Project Report

on

ATTENDANCE AUTOMATION SYSTEM USING FACE DETECTION AND RECOGNITION

Submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY

DEGREE

Session 2016-17
in

Computer Science Engineering

By

UMANG GUPTA
1303210175
YOGENDRA KUMAR
1303210187
SUYASH SRIVASTAVA
1303210167
SIDDHARTHA PRATAP SINGH
1303210160

Under the guidance of

MR. PRABHAT SINGH

ABES ENGINEERING COLLEGE, GHAZIABAD

AFFILIATED TO
DR. A.P.J. ABDUL KALAM TECHNICAL UNIVERSITY, U.P., LUCKNOW
(Formerly UPTU)

Project Report

on

ATTENDANCE AUTOMATION SYSTEM USING FACE DETECTION AND RECOGNITION

Submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY

DEGREE

Session 2016-17
in

Computer Science Engineering

By

UMANG GUPTA
1303210175
YOGENDRA KUMAR
1303210187
SUYASH SRIVASTAVA
1303210167
SIDDHARTHA PRATAP SINGH
1303210160

Under the guidance of

MR. PRABHAT SINGH

ABES ENGINEERING COLLEGE, GHAZIABAD

AFFILIATED TO
DR. A.P.J. ABDUL KALAM TECHNICAL UNIVERSITY, U.P., LUCKNOW
(Formerly UPTU)

STUDENT'S DECLARATION

We hereby declare that the work being presented in this report entitled "ATTENDANCE AUTOMATION SYSTEM USING FACE DETECTION AND RECOGNITION" is an authentic record of our own work carried out under the supervision of Mr. Prabhat Singh.

The matter embodied in this report has not been submitted by us for the award of any other degree.

Dated: **Signature of Students(s)**

UMANG GUPTA

YOGENDRA KUMAR

SUYASH SRIVASTAVA

SIDDHARTHA PRATAP SINGH

Department: Computer Science

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Signature of HOD
Prof. (Dr.) Shailesh Tiwari
(Computer Science & Engineering Department)
Date............................

Signature of Supervisor
Mr. Prabhat Singh
(Assistant Professor)
(Computer Science & Engineering Department)

CERTIFICATE

This is to certify that Project Report entitled "**ATTENDANCE AUTOMATION SYSTEM USING FACE DETECTION AND RECOGNITION**" which is submitted by **Umang Gupta, Suyash Srivastava, Siddhartha Pratap Singh** and **Yogendra Kumar** in partial fulfillment of the requirement for the award of degree **B. Tech.** in **Department of Computer Science** of **DR. A.P.J. ABDUL KALAM Technical University**, is a record of the candidates own work carried out by them under my supervision. The matter embodied in this thesis is original and has not been submitted for the award of any other degree.

Mr. Prabhat Singh
(Supervisor)

Date

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B. Tech Project undertaken during B. Tech. Final Year. We owe special debt of gratitude to Professor Prabhat Singh, Department of Computer Science & Engineering, ABES Engineering College, Ghaziabad for his constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen light of the day.

We also take the opportunity to acknowledge the contribution of Professor (Dr.) Shailesh Tiwari, Head, Department of Computer Science & Engineering, ABES Engineering College, Ghaziabad for his full support and assistance during the development of the project.

We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind assistance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

Signature:	*Signature:*
Name : UMANG GUPTA	*Name : SUYASH SRIVASTAVA*
Roll No.: 1303210175	*Roll No.: 1303210167*
Date :	*Date :*
Signature:	*Signature:*
Name : YOGENDRA KUMAR	*Name : SIDDHARTHA PRATAP SINGH*
Roll No.: 1303210187	*Roll No.: 1303210160*
Date :	*Date :*

ABSTRACT

Face recognition is an essential field in many applications, one which is Attendance Management System. Now days taking the attendance of the Employees in Company had become a tedious job. Person identification is one of the most crucial building blocks for smart interactions. Among the person identification methods, face recognition is known to be the most natural ones, since the face modality is the modality that uses to identify people in everyday lives. Although other methods, such as fingerprint identification, can provide better performance, those are not appropriate for natural smart interactions due to their intrusive nature. In contrast, face recognition provides passive identification that is the person to be identified does not need to cooperate or take any specific action. So a company can recognize its regular employees while they are entering the company.

This Project is aimed for implementing a system that is capable of identifying the employees in an organization, marking their attendance. Therefore face recognition is used to mark the attendance of the employees. Smart Attendance using Real Time Face Recognition (SMART-FR) provides flexibility to identify several employees at the same time separately rather than identifying one by one. To increase the accuracy, efficiency and reliability of the recognition, algorithms are needed. Principle Component Analysis (PCA) and Haar cascade are used to address those tasks. The PCA is one of the most successful techniques that had been used in image recognition and compression

The software is to provide employers an easier and fool-proof way to mark attendance of the employees. The software will allow administrators enroll employees. The software will allow employees have their face captured for attendance marking purposes. Only administrators will have direct manipulation rights to the system. Employees will have no other interaction with the system except for having their faces captured whilst they are entering the office. It will be a console application and would require maximum uptime due to its sensitive purpose.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS

[x]	Integer value of x.
$\neq$	Not Equal
$\in$	Belongs to
€	Euro- A Currency
_	Optical distance
_o	Optical thickness or optical half thickness
Γ_i.	Face Image vector
Φ_i	Mean centered image.
Ψ	Average face Image
Ω_1	Projection of images
Θ_c,	Half the largest distance between any two face images
Λ	Eigen value

LIST OF ABBREVIATIONS

AAM	Active Appearance Model
ICA	Independent Component Analysis
ISC	Increment Sign Correlation
PCA	Principal Component Analysis
ROC	Receiver Operating Characteristics
AASFDR	Attendance Automation System using face Detection and Recognition
SMART FR	Smart Face Recognition
ATR	Automatic Target Recognition
SNoW	Sparse network of winnows

CHAPTER 1

INTRODUCTION

1.1. Problem Introduction

Organizations of all sizes use time and attendance systems to record when employees start and stop work, and the department where the work is performed. However, it's also common to track meals and breaks, the type of work performed, and the number of items produced. In addition to tracking when employees work, organizations also need to keep tabs on when employees are not working. Vacation time, compensation time, Family and Medical Leave Act (FMLA) time, and jury duty must be recorded. Some organizations also keep detailed records of attendance issues such as who calls in sick and who comes in late.

