenvector (Author's schema).

3.6. Design Implementation

The design of a biometric database system using secured network integration was implemented considering the following subsystems, viz:

a. The Biometric Attendance Register (BAR)

b. The Network integration involving firewall security and server carley graph interconnect.For the BAR, Figure 2 shows the system implementation flowchart following previous discussions on PCA and Eigen vector images. In the design, upon deploying the IP camera and starting the application, the database which has been created is loaded. The Eigen vector algorithm is now instantiated to search for student records. Once a student's face is captured alongside with the relative data record, the facial analysis is enrolled and the datasets is stored in the database. On verification, once the system fails to locate the trained face, an error flag is displayed showing an illegal entity. Figure 2 to 5 describes the flowchart of the Proposed Biometric Attendance Register (BAR).

From Figure 2, once the database is loaded, enrolment can proceed using the LDA, PCA and EGBM. Afterwards, the image is added and the student information is enrolled also in this phase.

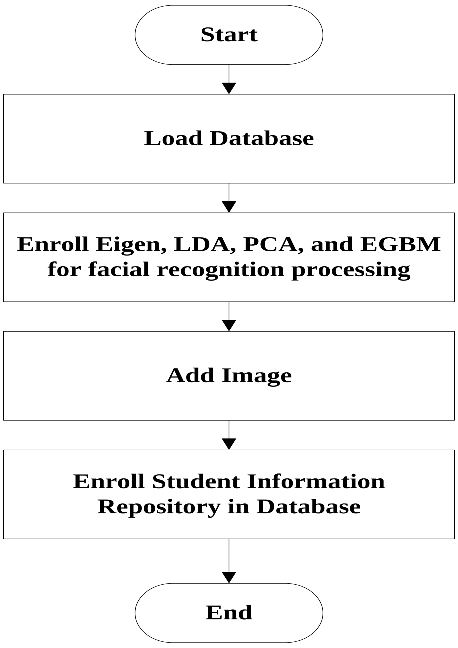

Start
Load Database
Enroll Eigen, LDA, PCA, and EGBM for facial recognition processing
Add Image
Enroll Student Information Repository in Database
End

Figure 3: Phase 1-Enrollment Module Flow diagram.

Figure 3 depicts the second phase where the training module uses the training algorithm to train the captured facial record. Within 5seconds, this is completed and tagged.

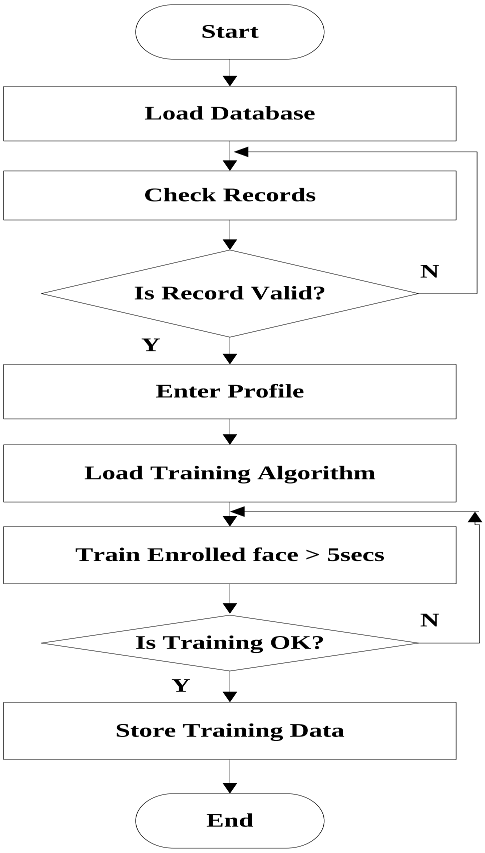


Figure 4: Phase-2 Training Module Flow diagram.

