THE JOURNAL OF COMPUTER SCIENCE AND ITS APPLICATION
AN INTERNATIONAL JOURNAL OF THE NIGERIA COMPUTER SOCIETY (NCS)
Vol. 19 No.2, December, 2012

nigeria computer society
www.ncs.org.ng.

AUTOMATED STUDENTS' ATTEDANCE TAKING IN TERTIARY INSTITUTION USING HYBRIDIZED FACIAL RECOGNITION ALGORITHM

*Akinnuwesi Boluwaji A. and Agagu Tosin T.
Department of Information Technology, Bells University of Technology, Ota, Ogun State Nigeria
(e-mail: boluakinnuwesi@bellsuniversity.edu.ng, temitop444@yahoo.com)

ABSTRACT

Students' identity management in tertiary institutions is one of the means of controlling access to the resources and services of the institutions, thus appropriate restrictions are placed on them. Among the services rendered in tertiary institutions are delivering of lectures and conducting examination for recognized students. In this paper facial recognition system is considered as an automatic tool to control, monitor and keep track of students' attendance at lectures and examination. The prototype developed in this research was customized for students' attendance taking at lectures in tertiary institutions with the view to eliminating the traditional manual method of attendance taking which has always been by writing and signing on attendance sheets and the lecturers keeping stock of the times the students sign the attendance sheet for lectures. Different algorithms at different stages of facial recognition were studied in order to develop a hybridized facial recognition algorithm. The algorithm developed was implemented using C# and C bindings in a dynamic link library format. Structured Query Language (SQL) lite and Microsoft Access were used to develop the relational databases. The prototype system automatically identifies and verifies a person from a digital image or video source and mark the person present provided the face matches the face registered in the institution student database. It was tested with some real life faces and data from Bells University of Technology, Ota, Nigeria. With the use of computer tools (hardware and software) for processing, the performance of the prototype was observed to be better than the existing manual system in terms of speed, accuracy, discovering of impersonation, ability to process large volume of data and ease of use by users.

Keywords: Face, Facial recognition, Prototype, Database system, Attendance, Tertiary Institution

1.0 INTRODUCTION

Face recognition is one of the important biometric methods; it deals with automatically identifying or verifying a person from a digital image or video source by comparing selected facial features. It is a form of identity access management and access control. Moreover, face recognition is considered a passive and non-intrusive approach to verifying and identifying people [1]. Though there are other forms of identification such as password, PIN (personal identification number), fingerprints and iris but in some cases it is better to have an identification approach that is closer to the way human beings recognize each other [2] and this informed the application of the proposed hybridized face recognition algorithm to students' attendance taking in tertiary institutions.

Attendance in institutions has traditionally been taken manually by writing and signing on attendance sheets and the lecturer keeping stock of the times the students sign the attendance sheets. This poses the following problems: Problem of keeping track of the number of times the student comes to the class using the attendance sheet; Lecturers manually recognizing students that are rightful for the class; Impersonation at lectures and examination; Students' identity management and access control problem. Though fingerprint authentication has been adopted in developing countries and has done very well but still not suitable in certain situation where people have no finger or the finger has been mutated; situations where identification needs to be done quickly during examination because of time; situations where identification needs to be done without intruding. These informed the need for a complementary facial recognition system to take attendance in tertiary institutions.

Facial recognition has been used in several areas such as security and detection of criminals or suspects. It has been a means of authentication and access control and identity management in some private corporations but has not been applied in tertiary institutions to automatically take attendance in developing countries. That is, the technology has suffered low adoption in educational institutions.

In view of this, there is the need for an automated system for taking attendance at both lectures and examination. This paper considers facial recognition system among other biometric systems. Automated facial recognition system proposed in this work will take attendance automatically by capturing the student's facial picture and perform necessary verification. The facial recognition system is envisaged to provide suitable and reliable way for detecting the face of the students, taking the attendance and verifying the studentship of the person. Face recognition is one of the few biometric methods that possess the merits of both high accuracy and low intrusiveness.

The rest of the paper is organized as follows: Review of related literature is presented in Section 2. Presented in Section 3 are: Proposed hybridized facial recognition algorithm, Architecture of the system prototype, Database models and envisaged query transactions. System implementation and Real life application structure of the system are presented in Section 4. Conclusion and recommendation for future works are presented in Section 5.

2.1 Related Works

In [3], a classification of face detection methods was presented. The methods are: Knowledge-based, Template matching and Appearance based. Knowledge based method (i.e. rule-based method) of face detection was examined in [4]. It captures knowledge of faces, and translates them into a set of rules. The problem with methods in this category is the difficulty in building an appropriate set of rules. A solution is to build hierarchical knowledge-based methods to overcome these problems. However, this approach alone is limited. It's unable to find many faces in a complex image.

In [5], the knowledge based method developed is divided into several steps. Firstly, it tries to find eye-analogue pixels so as to remove unwanted pixels from the image. After performing the segmentation process, it considers each eye-analogue segment as a candidate of one of the eyes. Then, a set of rule is executed to determine the potential pair of eyes. Once the eyes are selected, the algorithm calculates the face area as a rectangle. The four vertexes of the face are determined by a set of functions. Thus, the potential faces are normalized to a fixed size and orientation. Then the face regions are verified. Finally, a cost function is applied to make the final selection. The authors reported a success rate of 94%, even in photographs with many faces. This method seems to be efficient with simple inputs. The challenges and limitations are inefficiencies and errors when a man is wearing glasses.

In [6], their paper incorporates color information into a face detection method based on principal components analysis. Instead of performing pixel-based color segmentation, a new image which indicates the probability of each image pixel belonging to a skin region (skin probability image) was created.

In [7], an overview of the Viola-Jones face detection algorithm in [8] was carried out. Viola-Jones face detector contains three main algorithms that make it possible to build a successful face detector that can run *in real time. These* are: the integral image, classifier learning with Adaptive Boosting (AdaBoost) and the attentional cascade structure algorithms.

The facial recognition process normally has four interrelated phases or steps. The first step is face detection, the second is normalization, the third is feature extraction, and the final step is face recognition. These steps are separate components of a facial recognition system and depend on each other [4, 9]. Figure 1 present the relationship diagram between the phases.