A time and attendance system provides many benefits to organizations. It enables an employer to have full control of all employees working hours. It helps control labor costs by reducing over-payments, which are often caused by transcription error, interpretation error and intentional error. Manual processes are also eliminated as well as the staff needed to maintain them. It is often difficult to comply with labor regulation, but a time and attendance system is invaluable for ensuring compliance with labor regulations regarding proof of attendance.

Companies with large employee numbers might need to install several time clock stations in order to speed up the process of getting all employees to clock in or out quickly or to record activity in dispersed locations. In the business world of today we all know one simple truth…TIME IS MONEY! We work to keep the amount of time it takes to complete even the simplest tasks down to the minimum.

Face recognition is an essential field in many applications, one which is Attendance Management System. Now days taking the attendance of the Employees in Company had become a tedious job. Person identification is one of the most crucial building blocks for smart interactions. Among the person identification methods, face recognition is known to be the most natural ones, since the face modality is the modality that uses to identify people in everyday lives. Although other methods, such as fingerprint identification, can provide better performance, those are not appropriate for natural smart interactions due to their intrusive nature. In contrast, face recognition provides passive identification that is the person to be identified does not need to cooperate or take any specific action. So a company can recognize its regular employees while they are entering and leaving the company.

1.1.1 Motivation

Face Detection and Recognition is an important area in the field of substantiation. As far as the attendance process of today organizations are concerned, most of them are using finger print identification to handle attendance of employees. So it is time consuming, inefficient and unreliable.

1.1.2 Project Objective

This Project is aimed for implementing a system that is capable of identifying the employees in an organization, marking their attendance. Therefore face recognition is used to mark the attendance of the employees. Smart Attendance using Real Time Face Recognition (SMART-FR) provides flexibility to identify several employees at the same time separately rather than identifying one by one. To increase the accuracy, efficiency and reliability of the recognition, algorithms are needed. Principle Component Analysis (PCA) and Haar cascade are used to address those tasks. The PCA is one of the most successful techniques that had been used in image recognition and compression.

Summarizing objectives of this project are:

- Creating a database that contains attendance information of the employees.
- Linking this system to an existing employee management system.
- Capturing live feed from camera to record attendance times.
- Linking the information captured by the feed to the database for accuracy of results.

1.1.3 Scope of the Project

The software is to provide employers an easier and fool-proof way to mark attendance of the employees. The software will allow administrators enroll employees. The software will allow employees have their face captured for attendance marking purposes. Only administrators will have direct manipulation rights to the system. Employees will have no other interaction with the system except for having their faces captured whilst they are entering the office. It will be a console application and would require maximum uptime due to its sensitive purpose.

1.2 Related Previous Work

Robust Real-Time Face Detection was proposed by Paul Viola and Michael J. Jones. Systems built on their proposal were beneficial only when used under various constraints. These constraints included disparate parameters that could not be controlled at times such as variation in the posture of the person, fluctuation in the luminosity of the surrounding, etc. Hence, the systems were termed to be inefficient when not utilized under the required constraints.

Real Time Human Face detection and Tracking was proposed by J. Chatrath, P. Gupta and P. Ahuja, A. Goel. Their paper describes the technique of real time profile detection and recognition by modifying Viola-Jones algorithm. Results achieved by the developed algorithm showed that up to 50 human faces could be detected and tracked by systems using the modified algorithm. Processing of data and time consumption is comparatively less in such systems.

Implementation of Attendance Management System was proposed by G. Lakshmi Priya and M. Pandimadevi. Systems built around this proposal would capture an image using a web camera at divergent instances. An accuracy of 68% was observed in such systems respectively.

1.3.Organization of the Report

This project work is structured as follow: Chapter one introduces the topic of Employee attendance and its relevance to any company and how it can reduce cost and increase productivity.

In chapter two, we will describe a variety of facial recognition algorithms and how they are interfaced. We will then move on to review recent works related to employee attendance and discuss some of their limitations.

Chapter three is based on the analysis we will be able to integrate employee attendance systems. Furthermore we will document the step by step procedures we used to fully implement this project.Chapter four is about testing and observing our newly implemented system to verify if we met our stated objective.Chapter five will draw conclusion on the work and make useful recommendations.

CHAPTER 2

LITERATURE SURVEY

2.1 INTRODUCTION

Attendance as defined by the Merriam-Webster dictionary:

- The number of people present at an event, meeting, etc.
- The act of being present at a place
- A record of how often a person goes to classes, meetings, etc.

Companies with large employee numbers might need to install several time clock stations in order to speed up the process of getting all employees to clock in or out quickly or to record activity in dispersed locations.

Depending on the supplier, identification method and number of clocking points required, prices vary widely. A time and attendance system protects a company from payroll fraud and provides both employer and employees with confidence in the accuracy of their wage payments all while improving productivity.

2.1.1 Manual systems

Manual systems rely on highly skilled people laboriously adding up paper cards which have times stamped onto them using a time stamping machine such as the Boundary Clock. Time stamping machines having been in use for over a century are still used by many organizations as a cheaper alternative to time and attendance software.

2.1.2 Automated systems

Automated time and attendance systems can use electronic tags, barcode badges, magnetic stripe cards, biometrics (vein reader, hand geometry, fingerprint, or facial), and touch screens in place of paper cards which employees touch or swipe to identify themselves and record their working hours as they enter or leave the work area. The recorded information is then ideally automatically transferred to a computer for processing although some systems require an operator to physically transfer data from the clocking point to the computer using a portable memory device. The computer may then be employed to perform all the necessary calculations to generate employee timesheets which are used to calculate the employees' wages. An automated system reduces the risk of errors that are common in a manual system, and allows the workforce to be more productive instead of wasting time on tedious administrative tasks.

2.2 Our methodology we used to gather ideas

2.2.1 Biometrics

Biometrics is used in the process of authentication of a person by verifying or identifying that a user requesting a network resource is who he, she, or it claims to be, and vice versa. It uses the property that a human trait associated with a person itself like structure of finger, face details etc. By comparing the existing data with the incoming data we can verify the identity of a particular person.