4. System Design and Analysis

4.1 Development Languages

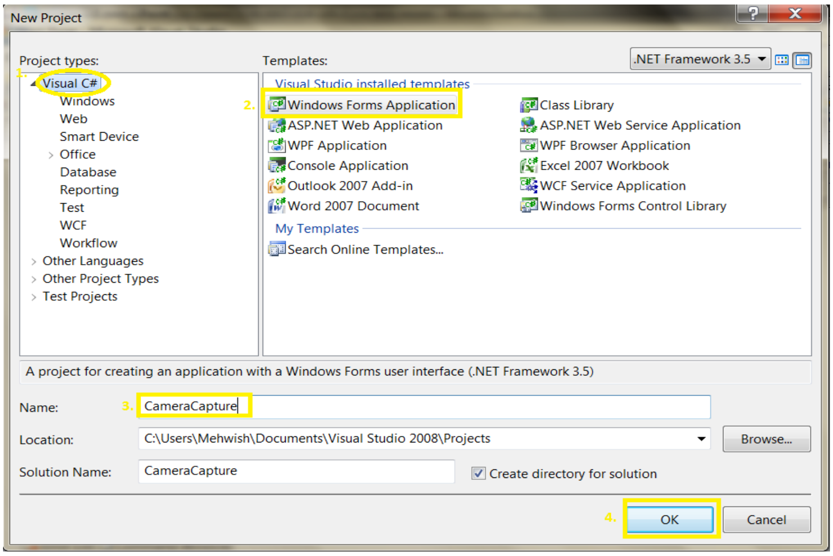
The software for the system integration was written in C++. For the main user GUI, it was decided to use Visual studio 2010. C++ is preferred to other languages due to a pool of reasons: C++ is pure object-oriented, C++ is more type safe, memory leakage problem is reduced, the assembly concept solves the versioning control problem well, ease-to-development, the rich class library makes many functions easy to be implemented, good support for distributed system, etc.

4.2 Microsoft Visual Studio

The Microsoft visual studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs for Microsoft Windows, as well as web sites, web applications and web services. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code.[25] Visual Studio does not support any programming language, solution or tool intrinsically; instead, it allows the plugging of functionality coded as a VSPackage. When installed, the functionality is available as a Service. The IDE provides three services: SVsSolution, which provides the ability to enumerate projects and solutions; SVsUIShell, which provides windowing and user interface functionality (including tabs, toolbars and tool windows); and SVsShell, which deals with registration of VSPackages. In addition, the IDE is also responsible for coordinating and enabling communication between services.[25] This supports various languages, but VB.NET was leveraged in this research. Object oriented programming model was adopted in this case were features such as Classes and Objects, Inheritance, and Polymorphism, Encapsulation were used in developing the biometric database GUI model. The step by step approach for the implementation is detailed below.

STEP-1: open visual Studio 2007 and select File-> New->Project.

STEP-2: in the Visual C# Project menu, Select "Windows Forms Application" and name the project "Facial_CameraCapture" as in figure below, and Click "OK"



New Project

Templates: .NET Framework 3.5

A project for creating an application with a Windows Forms user interface (.NET Framework 3.5)

Name: CameraCapture

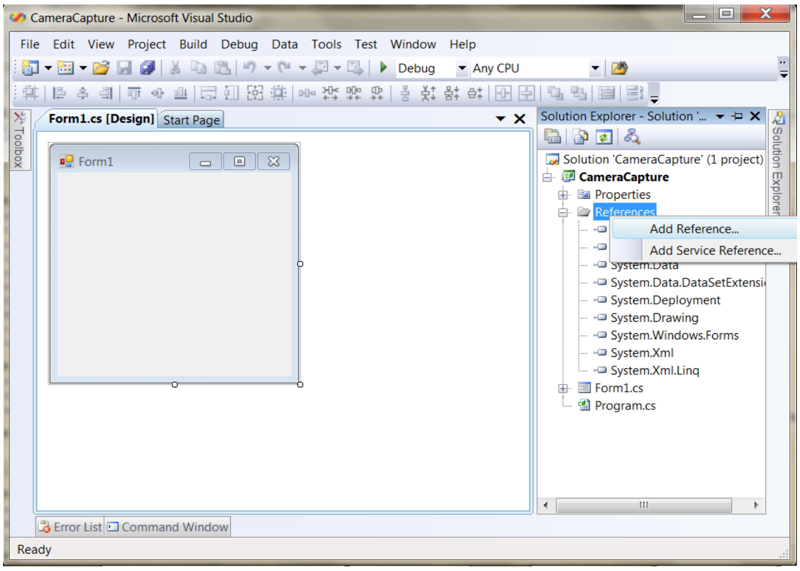
Location: C:\Users\Mehwish\Documents\Visual Studio 2008\Projects

Solution Name: CameraCapture

☑ Create directory for solution

Figure 5: VB.Net Framework Application Creation IDE.