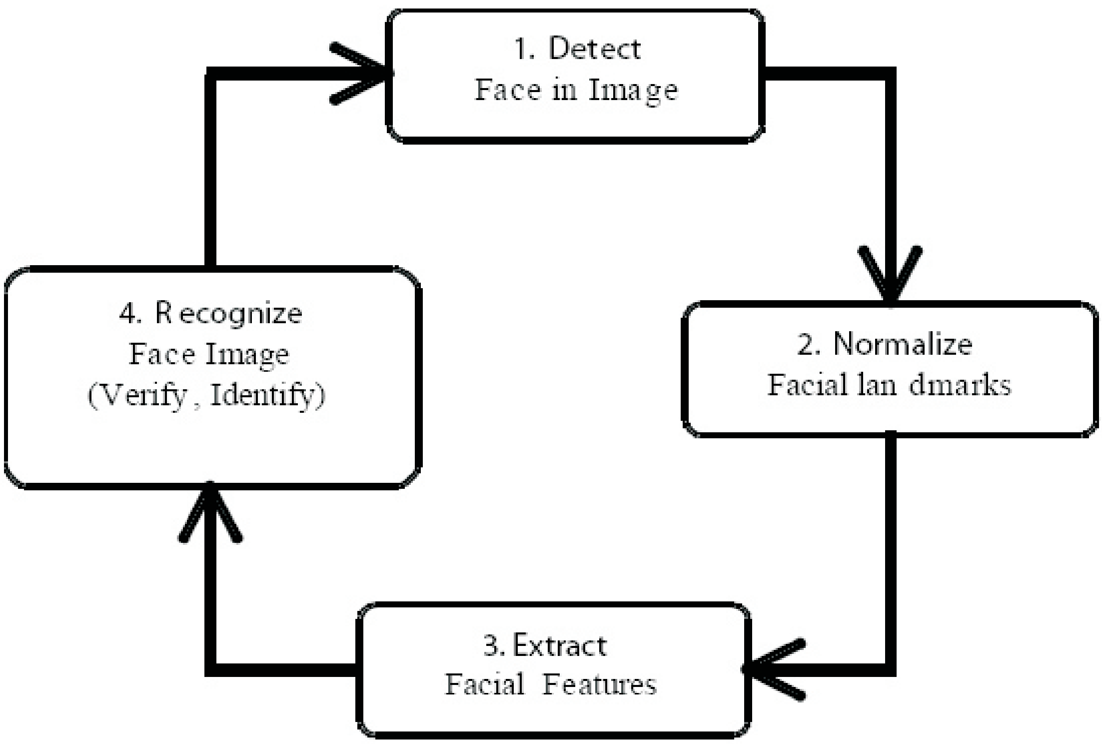

Figure 1:Steps in the facial recognition process [4]

2.2 Face Recognition Methods

a. Geometric Based methods

The geometry feature based methods analyze both local features and their geometric relationships. This approach is often called feature based method. Examples of this approach are some of the Elastic Bunch Graph Matching algorithms developed in [10].

b. Piecemeal methods

Piecemeal approach is a type of approach in facial feature detection which deals with minimalism; the idea is to use very few facial features detected, instead of waiting to get all features. It assumes it is a face as long as those few features have been detected [10, 11].

c. Appearance-based/Model-based methods

Appearance based methods represent a face in terms of several raw intensity images. An image is considered as a high-dimensional vector. Then statistical techniques are usually used to derive a feature space from the image distribution. The sample image is compared to the training set. On the other hand, the model-based approach tries to model a human face. The new sample is fitted to the model, and the parameters of the model are used to recognize the image. Appearance methods can be classified as linear or non-linear, while model-based methods can be 2D or 3D [12].

d. Template matching face recognition methods

Template matching process uses pixels, samples, models or textures as pattern. The recognition function computes the difference between these features and the stored templates. It uses correlation or distance measures [10].

e. Statistical approach to Face recognition algorithm

In statistical approach, each image is represented in terms of the features. It is viewed as a point (vector) in a dimensional space. Therefore, the goal is to choose and apply the right statistical tool for extraction and analysis of the underlying manifold. These tools must define the embedded face space in the image space and extract the basic functions from the face space. This would permit patterns belonging to different classes to occupy disjoint and compacted regions in the feature space. Consequently, a line, curve, plane or hyper plane that separates faces belonging to different classes could be defined [4].

f. Eigenface (or Principal Component Analysis (PCA)) method

The basic concept behind eigenface method is information reduction, when an evaluation of a small image is done; there is a great amount of information present. This method generates base-faces and then represents any image being analyzed by the system as a linear combination of the base faces. Once the base faces have been chosen, the problem has been reduced to a standard classification problem. Euclidian distance measure is used here for classification [13]. This Facial recognition method can be broken down into the following components: Generate the eigenfaces; Project training data into face-space to be used with a classification method; Evaluate a projected test element by projecting it into face space and comparing to training data.

g. Linear Discriminant Analysis (LDA)

In [9], LDA was described as a statistical approach that is based on the same statistical principles as PCA. LDA classifies faces of individuals unknown based on a set of training images of the individuals. This method finds the underlying vectors in the facial feature space that will, to a good extent, maximize the variance between individuals and minimize the variance within a number of the samples of the same individuals. Using this algorithm, there must be an appropriate training set. The database used should contain several samples of faces for each subject in the training set and at least one sample in the test set. The samples normally represent different frontal views of subjects with minor variation in view angle and they should also include different facial expressions, lighting and background conditions. Increase in the number of varying samples of the same person will allow the algorithm to optimize the variance between classes and therefore become more accurate. For LDA to work well the probe image must be similar to the gallery image in terms of size, pose and illumination.

h. Elastic Bunch Graph Matching (EBGM)

According to Lucas and Helen in [9], Elastic Bunch Graph Matching relies on the concept that the real face images have many and several nonlinear characteristics that are not addressed by the linear analysis methods such as PCA and LDA, such as variations in illumination, pose and expression. This method places some blocks of numbers known as Gabor Filters over small areas of the image, multiplying and adding the blocks with the pixel values to produce numbers known as jets at various locations on the image. These locations can be adjusted to accommodate minor variations. The success and correctness of Gabor Filters is in the fact that they remove much variability in images due to variation in lighting and contrast. But it is only against small shifts and little deformations. This technique has enhanced facial recognition performance under variations of pose, angle and expression.