There are many types of biometric system like fingerprint recognition, face detection and recognition, iris recognition etc., these traits are used for human identification in surveillance system, criminal identification. Advantages of using these traits for identification are that they cannot be forgotten or lost. These are unique features of a human being which is being used widely

2.3 Face detection

Face detection is concerned with finding whether or not there are any faces in a given image (usually in gray scale) and, if present, return the image location and content of each face. This is the first step of any fully automatic system that analyses the information contained in faces (e.g., identity, gender, expression, age, race and pose) [fig 2.1]. While earlier work dealt mainly with upright frontal faces, several systems have been developed that are able to detect faces fairly accurately with in-plane or out-of-plane rotations in real time. Although a face detection module is typically designed to deal with single images, its performance can be further improved if video stream is available

Fig 2.1 Face Examples

2.3.1 Importance of Face Detection

- The first step for any automatic face recognition system
- First step in many Human Computer Interaction systems
- Expression Recognition
- Cognitive State/Emotional State Recognition
- First step in many surveillance systems
- Tracking: Face is a highly non rigid object
- A step towards Automatic Target Recognition(ATR) or generic object detection/recognition
- Video coding

2.3.2 Current state of research

- Front-view face detection can be done at >15 frames per second on 320x240 black-and-white images on a 700MHz PC with ~95% accuracy.
- Detection of faces is faster than detection of edges!
- Side view face detection remains to be difficult.

2.3.2.1 Challenges

1. Out-of-Plane Rotation: frontal, 45 degree, profile, upside down
2. Presence of beard, moustache, glasses etc.
3. Facial Expressions
4. Occlusions by long hair, hand
5. In-Plane Rotation
6. Image conditions:
    - Size
    - Lighting condition
    - Distortion
    - Noise
    - Compression

2.3.3 Different approaches of face detection

2.3.3.1 Knowledge-based methods

Encode what constitutes a typical face, e.g., the relationship between facial features.

2.3.3.2 Feature invariant approaches

Aim to find structure features of a face that exist even when pose, viewpoint or lighting conditions vary.

2.3.3.3 Template matching

Several standard patterns stored to describe the face as a whole or the facial features separately.

2.3.3.4 Appearance-based methods

The models are learned from a set of training images that capture the representative variability of faces.

2.3.3.1 Knowledge-based methods

1. Top Top-down approach: Represent a face using a set of human-coded rules.

Example:

- The center part of face has uniform intensity values.
- The difference between the average intensity values of the center part and the upper part is significant
- A face often appears with two eyes that are symmetric to each other, a nose and a mouth

2. Use these rules to guide the search process

- Level 1 (lowest resolution): apply the rule "the center part of the face has 4 cells with a basically uniform intensity" to search for candidates
- Level 2: local histogram equalization followed by edge equalization followed by edge detection
- Level 3: search for eye and mouth features for validation

2.3.3.2 Feature-Based Methods

1. Bottom-up approach: Detect facial features (eyes, nose, mouth, etc.) first
2. Facial features: edge, intensity, shape, texture, color, etc.
3. Aim to detect invariant features
4. Group features into candidates and verify them.

2.3.3.3 Template Matching Methods

1. Store a template

- Predefined: based on edges or regions

2. Deformable: based on facial contours (e.g., Snakes)
3. Templates are hand-coded (not learned)
4. Use correlation to locate faces [fig 2.2]

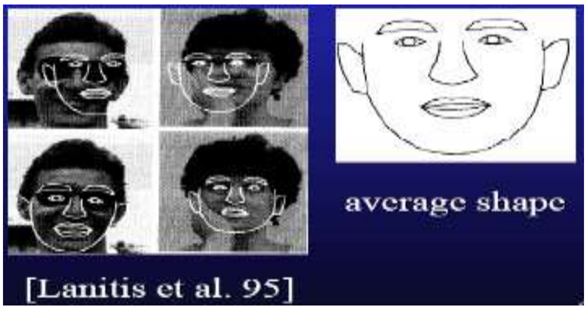


Fig 2.2 Template matching

2.3.3.4 Appearance-Based Methods

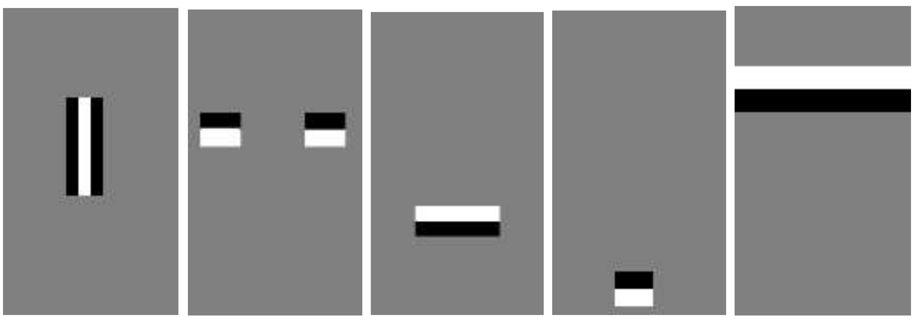
Classifiers

1. Neural network
2. Multilayer Perceptron's
3. Principal Component Analysis (PCA), Factor Analysis
4. Support vector machine (SVM)
5. Mixture of PCA, Mixture of factor analyzers
6. Distribution Distribution-based method
7. Naïve Bayes classifier
8. Hidden Markov model
9. Sparse network of winnows (SNoW)
10. Fullback relative information
11. Inductive learning: C4.5
12. Adaboost

2.3.4 Our Approach to Face Detection

As can be assumed, detecting a face is simpler than recognizing a face of a specific person. In order to be able to determine that a certain picture contains a face (or several) we need to be able to define the general structure of a face. Luckily human faces do not greatly differ from each other; we all have noses, eyes, foreheads, chins and mouths; and all of these compose the general structure of a face.

Consider the following 5 figures:



Each of these figures represents a general feature of a human face. Combining all the features together we, indeed, receive something that resembles a face.