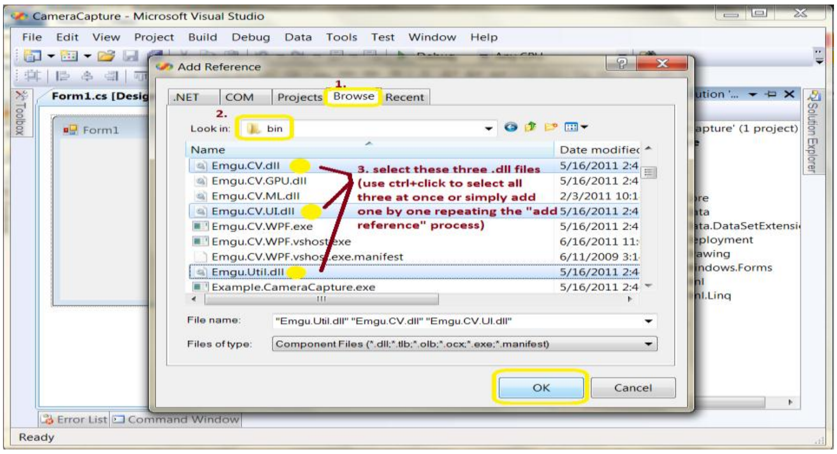
STEP-3: Emgu References was added to the project. Right-click project's "References" in Solution explorer, and select "Add Reference". As shown in Figure 5



CameraCapture - Microsoft Visual Studio

Figure 6: Emugu Reference Addition.

STEP-4: Select the Browse tab in the window that pops up, go to EmguCv's bin folder as in Level-0 tutorial, and select the following three .dll files (Emgu.CV.dll, Emgu.CV.UI.dll and Emgu.Util.dll) click OK to continue. Figure 6: illustrates this dynamic link libraries.



CameraCapture - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Test Window Help
Add Reference
.NET COM Projects Browse Recent
OK Cancel
Error List Command Window
Ready

1\.

2. Look in: bin

Name	Date modified
Emgu.CV.dll	5/16/2011 2:4
Emgu.CV.GPU.dll	5/16/2011 2:4
Emgu.CV.ML.dll	2/3/2011 10:1
Emgu.CV.UI.dll	5/16/2011 2:4
Emgu.CV.WPF.exe	5/16/2011 2:4
Emgu.CV.WPF.vshost.exe	6/16/2011 11:
Emgu.CV.WPF.vshost.exe.manifest	6/11/2009 3:1
Emgu.Util.dll	5/16/2011 2:4
Example.CameraCapture.exe	5/16/2011 2:4

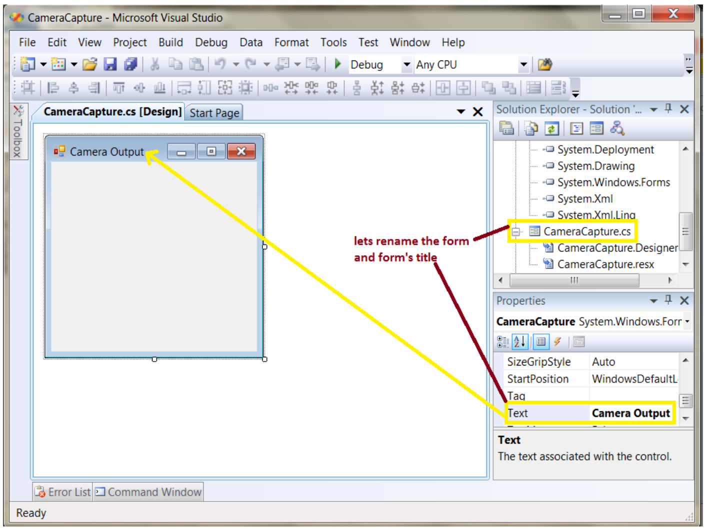
3. select these three .dll files (use ctrl+click to select all three at once or simply add one by one repeating the "add reference" process)

File name: "Emgu.Util.dll" "Emgu.CV.dll" "Emgu.CV.UI.dll"

Files of type: Component Files (*.dll;*.tlb;*.olb;*.ocx;*.exe;*.manifest)

Figure 7: Emugu Dynamic link library interfaces.