According to Marques in [4], the challenges of face recognition system can be attributed to some factors such as:

a. Pose variation: The ideal scenario for face detection would be one in which only frontal images were involved. But this is very unlikely in general uncontrolled conditions. Moreover, the performance of face detection algorithms drops when there are large pose variations. Pose variation can happen due to subject's movements or camera's angle.

b. Feature occlusion: The presence of elements like beards, glasses or hats introduces variability. Faces can also be partially covered by objects or other faces. Facial expression and facial features also vary because of different facial gestures.

c. Imaging conditions: Different cameras and ambient conditions can affect the quality of an image thus affecting the appearance of a face.

3.1 Proposed Hybridized Facial Recognition Algorithm

The proposed hybridized algorithm is a combination of existing algorithms that complement one another. It combines the functionalities of existing algorithms and thus takes advantage of their strengths.

Main Algorithm: **Hybridize face detection and recognition algorithm with preprocessing, normalization and extraction components.**

Step 1: Using the underlying operating system level integration, initialize and detect camera or any video/image source in the system. i.e. on windows, use gdi32.dll library and win API to detect and get camera or image source

Step 2: Get a picture frame from the camera or image source.

Step 3: Scale image to specified size of the program, sometimes at run-time.

Step 4: Adaptive Smoothing Algorithm starts for preprocessing:

Invoke the Adaptive Smoothing Algorithm in sub-module 1.

Step 5: Erosion Operator Algorithm starts for preprocessing:

Invoke the operator algorithm in sub-module 2.

Step 6: Start Eigenface Algorithm for face detection:

Invoke the Eigenface Algorithm in sub-module 3.

If algorithm detects face in the image then go to step 7 else go to step 9.

Step 7: Luzand facial extraction Algorithm library starts for facial feature extraction:

Invoke the Luzand facial extraction Algorithm library in sub-module 4.

Step 8: Luzand face Recognition Algorithm library starts for face recognition:

Invoke the Luzand face Recognition Algorithm library in sub-module 5.

Step 9: Algorithm terminates

Sub-modules of the hybridized face recognition algorithm

Sub-module 1: Adaptive Smoothing Algorithm

The algorithm developed for Adaptive Smoothing is adopted from [14], a framework developed by Benjamin and Kitney:

Step 1: Initialize structures or class for storing different face templates

Step 2: Break image into pixels contained in an array or any similar data structure.

Step 3: calculate weights for 9 pixels and 8 neighbors using Equation 1.

$$\left.\begin{array}{l} w(x,y) = \exp(-1 * (Gx^2 + Gy^2)/(2 * factor^2)) \\ Gx(x,y) = (I(x+1,y) - I(x-1,y))/2 \\ Gy(x,y) = (I(x,y+1) - I(x,y-1))/2 \end{array}\right\} \text{Equation 1}$$

where *factor* is a configurable value determining smoothing's quality.

i. Calculate sum of 9 weights (Weight Total).

ii. Calculate sum of 9 weighted pixels values (Total).

iii. Calculate the destination pixel as Total/ Weight Total.

Step 4: Algorithm terminates.

Sub-module 2: Erosion operator Algorithm

The algorithm developed for Erosion operator is adopted from [14]:

Step 1: Assign minimum value of surrounding pixels to each pixel of the captured image.

Step 2: Surrounding pixels which are to be processed are specified by structuring element: 1- to process the neighbor, -1 to skip it.

Step 3: Accept 8 and 16 bpp (bit per pixel) gray scale images and 24 and 48 bpp color images for processing.

Step 4: Call the Apply function (Mathematical Library function of MATLAB) on the image and the noise is removed.

Step 5: Terminate Algorithm.

Sub-module 3: Eigenface Algorithm

The algorithm developed for the eigenface detection is adopted from [13]:

Step 1: Get the sample images for training

i. Gather all sample images being used and assume that M sample images are used. Each sample image will be referred to as where n indicates n^{th} sample image such that. $1 \le n \le M.$

ii. Each $\tilde{A}_n$ should be a column vector. Represent it in the computer program as pixels having (x,y) coordinates with (0,0) being at the upper left corner either as multidimensional array or jagged array. If the images are x pixels across and y pixels tall, then the column vector will be of size $(x*y)*1$.

Step 2: Calculate the Average Face using Equation 2

$$\phi = \frac{1}{M} \sum_{i=1}^{M} \tilde{A}_i \quad \text{Equation 2}$$

Where ϕ=average image found from the sample images.

Step 3: Generate the difference in faces by subtracting the average face from each sample image using Equation 3,

$$\ddot{O}_n = \tilde{A}_n - \phi \;, \quad 1 \le n \le M \quad \text{Equation 3}$$

Where $\ddot{O}_n$ are vectors that are the difference between each sample image and the average image.

Step 4: Calculate the Covariance matrix AA^T from the different faces using Equation 4

$$A = [\ \ddot{O}_1\ \ddot{O}_2\ \ddot{O}_3\ \ldots\ \ddot{O}_m] \quad \text{Equation 4}$$

The matrix A will be of size $(x*y)*M$

Step 5: Calculate the Eigenfaces of the

This cannot be directly done, because the size of AA^T is $(x*y)*(x*y)$. Doing these calculations on the image 200*200 pixels will be tasking even for a specialized hardware. The eigenvectors of AA^T matrix can be found by considering the linear combinations of the eigenvectors of the matrix using Equation 5

$$u_k = \frac{\sum_{i=1}^{M} \ddot{O}_l\, x_{lk}}{\sqrt{\lambda_k}} \quad \text{Equation 5}$$

where u_k is the k[th] eigenface of the training data (the eigenvector of AA^T

Step 6: Calculate the discriminating power of Eigenfaces using Equation 6

$$\hat{U} = U^T(\tilde{A}_n - \phi) \quad \text{Equation 6}$$

Step 7: After the image is projected into face space the distance to the closest mean for a known individual is found. The distance is compared to two thresholds, which determine if the point is close enough to a cluster to be considered a face image.