Fig 2.3 Face Detection using Templates

By determining if each of these features is similar to some part of our picture, we can conclude if the picture contains a face or not. Notice that this does not have to be an accurate match; we just need to know if, roughly, each of these features corresponds to some part of the image. The technique used for this purpose is Template Matching.

By gathering statistics about which such features compose faces and how, we can train our algorithm to use the right features in the right positions; and thus detect faces.

Let's see an example. See in the figures below how the above features can be used to detect a face (namely, the face of President Barack Obama) [fig 2.4].

Fig 2.4 Face Detection Example

In order for this process be quick, we design it in such a way that we first check the coarse features which represent the coarse structure of a face; and only if these features match, we continue to the next iteration and use finer features. In each such iteration we can quickly reject areas of the picture which do not match a face, and keep checking those which we are not sure about. In every iteration we increase the certainty that the checked area is indeed a face, until finally we stop and make our determination.

In other words, rather than determining if the image does contain a face, we can more quickly determine if the image does not contain a face; because eliminations can be done quickly, while acceptance of faces will require more time. We call such a process a cascading process.

2.4 Face Recognition

It is a set of two task:

- **Face Identification:** Given a face image that belongs to a person in a database, tell whose image it is.
- **Face Verification:** Given a face image that might not belong to the database, verify whether it is from the person it is claimed to be in the database.

Difference between Face Detection and Recognition

- **Detection** –
  two-class classification
  Face vs. Non-face

- **Recognition**
  multi-class classification
  One person vs. all the others

2.4.1 Applications of Face Recognition

1. Access Control
2. Face Databases
3. Face ID
4. HCI - Human Computer Interaction
5. Law Enforcement
6. Multimedia Management
7. Security
8. Smart Cards
9. Surveillance
10. Others

2.4.2 Our approach to Face Recognition:

Our approach to perform the face recognition is simple. It can be broken down into the following steps:

1. Load flattened training images into array
2. Perform Histogram Equalization on training images.
3. Perform principal component analysis on the Histogram Equalized images.
4. Load test image(s)
5. Perform Histogram Equalization on test images.

6. Perform principal component analysis on Histogram Equalized test image(s)
7. Calculate Euclidian distance from test image to each training image and pick result with the smallest distance (=recognized face)

2.4.2.1 Histogram Equalization

We can consider histogram as a graph or plot, which gives you an overall idea about the intensity distribution of an image. It is a plot with pixel values (ranging from 0 to 255, not always) in X-axis and corresponding number of pixels in the image on Y-axis.

It is just another way of understanding the image. By looking at the histogram of an image, you get intuition about contrast, brightness, and intensity distribution etc. of that image. Almost all image processing tools today, provides features on histogram. Below is an image from Cambridge in Color website. [fig 2.5].

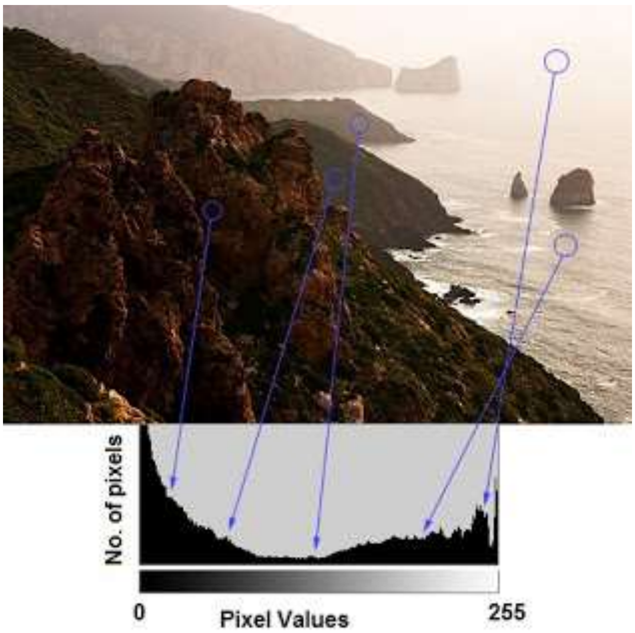


Fig 2.5 Histogram Example

We can see the image and its histogram. (Remember, this histogram is drawn for grayscale image, not color image). Left region of histogram shows the amount of darker pixels in image and right region shows the amount of brighter pixels. From the histogram, you can see dark region is more than brighter region, and amount of midtowns (pixel values in mid-range, say around 127) are very less.

Consider an image whose pixel values are confined to some specific range of values only. For example brighter image will have all pixels confined to high values. But a good image will have pixels from all regions of the image. So you need to stretch this histogram to either ends (as given in below image, from Wikipedia) and that is what Histogram Equalization does (in simple words). This normally improves the contrast of the image. [fig 2.6]

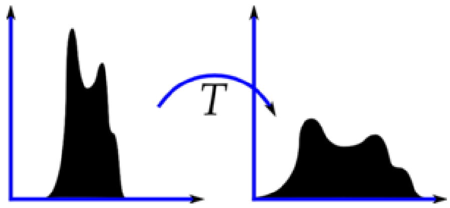


Fig 2.6 Histogram Equalization

In face recognition, before training the face data, the images of faces are histogram equalized to make them all with same lighting conditions. For Example:

- ***Original Image***

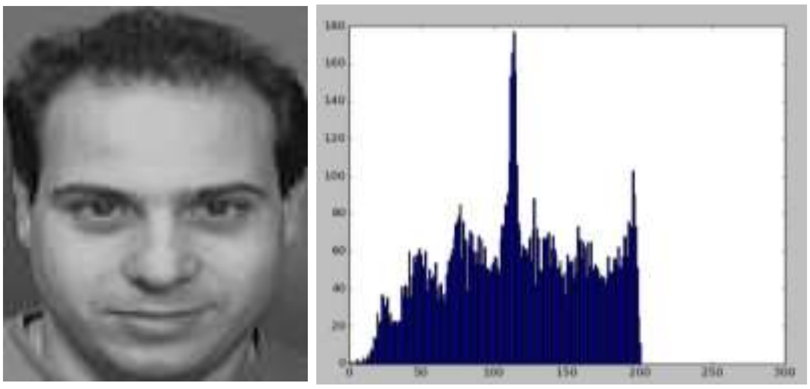


Fig 2.7 Original Image

- ***Histogram Equalized Image***

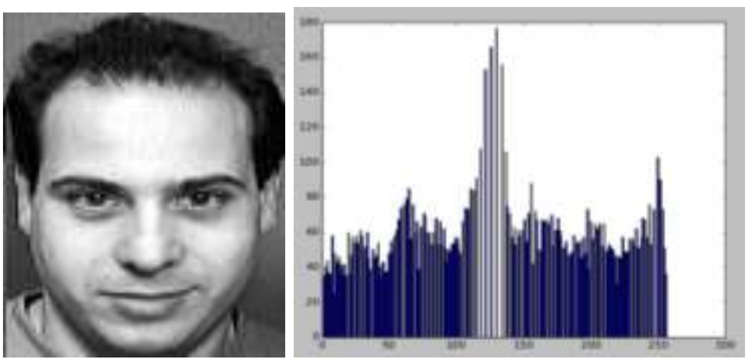


Fig 2.7 Histogram Equalized Image

2.4.2.2 Principal Component Analysis (PCA)

Principal component analysis (PCA) was invented in 1901 by Karl Pearson. PCA is a variable reduction procedure and useful when obtained data have some redundancy. This will result into reduction of variables into smaller number of variables which are called Principal Components which will account for the most of the variance in the observed variable.

Problems arise when we wish to perform recognition in a high-dimensional space. Goal of PCA is to reduce the dimensionality of the data by retaining as much as variation possible in our original data set. On the other hand dimensionality reduction implies information loss. The best low-dimensional space can be determined by best principal components.

The major advantage of PCA is using it in Eigenface approach which helps in reducing the size of the database for recognition of a test images. The images are stored as their feature vectors in the database which are found out projecting each and every trained image to the set of Eigen faces obtained. PCA is applied on Eigen face approach to reduce the dimensionality of a large data set.

2.4.2.2.1 Eigen Face Approach

It is adequate and efficient method to be used in face recognition due to its simplicity, speed and learning capability. Eigen faces are a set of Eigen vectors used in the Computer Vision problem of human face recognition. They refer to an appearance based approach to face recognition that seeks to capture the variation in a collection of face images and use this information to encode and compare images of individual faces in a holistic manner.

The Eigen faces are Principal Components of a distribution of faces, or equivalently, the Eigen vectors of the covariance matrix of the set of the face images, where an image with N by N pixels is considered a point in N 2 dimensional space. Previous work on face recognition ignored the issue of face stimulus, assuming that predefined measurement were relevant and sufficient. This suggests that coding and decoding of face images may give information of face images emphasizing the significance of features. These features may or may not be related to facial features such as eyes, nose, lips and hairs. We want to extract the relevant information in a face image, encode it efficiently and compare one face encoding with a database of faces encoded similarly. A simple approach to extracting the information content in an image of a face is to somehow capture the variation in a collection of face images.

We wish to find Principal Components of the distribution of faces, or the Eigen vectors of the covariance matrix of the set of face images. Each image location contributes to each Eigen vector, so that we can display the Eigen vector as a sort of face. Each face image can be represented exactly in terms of linear combination of the Eigen faces. The number of possible Eigen faces is equal to the number of face image in the training set. The faces can also be approximated by using best Eigen face, those that have the largest Eigen values, and which therefore account for most variance between the set of face images. The primary reason for using fewer Eigen faces is computational efficiency.

2.4.2.2.2 Eigen Values and Eigen Vectors

In linear algebra, the eigenvectors of a linear operator are non-zero vectors which, when operated by the operator, result in a scalar multiple of them. Scalar is then called Eigen value (λ) associated with the eigenvector (X). Eigen vector is a vector that is scaled by linear transformation. It is a property of matrix. When a matrix acts on it, only the vector magnitude is changed not the direction.

$AX = \lambda X$, *where A is a vector function.*

$(A - \lambda I)X = 0$, *where I is the identity matrix.*

This is a homogeneous system of equations and form fundamental linear algebra. We know a non-trivial solution exists if and only if-

$Det(A - \lambda I) = 0$, *where det denotes determinant.*

When evaluated becomes a polynomial of degree n. This is called characteristic polynomial of A. If A is N by N then there are n solutions or n roots of the characteristic polynomial. Thus there are n Eigen values of A satisfying the equation.

$AX_i = \lambda_i X_i$, *where i = 1 ,2,3,.....n*

If the Eigen values are all distinct, there are n associated linearly independent eigenvectors, whose directions are unique, which span an n dimensional Euclidean space.

2.4.2.2.3 Face Image Representation

Training set of m images of size NxN are represented by vectors of size N^2.

Each face is represented by $\Gamma_1, \Gamma_2, \Gamma_3, , \Gamma_M$.

Feature vector of a face is stored in a $N \times N$ matrix. Now, this two dimensional vector is changed to one dimensional vector.

For Example $\begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 2 \\ 1 \end{bmatrix}$

Each face image is represented by the vector Γ_i.

$$\Gamma1 = \begin{bmatrix} 1 \\ -2 \\ 1 \\ -3 \end{bmatrix} \; \Gamma2 = \begin{bmatrix} 1 \\ 3 \\ -1 \\ 2 \end{bmatrix} \; \Gamma2 = \begin{bmatrix} 2 \\ 1 \\ -2 \\ 3 \end{bmatrix} \; \ldots\ldots\ldots\ldots\ldots\ldots \Gamma M \begin{bmatrix} 1 \\ 2 \\ 2 \\ 1 \end{bmatrix}$$

2.4.2.2.4 Mean and Mean Centered Images

Average face image is calculated by

$$\Psi = (1/M) \sum_{i=1}^{M} \Gamma_i$$

$$\begin{bmatrix} 1 \\ -2 \\ 1 \\ -3 \end{bmatrix} + \begin{bmatrix} 1 \\ 3 \\ -1 \\ 2 \end{bmatrix} + \begin{bmatrix} 2 \\ 1 \\ -2 \\ 3 \end{bmatrix} + \ldots\ldots\ldots\ldots\ldots + \begin{bmatrix} 1 \\ 2 \\ 2 \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} -1 \\ -1 \\ 2 \\ -3 \end{bmatrix}$$

$$\Psi = (\Gamma_1 + \Gamma_2 + \Gamma_3 + \ldots\ldots\ldots + \Gamma_M)/M$$

Each face dIffers from the average by $\Phi_i = \Gamma_i - \Psi$ which is called mean centered image.