STEP-5: We now, rename Form1**.**cs to facial CameraCapture.cs and change its Text Field to "Camera Output" shown in Figure 7.



lets rename the form and form's title

CameraCapture System.Windows.Forr

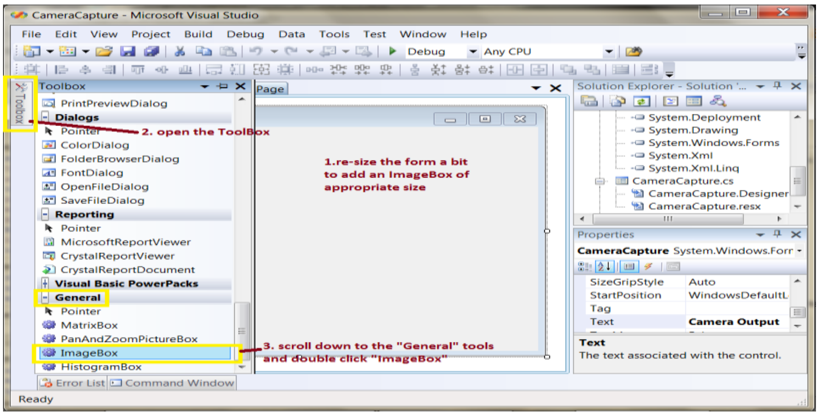
SizeGripStyle	Auto
StartPosition	WindowsDefaultL
Tag	
Text	Camera Output

Text
The text associated with the control.

Figure 8: Camera Capture naming.

STEP-6: Add EmguCv Tools to the .NET Visual Studio**,** since this work will be using those tools, such as ImageBox, in facial biometric design.

STEP-7: After adding EmguCv tools to the Toolbox, as in fig below, just re-size the form and follow the instruction in this fig to add an EmguCv ImageBox to the form created.



1.re-size the form a bit to add an ImageBox of appropriate size

2. open the ToolBox

3. scroll down to the "General" tools and double click "ImageBox"

CameraCapture System.Windows.Forr

SizeGripStyle Auto

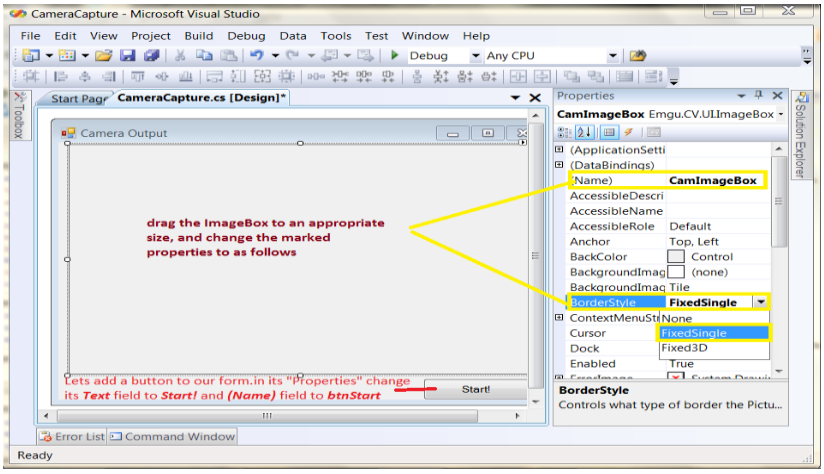
StartPosition WindowsDefaultL

Tag

Text Camera Output

Text
The text associated with the control.

Figure 9: Image Box Interfaces.