If distance is close to the cluster then image is a face else image is not a face

Step 7: Algorithm terminates.

Sub-module 4: Luxand facial extraction Algorithm library

The algorithm developed for the facial extraction is adopted from [15]:

Step 1: convert face image to CLR image using function ToCLRImage defined in the Luxand facial library.

Step 2: Detect all facial features by detecting individual features.

i. Detect Eyes in Region by passing the face image and getting the eye centers in an array data type and stored in points 0 and 1. Detect the Left eye, Right eye, left eye inner corner, left eye outer corner, left eye lower line1, left eye left iris corner etc. The remaining features are listed in Appendix A.

ii. Detect Nose in Region by passing the face image and getting the nose parts in an array data type. Detect the nose tip, nose bottom, nose bridge, nose left wing, nose right wing, nose right wing outer etc. The remaining features are listed in Appendix A.

iii. Detect Mouth in Region by passing the face image and getting the mouth parts in an array data type. Detect the mouth right corner, mouth left corner, mouth top etc. the remaining features are listed in Appendix A.

iv. Detect Chin in Region by passing the face image and getting the chin parts in an array data type. Detect the chin bottom, chin left, chin right etc. The remaining features are listed in Appendix A.

Step 3: For each features detected in the feature arrays, draw an Ellipse using *point.x* and *point.y* to show the points around the image.

Step 4: Algorithm terminates.

Sub-module 5: Luxand face Recognition Algorithm library

The algorithm developed for face recognition is adopted from [15].

Step 1: Initialize structures or class for storing different face templates

i. Initialize a structure or class that stores face templates of the face to be recognized.

ii. Initialize a structure or class that stores face templates of the faces already in the face database.

Step 2: Identify weather the recognition type is face verification or face identification

If the recognition type is face verification then go to step 3

else go to step 4.

Step 3: using face verification method. This is 1:1 match method that compares the face image with a template face in the face database.

i. Convert image to CLRImage using the ToCLRImage method of the FSDK.Cimage class.

ii. Get face templates in region of the face image using the face position in the facial feature algorithm. Name as *template1*.

iii. Set the threshold at far variable to any percentage. 30% was used to reduce the false reject rate.

iv. Set the similarity variable.

v. Add generated face templates to the structure or class that stores face templates of the face to be identified.

vi. Get the face image to match with from the face database.

vii. Generate the face templates in region of the face image to compare with. Generate templates and store in an array or list data structure as *template2* for 10 times where templates2 contains template at each element i.e. *templates2 [1]...templates2 [10]*.

viii. For each template in the *template2* array structure, iterate through and use the library algorithm function of the FSDK to match the face templates *template1*. Get similarity and threshold for each run.

ix. If similarity value is greater than threshold value then the face to identify matches with the face image in the database. Otherwise face image doesn't match, and then raise an error and message to notify.

Step 4: Using face identification method. This is 1: N match method that compares the face image with all the templates in the face database.

i. Convert image to CLRImage using the ToCLRImage method of the FSDK.Cimage class.
ii. Get face templates in region of the face image using the face position in the facial feature algorithm. Name as *template3*.
iii. Set the threshold at far variable to any percentage. 30% was used to reduce the false reject rate.
iv. Set the similarity variable.
v. Add generated face templates to the structure or class that stores face templates of the face to be identified.
vi. Get all the face images from the face database to use to match the *template3*.
vii. For each face image gotten from the database: Generate the face templates in the region of the face images to compare with. Generate templates and store in an array or list data structure as *template_xn,* where;

$$1 \leq n \leq total\ no\ of\ face\ image\ in\ the\ database$$

T*emplates_xn* contains templates at each elements i.e. *templates_xn [1]...template_xn [10]*.
i. For each template in the *template_xn* array of structure iterate through and use the library algorithm function of the FSDK to match the face templates template3. Get similarity and threshold values for each run.
ii. If similarity value is greater than threshold then the face to identify matches the face image in the database. Otherwise face image doesn't match, and then raise an error and message to notify.

Step 5: Algorithm terminates.

3.2 Architecture of the System Prototype

A system prototype is developed to implement the proposed hybridized algorithm. The prototype architecture is presented in Figure 2. The system interfaces with a camera or video source that captures the probing face to verify or identify. The camera is placed inside an automated security door that the student has to pass through to enter the class. The security door is an automated door that guards the entrance to the lecture room/theatre. The student enters the door and positions his/her face up for the camera inside the door to capture the face and send it to the application server that runs the recognition system to verify and authenticate his/her face, if the face is valid and recognized, the automated security door opens and if otherwise the door beeps and alerts the student to step out, indicating rejection. The captured face is moved to the preprocessing component that performs preprocessing functions like smoothing, color balancing, transformations, scaling etc. in preparation for the real processing and operations. After the picture has been preprocessed, face detection techniques are then applied to the preprocessed image to detect whether a human face exists in the picture or not if a face exists it proceeds, else it will respond by generating an error message. After the face has been detected, the important facial features depending on the algorithm will be fine-tuned, boosted and enhanced.