$$\Phi_1 = \begin{bmatrix} 2 \\ -1 \\ -1 \\ 0 \end{bmatrix} \; \Phi_2 = \begin{bmatrix} 2 \\ 4 \\ -3 \\ 5 \end{bmatrix} \; \Phi_3 = \begin{bmatrix} 3 \\ 2 \\ -4 \\ 6 \end{bmatrix} \; \ldots\ldots\ldots\ldots\ldots \Phi_M = \begin{bmatrix} 2 \\ 3 \\ 0 \\ 4 \end{bmatrix}$$

2.4.2.2.5 Covariance Matrix

A covariance matrix is constructed as:

$C = AA^T$, where $A = [\Phi_1, \Phi_2,, \Phi_M]$ of size $N^2 \times N^2$.

$$A = \begin{bmatrix} 2 & 3 \\ -1 & -2 \\ -1 & 1 \\ 0 & 2 \end{bmatrix} \quad A^T = \begin{bmatrix} 2 & -1 & -1 & 0 \\ 3 & -2 & 1 & 2 \end{bmatrix}$$

Size of covariance matrix will be $N^2 \times N^2$ (4x 4 in this case).

Eigen vectors corresponding to this covariance matrix is needed to be calculated, but that will be a tedious task therefore,

For simplicity we calculate A^TA which would be a 2 × 2 matrix in this case.

$$A^TA = \begin{bmatrix} 6 & 7 \\ 7 & 18 \end{bmatrix}$$ size of this matrix is MxM.

Consider the eigenvectors $v_i$of A^T A such that

$ATAXi = \lambda iXi$

The eigenvectors v_i of A^TA are X_1 and X_2 which are 2 × 1. Now multiplying the above equation with A both sides we get-

$AA^TAX_i = A\lambda_iX_i$

$AA^T(AX_i) = \lambda_i(AX_i)$

Eigen vectors corresponding to AA^T can now be easily calculated now with reduced dimensionality where AX_i is the Eigen vector and λ_i is the Eigen value.

2.4.2.2.6 Eigen Face Space

The Eigen vectors of the covariance matrix AA^T are AX^i which is denoted by U^i. U^i resembles facial images which look ghostly and are called Eigen faces. Eigen vectors correspond to each Eigen face in the face space and discard the faces for which Eigen values are zero thus reducing the Eigen face space to an extent. The Eigen faces are ranked according to their usefulness in characterizing the variation among the images.

A face image can be projected into this face space by

$\Omega_k = U^T(\Gamma_k - \Psi)$; k=1,....,M, where $(\Gamma_k \Psi)$ is the mean centered image.

Hence projection of each image can be obtained as Ω_1 for projection of $image_1$ and Ω_2 for projection of $image_2$ and hence forth.

2.4.2.3 Recognition Step

The test image, Γ, is projected into the face space to obtain a vector, Ω as

$$\Omega = U^T(\Gamma - \Psi)$$

The distance of Ω to each face is called Euclidean distance and defined by

$\epsilon_k^2 = ||\Omega - \Omega_k||^2$; k = 1,,M where Ω_k is a vector describing the k^th face class. A face is classified as belonging to class k when the minimum $_k$ is below some chosen threshold Θ_c. otherwise the face is classified as unknown.

Θ_c, is half the largest distance between any two face images:

$$\Theta_c = (1/2)max_{j,k} ||\Omega_j - \Omega_k||; \; j,k = 1 ,.....,M$$

We have to find the distance *between* the original test image Γ and its reconstructed image from the Eigen face Γ_f

$$\epsilon^2 = ||\Gamma - \Gamma^f||^2, \text{ where } \Gamma^f = U * \Omega + \Psi$$

If $\epsilon \geq \Theta_c$ then input image is not even a face image and not recognized.

If $\epsilon < \Theta_c$ and $\epsilon_k \geq \Theta$ for all k then input image is a face image but it is recognized as unknown face.

If $\epsilon < \Theta_c$ and $\epsilon_k < \Theta$ for all k then input images are the individual face image associated with the class vector Ω_k.

CHAPTER 3

SYSTEM DESIGN AND METHODOLOGY

3.1 Introduction

This section introduces the system requirements specification for the Automated Attendance System Using Face Detection and Recognition.

3.2 System Scope

The software is to provide employers an easier and fool-proof way to mark attendance of the employees. The software will allow administrators enroll employees. The software will allow employees have their face captured for attendance marking purposes. Only administrators will have direct manipulation rights to the system. Employees will have no other interaction with the system except for having their faces captured whilst they are entering the office. It will be a console application and would require maximum uptime due to its sensitive purpose.

3.3 Methodology

In software engineering, a software development methodology (also known as a system development methodology, software development life cycle, software development process or software process) is a division of software development work into distinct phases (or stages) containing activities with the intent of better planning and management. It is often considered a subset of the systems development life cycle. The methodology may include the pre-definition of specific deliverables and artifacts that are created and completed by a project team to develop or maintain an application.

Common methodologies include waterfall, prototyping, iterative and incremental development, spiral development, rapid application development, extreme programming and various types of agile methodology. Some people consider a life-cycle "model" a more general term for a category of methodologies and a software development "process" a more specific term to refer to a specific process chosen by a specific organization. For example, there are many specific software development processes that fit the spiral life-cycle model.

3.3.1 Prototyping

Software prototyping, is the development approach of activities during software development, the creation of prototypes, i.e., incomplete versions of the software program being developed.

The basic principles are:

- Not a standalone, complete development methodology, but rather an approach to handle selected parts of a larger, more traditional development methodology (i.e. incremental, spiral, or rapid application development (RAD)).
- Attempts to reduce inherent project risk by breaking a project into smaller segments and providing more ease-of-change during the development process.
- User is involved throughout the development process, which increases the likelihood of user acceptance of the final implementation.
- Small-scale mock-ups of the system are developed following an iterative modification process until the prototype evolves to meet the users' requirements.
- While most prototypes are developed with the expectation that they will be discarded, it is possible in some cases to evolve from prototype to working system.
- A basic understanding of the fundamental business problem is necessary to avoid solving the wrong problems.