CameraCapture - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Test Window Help
Start Page
CameraCapture.cs [Design]*
Camera Output
Start!
Error List Command Window
Ready
Properties
Solution Explorer

drag the ImageBox to an appropriate size, and change the marked properties to as follows

Lets add a button to our form.in its "Properties" change its *Text* field to *Start!* and *(Name)* field to *btnStart*

CamImageBox Emgu.CV.UI.ImageBox

(ApplicationSetti

(DataBindings)

(Name) CamImageBox

AccessibleDescri

AccessibleName

AccessibleRole Default

Anchor Top, Left

BackColor Control

BackgroundImag (none)

BackgroundImag Tile

BorderStyle FixedSingle

ContextMenuSt None

Cursor FixedSingle

Dock Fixed3D

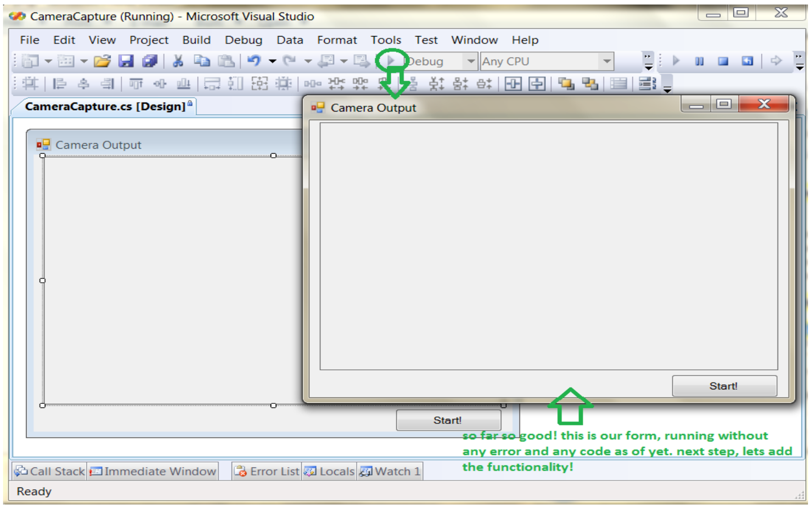
Enabled True

BorderStyle
Controls what type of border the Pictu...

Figure 10: Camera Capture design for boarder style.

STEP-9: Debug and Save Project

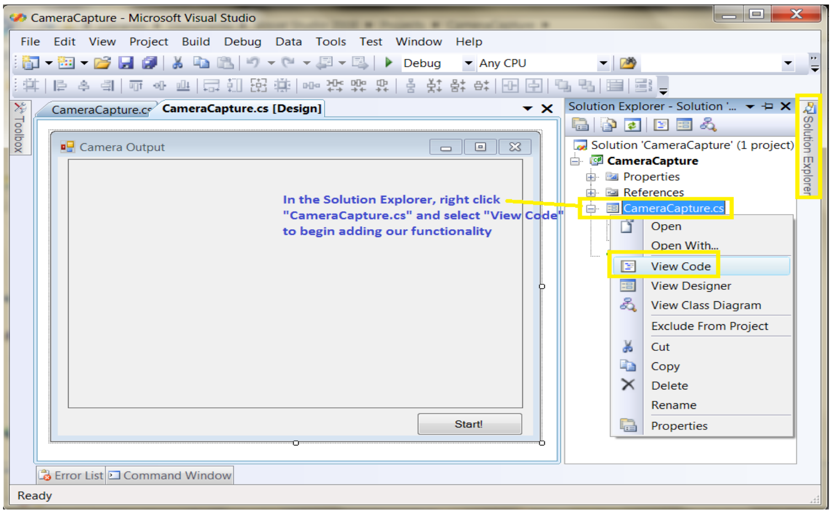
At this stage, the user interface design is ready while coding the objects. Also project debugging was carried out for successful compilation. The form is shown in Figure 11.


CameraCapture (Running) - Microsoft Visual Studio
File Edit View Project Build Debug Data Format Tools Test Window Help
Debug
Any CPU
CameraCapture.cs [Design]*
Camera Output
Start!
Camera Output
Start!
Call Stack Immediate Window Error List Locals Watch 1
Ready

so far so good! this is our form, running without any error and any code as of yet. next step, lets add the functionality!

Figure 11: Camera Output capture.

STEP-10: To begin coding, view code behind Camera Capture.cs as in figure 12.



In the Solution Explorer, right click "CameraCapture.cs" and select "View Code" to begin adding our functionality

Figure 12: Camera Coding Window.