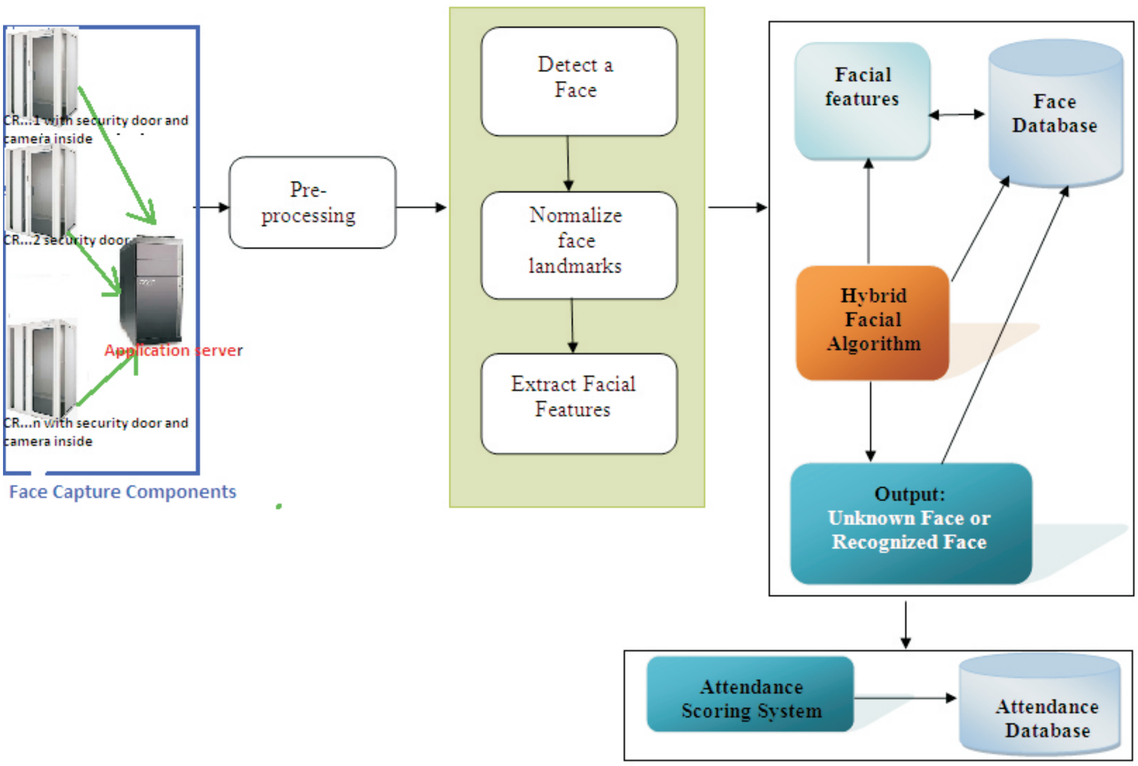

Figure 2:Architecture of the Facial Recognition System for Attendance Taking (FRSAT)

Next is Normalization of the face detected. Firstly, separation of the background and the face is done, then the image is standardized in terms of size, pose and illumination; this is done relative to the images in the facial database. Normalization algorithms can be used to correct the face image. After normalization, facial feature extraction is then performed on the normalized face image. In feature extraction, a mathematical representation of the face is generated and this serves as the basis of the recognition tasks. Also some facial features like nose, eyes etc. at specified points can be located and extracted. The facial features generated can be stored in the facial database and used in the system as vector representation. Feature selection then selects the best and relevant features extracted by extraction algorithms. This process will help in the performance and success of the system, because when the right features are selected, the chance of better recognition increases.

The facial recognition algorithm which consists of appearance-based and statistical methods will then be applied. This method consists of eigenfaces and linear discriminant analysis. Other approaches and light-weighted algorithm will be applied as part of the hybrid facial recognition algorithm. The facial recognition algorithm processes the facial features and it also interacts with the database to get training set and success rate data to improve and enhance the recognition power. The result of the recognition algorithm is whether the face exists in the facial database or it does not exist. That is, whether the face is recognized or unknown. The result is then stored in the facial database to form the training set and data for improvement.

The facial database contains faces and templates of a particular subject. The stored template faces are normalized and standardized. The limitation of this method is that the facial templates must be the same in size and pose. The illumination must be controlled and balanced. The facial database also contains the result of every processing done as part of the training set, false and success rate. This data helps to boost performance on where to improve in the recognition process.

After the face verification of the students has been concluded the system then marks the attendance if the face matches with the student or declines if the student face does not exist in the facial database. The attendance system takes care of attendance scoring, reporting and compilation.

3.3 Database Models and Envisaged Query Transactions

Relational database model is adopted and the relations envisaged are stated below; however more relations can be identified in future work.

a. STUDENT[Matric_number, student_name, level, college, department]
b. LOGIN[user_id, user_password, user_type, course_code, lecturer_name, date_registered]
c. STUDENTFACE[Matric_number, face_id, student_picture]
d. ATTENDANCE[Matric_number, course_code, attendance_score]
e. FACEFEATURE[Matric_number, feature1, feature2, feature3, face_id]
f. COURSE[course_code, course_desc, lecturer_name]

The logical diagram of the interrelationship between the relations in the database is presented in Figure 3.

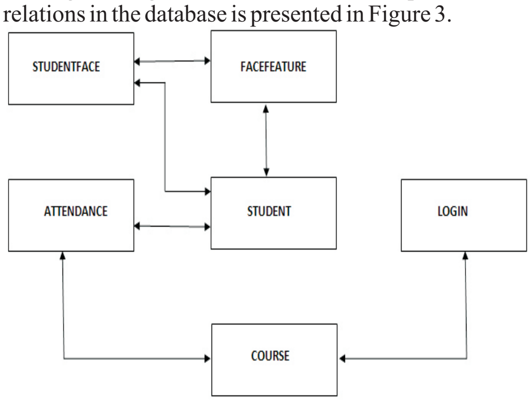

Figure 3: Logical diagram of the interrelationship between the database relations

The envisaged query transactions on the database are listed below:

a. List specific / all the lecturers that registered in the system
b. List specific / all the students that are in the database
c. List of student's attendance for a given course at a given date
d. Get the Record of attendance of a given student in a given course, semester and a given session.
e. Get the records of attendance of a group of students in a given course, semester, session.
f. Get the number of time(s) the student is absent.
g. List specific / all the courses available in the database for a semester.
h. Get the attendance score for students.

4.1 System Implementation

The system is implemented using C# and C binding as DLLs (Dynamic Link Library) exposed as APIs (Application Programming Interface) with some C bindings/header files and library. The front end is implemented entirely using .NET's graphical user interface library and windows underlying Graphical User Interface logic. In the system, the homepage has a username and password to grant access and authentication to valid lecturers to register and sign in students for a course. The administrator registers the lecturers and allows them to sign in and use the functionalities applied to them.