We decided to go with **Rapid application development** (**RAD**). This is because:

- **Better Quality**. By having users interact with evolving prototypes the business functionality from a RAD project can often be much higher than that achieved via a waterfall model. The software can be more usable and has a better chance to focus on business problems that are critical to end users rather than technical problems of interest to developers.

- **Risk Control.** Although much of the literature on RAD focuses on speed and user involvement a critical feature of RAD done correctly is risk mitigation. It's worth remembering that Boehm initially characterized the spiral model as a risk based approach. A RAD approach can focus in early on the key risk factors and adjust to them based on empirical evidence collected in the early part of the process. E.g., the complexity of prototyping some of the most complex parts of the system.More projects completed on time and within budget. By focusing on the development of incremental units the chances for catastrophic failures that have dogged large waterfall projects is reduced. In the Waterfall model it was common to come to a realization after six months or more of analysis and development that required a radical rethinking of the entire system. With RAD this kind of information can be discovered and acted upon earlier in the process.

3.3.2 Rapid application development (RAD)

It is both a general term used to refer to alternatives to the conventional waterfall model of software development as well as the name for James Martin's approach to rapid development. In general, RAD approaches to software development put less emphasis on planning tasks and more emphasis on development. In contrast to the waterfall model, which emphasizes rigorous specification and planning, RAD approaches emphasize the necessity of adjusting requirements in reaction to knowledge gained as the project progresses. This causes RAD to use prototypes in addition to or even sometimes in place of design specifications. RAD approaches also emphasize a flexible process that can adapt as the project evolves rather than rigorously defining specifications and plans correctly from the start. In addition to James Martin's RAD methodology, other approaches to rapid development include agile methods and the spiral model. RAD is especially well suited (although not limited to) developing software that is driven by user interface requirements. Graphical user interface builders are often called rapid application development tools.[Fig 3.1]

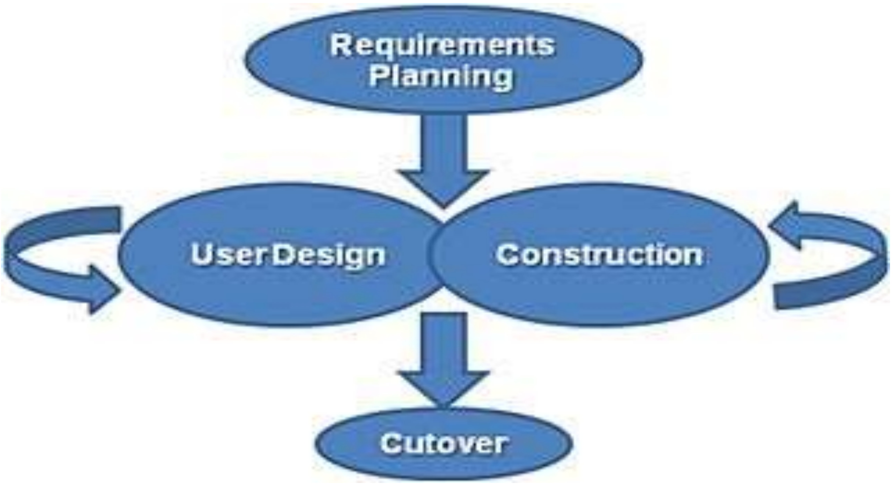


Fig 3.1 The James Martin RAD methodology

3.3.2.1 The disadvantages of RAD include:

The risk of a new approach. For most IT shops RAD was a new approach that required experienced professionals to rethink the way they worked. Humans are virtually always averse to change and any project undertaken with new tools or methods will be more likely to fail the first time simply due to the requirement for the team to learn.

- Requires time of scarce resources. One thing virtually all approaches to RAD have in common is that there is much more interaction throughout the entire life-cycle between users and developers. In the waterfall model, users would define requirements and then mostly go away as developers created the system. In RAD users are involved from the beginning and through virtually the entire project. This requires that the business is willing to invest the time of application domain experts. The paradox is that the better the expert, the more they are familiar with their domain, the more they are required to actually run the business and it may be difficult to convince their supervisors to invest their time. Without such commitments RAD projects will not succeed.
- Less control. One of the advantages of RAD is that it provides a flexible adaptable process. The ideal is to be able to adapt quickly to both problems and opportunities. There is an inevitable trade-off between flexibility and control, more of one means less of the other. If a project (e.g. life-critical software) values control more than agility RAD is not appropriate.

- Poor design. The focus on prototypes can be taken too far in some cases resulting in a "hack and test" methodology where developers are constantly making minor changes to individual components and ignoring system architecture issues that could result in a better overall design. This can especially be an issue for methodologies such as Martin's that focus so heavily on the User Interface of the system.
- Very large systems. RAD typically focuses on small to medium-sized project teams. The other issues cited above (less design and control) present special challenges when using a RAD approach for very large scale systems.

The James Martin approach to RAD divides the process into four distinct phases:

1. REQUIREMENTS PLANNING PHASE
2. USER DESIGN PHASE
3. CONSTRUCTION PHASE (RAD)
4. MAINTENANCE PHASE

3.4 Requirements Planning Phase

Combines elements of the system planning and systems analysis phases of the Systems Development Life Cycle (SDLC). Users, managers, and IT staff members discuss and agree on business needs, project scope, constraints, and system requirements. It ends when the team agrees on the key issues and obtains management authorization to continue.

3.4.1 Functional Requirements

- System must capture faces
- System must store faces in a DB attached to the employee’s ID
- System must recognize the employee to enable attendance marking

3.4.2 Non- Functional Requirements

- System shall be error-free
- System shall operate in real-time

- System should prevent data manipulation
- System should have a maximum uptime

3.5 USER DESIGN PHASE

During this phase, users interact with systems analysts and develop models and prototypes that represent all system processes, inputs, and outputs. The RAD groups or subgroups typically use a combination of Joint Application Development (JAD) techniques and CASE tools to translate user needs into working models. *User Design* is a continuous interactive process that allows users to understand, modify, and eventually approve a working model of the system that meets their needs.