STEP-11: In this phase, the code is finally saved. Actually, the Emgu CV references, were added using the directives to the existing ones:

using Emgu.CV;

using Emgu.CV.Structure;

using Emgu.Util;

Now inside the class, the work declared the following global variables just above the public CameraCapture() initializer

//declaring global variables

private Capture capture; //takes images from camera as image frames

private bool captureInProgress; // checks if capture is executing

Add a user defined function to the code and name it Process Frame() as shown below. In this function we'll create an EmguCv type image called ImageFrame. Then capture a frame from camera and save it in "ImageFrame"(line 1). And then this is loaded this into CamImageBox to show it to the user(line 2)

private void ProcessFrame(object sender, EventArgs arg)

{Image<Bgr, Byte> ImageFrame = capture.QueryFrame(); //line 1

CamImageBox.Image = ImageFrame; //line 2}

To Display an EmguCV image in Windows Form Picture Box, use this code instead of the one above:

//Show the image in Windows Form PictureBox called "pictureBox1"

pictureBox1.Image = ImageFrame.ToBitmap();

The start buttons are node coded from behind. By using the design View of ameraCapture.cs and double click the Start button recently added, it will take the developer back to the code view with an empty function of button click event as follows:

private void btnStart_Click(object sender, EventArgs e) {}

It was expected that when the Start button is pressed then camera should start working and the image stream should be visible in our ImageBox. In case the capture was already created (i.e once the application had begun) then now it will do either of the following based on value of captureInProgress:

when captureInProgress = true

then Pause the capture when image is acquired. This is done by the code: Application.Idle -= ProcessFrame; //ProcessFrame() will be called here to hold its job when captureInProgress = false

then **Start** the capture when btnStart is pressed. this is done by the code : Application.Idle += ProcessFrame; //ProcessFrame() will be called here to resume its job. Therefore into the btnStart_Click() function, add the following code:

#region if capture is not created, create it now

if (capture == null)

{try { capture = new Capture();}catch (Null Reference Exception excpt){ MessageBox.Show(excpt.Message);} }

#endregion

if (capture != null)

{ if (captureInProgress) { //if camera is getting frames then stop the capture and set button Text

// "Start" for resuming capture btnStart.Text = "Start!"; // Application.Idle -= ProcessFrame;}

else

{ //if camera is NOT getting frames then start the capture and set button// Text to "Stop" for pausing capture btnStart.Text = "Stop"; Application.Idle += ProcessFrame;}captureInProgress = !captureInProgress;}

Finally, this work added the following function to the code which takes care of closing the application in a safe way.

private void ReleaseData(){if (capture != null)capture.Dispose();}

Step 12: Error handling

5.1. Summary of Achievements

This research focused on the design of a secured flexible and efficient method for recognition, known as Principal Component Analysis was leveraged while ensuring the weighted capabilities of LDA and EGBM. Here, the work took all the images in training set as a linear combination of weighted Eigen vectors. The system receives the input face and it is recognized from the training set. Recognition is done by finding the Euclidean distance between the input face and our training set. The system has been simulated using V.B.Net framework and it shows appreciable result and faster detection rate. The approach is definitely simple, easy and faster to implement. Also, the performance of various PCA-based face recognition techniques was characterized using distinct design flowcharts. This work observed that that the performance of PCA is efficient in the cases when sum of Euclidean distance is taken as distance classifier. Facial image equalization is applied in the PCA concept ensuring that the number of Eigen faces is equal to the number of images in the database. The proposed PCA-based Face recognition system combines all of eigenvectors LDA, elastic bunch graphing features to form a face recognition System. This work argues that system outperforms the classical models though it can still be adapted for a multimodal architecture in which the detection of a person is performed on the basis of face and the facial expression values and fingerprint images. The user will have to provide the input in the form of face image and the comparison will be performed on both the face and the facial expression images.

REFERENCES

1. Online:http://en.wikipedia.org/wiki/Database_security/Retrievd 28th August, 2014.
2. Emil B., “Database Security-Attacks and Control methods, Journal of Applied Quantitative methods, winter, 2009; 4(4): 449-454.
3. Kai Xi, Tohari Ahmad, Fengling Han and Jiankun Hu, “A fingerprint based bio-cryptographic security protocol designed for client/server authentication in mobile computing environment, Security and Communication Networks”, Security Comm. Networks, 2010.
4. Rivest RL, Shamir A, Adleman L. A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 1978; 21(2): 120-126.
5. Miller V. Use of elliptic curves in cryptography, CRYPTO, 1985; 85.
6. Koblitz N. Elliptic curve cryptosystems, Mathematics of Computation, 1987; 48: 203-209.