The system is composed of three main modules which are student registration, student attendance and sign-in module and the reporting module. Firstly the system authorizes the lecturers and administrators before gaining access to the full system. The students' face capture and registration module consists of functionalities to register the students' faces and credentials to grant access and eligibility to students for a particular course and the

module interface is presented in Figure 4. The student attendance and sign-in module allows the students' faces to be captured and processed for recognition and authentication into the class and attendance scoring. This module lists the students that have signed into the class. The sign in is automatically performed when the system detects the student face and it performs authentication processing to check if the face exists among the students that are eligible to take the course.
The reporting module provides the platform to present various tabular reports needed by the lecturers and administrators and the interface is presented in Figure 5.

4.2 Real life Application Structure

Presented in Figure 6 is the proposed structure for real life application of the system. The following are infrastructures that will be required: an automated security door, an IP camera/CCTV, a DVR, a computer server containing the recognition/attendance component in the application server and control logic, a database server and a network that connects these devices and systems together and other auxiliary devices.
The real life application structure uses 3-tier client server architecture. The three tiers are:

a. The client tier or user interface. This tier contains the entire user interfaces through which the system administrator, lecturers and students can interact with the system. It is the front-end of the application.

b. The middle tier or business logic controls the core functionalities of the system. It controls the face detection, extraction, recognition, control logic and reporting functionalities. It is the middle-end of the application.

c. The data storage tier. This consists of the database server that houses the face database and attendance database. This tier contains relational database managed by a Relational Database Management System (RDBMS) that keeps the student's faces and attendance data and it serves the application server. Records for query/transaction processes are retrieved from the databases. This is the back-end of the application.

The database server will be in the institution's server room and connected to the Intranet of the institution that links the other systems and devices. The users access the application via a given URL (Uniform Resource Locator). The cameras and doors in each lecture halls are connected to the application server through the network.

4.3 Performance Comparison

The performance of proposed FRSAT and the existing manual method of taking attendance were compared using the following performance metrics: processing speed, data volume, accuracy, discovering of impersonation and ease of use by users. Table 1 presents the performance details.

Table 1: Performance Comparison

Metrics	Proposed Automated Design (FRSAT)	Existing Manual Design
Speed of processing	Data are processed at the processor speed of computer	Data are processed at human speed which is slow compared to the high processor speed of computers
Volume of data processed	Large volume of data are processed depending on the capacity of the computer memory	Small volume of data and this depend on the human capacity
Accuracy	Very accurate provided the system is correctly programmed and with correct data	Not as accurate as FRSAT. It is subject to some human errors (e.g. error of omission, commission e.t.c.)
Discovering of impersonation	System easily and quickly reject illegitimate students from impersonating because there is a database of faces of registered and legitimate students for each course	Detecting impersonation is not very easy because students can clone fake identity cards and lecturers may not be able to detect
Ease of use	System is user friendly and very easy to use and it never gets tired. Thus no monotony of data processing.	It is always tedious to manually collate attendance records and data processing is monotonous because same process is repeated for several times

Conclusion and Future Works

The implementation of the hybridized facial recognition algorithm to take class attendance in tertiary institutions will help to make attendance taking in tertiary institution efficient and effective. The developed prototype is tested in Bells University of Technology, Ota, Ogun State, Nigeria using some life data. Though there are no security doors with cameras but the application was installed on a server with few workstations having inbuilt web camera installed. Some students were positioned before the workstations and their attendance were recorded automatically after their faces have been recognized. The manual method of taking attendance is characterized by a lot of flaws such as rigour of compiling attendance, impersonation and problem of identity management; thus this work proposes automated facial recognition based paradigm to solve these challenges.

Some limitations are identified in this work and they are:

a. Pose variation. Student has to appear in the same posture during registration and sign-in.

b. Feature occlusion: The presence of elements like beards, glasses or hats introduces high variability and problem.

c. Illumination: The problem of lighting variations in the environment.

In view of the above limitations, future works are suggested as follows:

a. Incorporate algorithms that handles pose variation so that students will be able to appear in any pose during registration and verification and thus make the system more flexible.

b. Incorporate algorithms that allow students to wear anything on their face, and appear with beards, glasses etc. during verification and still recognize the faces.

c. Develop algorithms that perform effectively in spite of lighting variations in the environment.

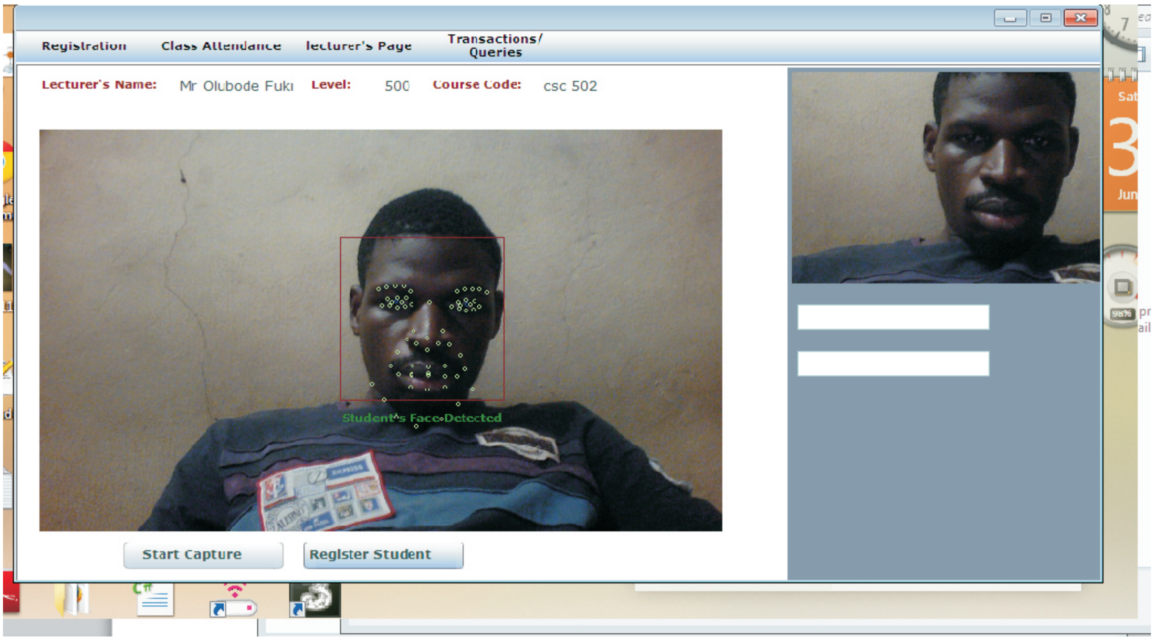

Figure 4:Student Face Capture and Registration Interface

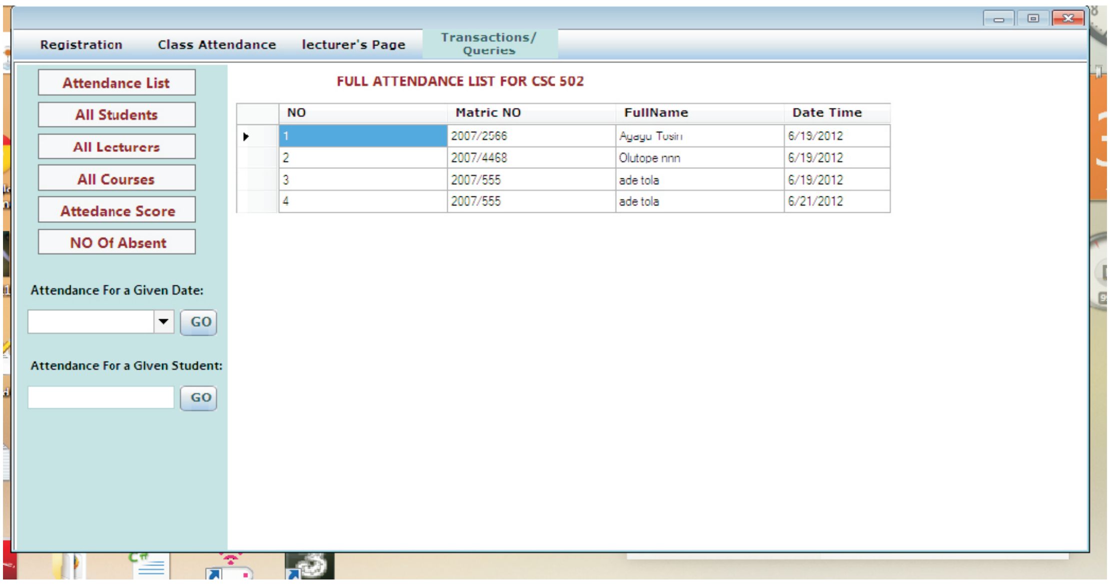

Figure 5:Reporting/Queries Interface

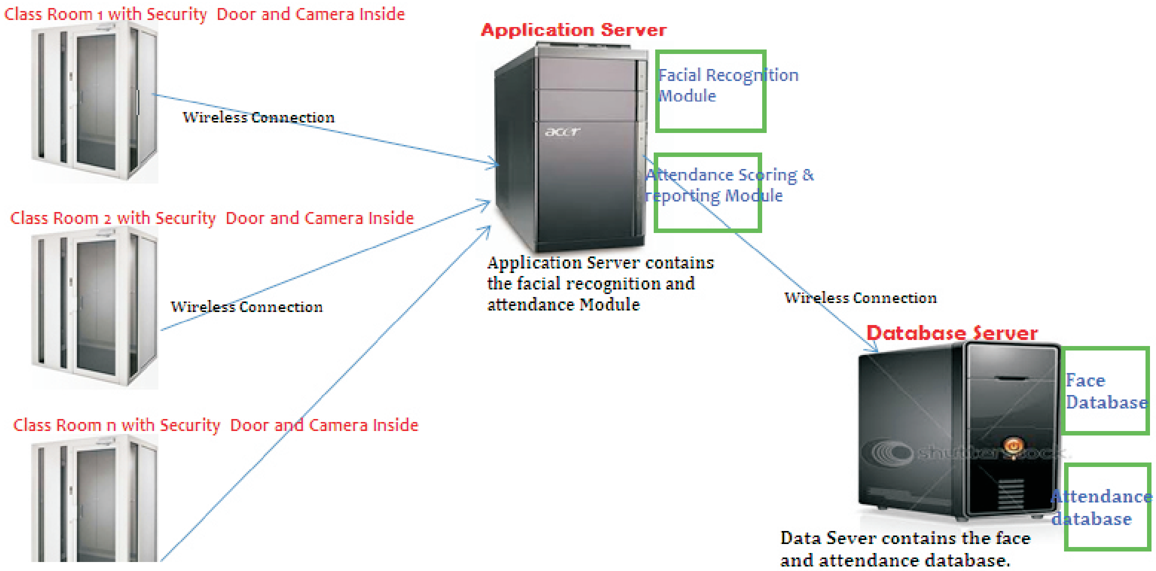

Figure 6:Real life application structure

Appendix A

List of Feature Extraction Methods Proposed by Luzand in [15]