7. Ahmad T, Hu J, Han S. Efficient Mobile Voting System Security Scheme based on Elliptic Curve Cryptography, International Workshop on Intelligent Decision Support Systems and Applications in Networked and Distributed Systems, IEEE 3rd International Conference on Network& System Security (NSS09), Gold Coast, Australia, October, 2009; 19-21.
8. Han F, Hu J, Yu X, Feng Y, Zhou J. A novel hybrid crypto-biometric authentication scheme for ATM based banking applications, IAPR International Conference on Biometrics (ICB2006), Hong Kong China, 5-7 January, 2006. Published at Lecture Notes in Computer Science, Springer, 2005; 3832/2005: 675-681.
9. Hu J, and Han F. A pixel-based scrambling scheme for digital medical images protection. Journal of Network and Computer Applications, Elsevier, 2009; 32: 788– 794.
10. Han F, Hu J, Yu X, Wang Y. Fingerprint images encryption via multi-scroll chaotic attractors. Applied Mathematics and Computation, Elsevier, 2007; 185: 931–939.
11. Han F, YuX, Feng Y, Hu J. On multi-scroll chaotic attractors in hysteresis-based piecewise linear systems. IEEE Transactions on Circuits and Systems-II, 2007; 54(11): 1004–1008.
12. Maltoni D, Maio D, Jain AK, Prabhakar S. Handbook of Fingerprint Recognition. Springer-Verlag: New York, 2003.
13. Soutar C, Roberge D, Stojanov SA, Gilroy R, Vijaya Kumar BVK. Biometric encryption - enrollment and verification procedures. Proceedings of SPIE, Optical Pattern Recognition IX, 1998; 3386: 24-35.
14. Soutar C, Roberge D, Stojanov SA, Gilroy R, Vijaya Kumar BVK. Biometric encryption. In ICSA Guide to Cryptography Nichols RK (ed.). McGraw Hill, New York, 1999.
15. Teoh A, Goh A, Ngo D. Random multispace quantization as an analytic mechanism for biohashing of biometric and random identity inputs. IEEE Transactions on Pat- tern Analysis and Machine Intelligence, 2006; 28(12): 1892-1901.
16. Savvides M, Vijayakumar B. Cancellable Biometric Filters for Face Recognition. Proceedings of IEEE International Conference Pattern Recognition, Cambridge, UK, August 2004; 3: 922-925,
17. Ratha N, Chikkerur S, Connell J, Bolle R. Generating Cancelable Fingerprint Templates. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007; 29(4): 561-572.
18. Teoh A, Toh K, Yip W. 2N Discretisation of Bio Phasor in Cancellable Biometrics, Proceedings of Second International Conference on Biometrics, Seoul, South Korea, 2007; 435-444.

19. Dodis Y, Ostrovsky R, Reyzin L, Smith A. Fuzzy extractors: how to generate strong keys from biometrics and other noisy data. SIAM Journal of Computing, 2008; 38(1): 97-139.
20. Juels A, Sudan M. A Fuzzy Vault Scheme. In Lapidoth A, Teletar E (eds). Proceedings of IEEE International Symposium on Information Theory, 2002; 408.
21. Uludag U, Pankanti S, Jain AK. Fuzzy vault for fingerprints, Proceedings of Audio- and Video-based Biometric Person Authentication. Rye Town: USA, 2005; 310-319.
22. Xi K, Hu J. Biometric mobile template protection: a composite feature based fingerprint fuzzy vault, IEEE International Conference on Communications, Dresden, Germany, 2009.
23. P, Singh, "Database management system concept V.K (India) Enterprises, 2009.
24. Kumar et al Managing Cyber threats: Issues, Approaches and Challenges Springer Publishers, 2005.
25. Msdn.mirosoft.com "Visual Studio Development Environment Model". MSDN. Microsoft. Retrieved, 2008-01-01.