Facial Feature Name	Value	Facial Feature Name	Value
FSDKP_LEFT_EYE	0	FSDKP_LEFT_EYEBROW_MIDDLE	16
FSDKP_RIGHT_EYE	1	FSDKP_LEFT_EYEBROW_MIDDLE_LEFT	18
FSDKP_LEFT_EYE_INNER_CORNER	24	FSDKP_LEFT_EYEBROW_MIDDLE_RIGHT	19
FSDKP_LEFT_EYE_OUTER_CORNER	23	FSDKP_LEFT_EYEBROW_OUTER_CORNER	12
FSDKP_LEFT_EYE_LOWER_LINE1	38	FSDKP_RIGHT_EYEBROW_INNER_CORNER	14
FSDKP_LEFT_EYE_LOWER_LINE2	27	FSDKP_RIGHT_EYEBROW_MIDDLE	17
FSDKP_LEFT_EYE_LOWER_LINE3	37	FSDKP_RIGHT_EYEBROW_MIDDLE_LEFT	20
FSDKP_LEFT_EYE_UPPER_LINE1	35	FSDKP_ RIGHT_EYEBROW_MIDDLE_RIGHT	21
FSDKP_LEFT_EYE_UPPER_LINE2	28	FSDKP_RIGHT_EYEBROW_OUTER_CORNER	15
FSDKP_LEFT_EYE_UPPER_LINE3	36	FSDKP_NOSE_TIP	2
FSDKP_LEFT_EYE_LEFT_IRIS_CORNER	29	FSDKP_NOSE_BOTTOM	49
FSDKP_LEFT_EYE_RIGHT_IRIS_CORNER	30	FSDKP_NOSE_BRIDGE	22
FSDKP_RIGHT_EYE_INNER_CORNER	25	FSDKP_NOSE_LEFT_WING	43
FSDKP_RIGHT_EYE_OUTER_CORNER	26	FSDKP_NOSE_LEFT_WING_OUTER	45
FSDKP_RIGHT_EYE_LOWER_LINE1	41	FSDKP_NOSE_LEFT_WING_LOWER	47
FSDKP_RIGHT_EYE_LOWER_LINE2	31	FSDKP_NOSE_RIGHT_WING	44
FSDKP_RIGHT_EYE_LOWER_LINE3	42	FSDKP_NOSE_RIGHT_WING_OUTER	46
FSDKP_RIGHT_EYE_UPPER_LINE1	40	FSDKP_NOSE_RIGHT_WING_LOWER	48
FSDKP_RIGHT_EYE_UPPER_LINE2	32	FSDKP_MOUTH_RIGHT_CORNER	3
FSDKP_RIGHT_EYE_UPPER_LINE3	39	FSDKP_MOUTH_LEFT_CORNER	4
FSDKP_RIGHT_EYE_LEFT_IRIS_CORNER	33	FSDKP_MOUTH_TOP	54
FSDKP_RIGHT_EYE_RIGHT_IRIS_CORNER	34	FSDKP_MOUTH_TOP_INNER	61
FSDKP_LEFT_EYEBROW_INNER_CORN	13	FSDKP_MOUTH_BOTTOM	55

ER			
FSDKP_MOUTH_RIGHT_TOP_INNER	62	FSDKP_NASOLABIAL_FOLD_RIGHT_LOWER	53
FSDKP_MOUTH_LEFT_BOTTOM	58	FSDKP_CHIN_BOTTOM	11
FSDKP_MOUTH_LEFT_BOTTOM_INNER	63	FSDKP_CHIN_LEFT	9
FSDKP_MOUTH_RIGHT_BOTTOM	59	FSDKP_CHIN_RIGHT	10
FSDKP_MOUTH_RIGHT_BOTTOM_INNER	65	FSDKP_FACE_CONTOUR1	7
FSDKP_NASOLABIAL_FOLD_LEFT_UPPER	50	FSDKP_FACE_CONTOUR2	5
FSDKP_NASOLABIAL_FOLD_LEFT_LOWER	52	FSDKP_FACE_CONTOUR12	6
FSDKP_NASOLABIAL_FOLD_RIGHT_UPPER	51	FSDKP_FACE_CONTOUR13	8

References

[1] Yougsheng G. and Leung K.H. (2002). Face Recognition using Line Edge Map. *IEEE Transactional Pattern Analysis Machine Intelligence*, Vol 24, pp 764-779.

[2] *Pentland A.* and *Choudhury T. (2000)*. Face Recognition for Smart Environments. *IEEE Computer*, Vol. 33, No. 2, pp. 50-55.

[3] Yang M.H., Kriegman D.J. and Ahuja N. (2002). Detecting faces in images: A survey. *IEEE Transactions on Pattern analysis and Machine intelligence*. Vol. 24, Issue 1, pp. 34-58.

[4] Marques I. (2010). Face Recognition Algorithms. Available @ www.makr.ca/papers.htm Access Date: 3/3/2012.

[5] *Han C.C.*, *Liao H.Y.*, *Yu* G.J. and *Chen L.H.* (1997). Fast Face Detection via Optimization. *Image and Vision Computing*, Vol. 15, pp. 511-527.

[6] Menser B. and Muller F. (2002). Face Detection in Color Images Using Principal Components Analysis. Aachen University of Technology (RWTH), Germany.

[7] *Zhengyou* Z. and Cha Z. (2010) "A Survey of Recent Advances in Face Detection".Microsoft Research, one way Redmond, CA, USA.Available @ http:// www.research.microsoft.uk. Access Date: 3/3/2012.

[8] Viola P. A. and Jones M.J. (2001). Fast and Robust Classification using Asymmetric AdaBoost and a Detector Cascade. *In Proceedings of NIPS 2001, NIPS 2001*, December 3-8, Vancouver, British Columbia, Canada, pp. 1311-1318.

[9] Lucas D. and Helen N. (2009). **Facial Recognition Technology** A Survey of Policy and Implementation Issues. Lancaster University, UK. Centre for the Study of Technology and Organization.

[10] Wiskott L., Fellous J.M. and Krüger N. (1997). **Face Recognition by Elastic Bunch Graph Matching.** *Proceedings of IEEE International Conference on Image Processing, ICIP'97*, Santa Barbara, Oct. 26-29, Vol. 1, pp. 129-132.

[11] Sinha P., Balas B. J., Ostrovsky Y. and Russell, R. (2006). Face Recognition by Humans. In *Face Recognition: Advanced Modeling and Methods*, Academic Press.

[12] *Blanz* and *Vetter* T. (1990). Face Identification based on Fitting a 3d Morphable Model. *IEEE Transaction, PAMI*, Vol. 9, pp. 10631074.

[13] Burns C. and Morlok R. (2003). Comparison of Face Recognition Methods. Available @ www.morburns/articles/facemethods.html. Access Date: 3/3/2012.

[14] Aforge.Net (2006). Aforge.net Framework. Available @ www.Aforgenet.com.

[15] Luxand (2005). Face Detection and Recognition Library: Developer's Guide. Available @ www.luxand.com