3.5.1 Entity Relation Diagram

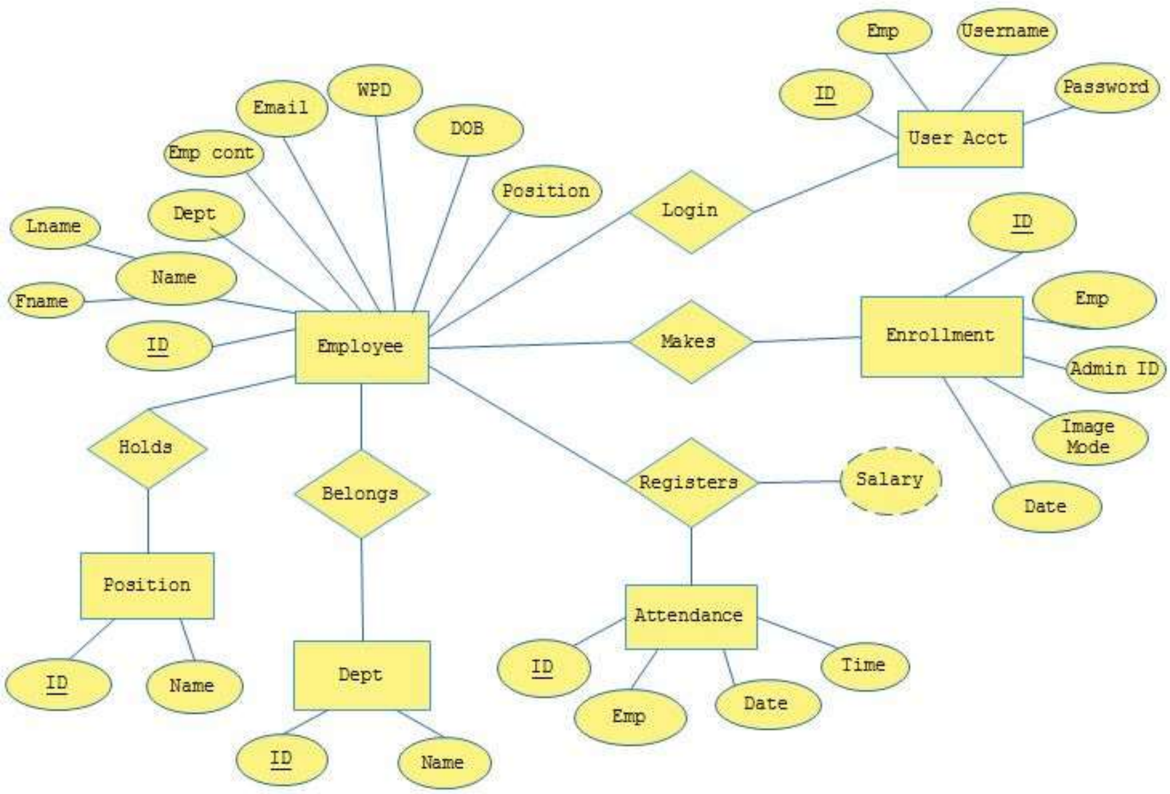


Fig 3.2 Entity Relation Diagram

An entity-relationship diagram (ERD) [Fig3.2] is a type of data modeling that show a graphical representation of objects or concepts within an information system or organization and their relationship to one another. This entity relationship diagram describes an employee belonging to a department holding the admin position logging in to a user account. He can choose to make enrollment of a new employee. An employee registers attendance and his salary calculated based on attendance and Wage per Day (WPD).

3.5.2 Use Case Diagram

The Use Case Model [Fig 3.3] describes the proposed functionality of the new system. A Use Case represents a discrete unit of interaction between a user (human or machine) and the system. A Use Case is a single unit of meaningful work; for example login to system, register with system and create order are all Use Cases. Each Use Case has a description which describes the functionality that will be built in the proposed system. A Use Case may 'include' another Use Case's functionality or 'extend' another Use Case with its own behavior.
In the use case above, the administrator adds employees to the system by enrollment. The employee therefore login by the help of the camera to detect the face. The administrator is able to have access the checked attendance, recorded attendance generate report the he or she logs out

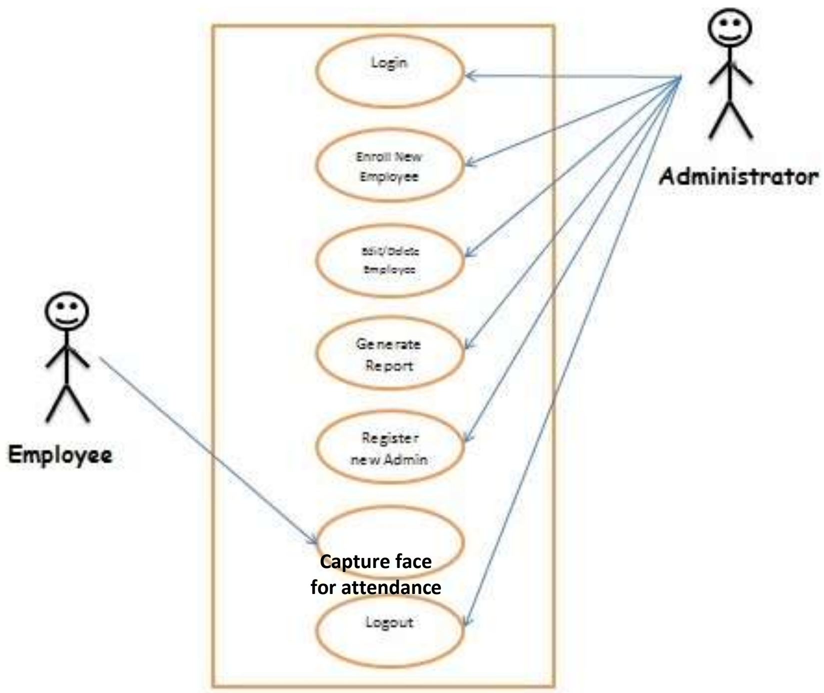


Fig 3.3 Use Case Diagram

3.5.3 System Architecture

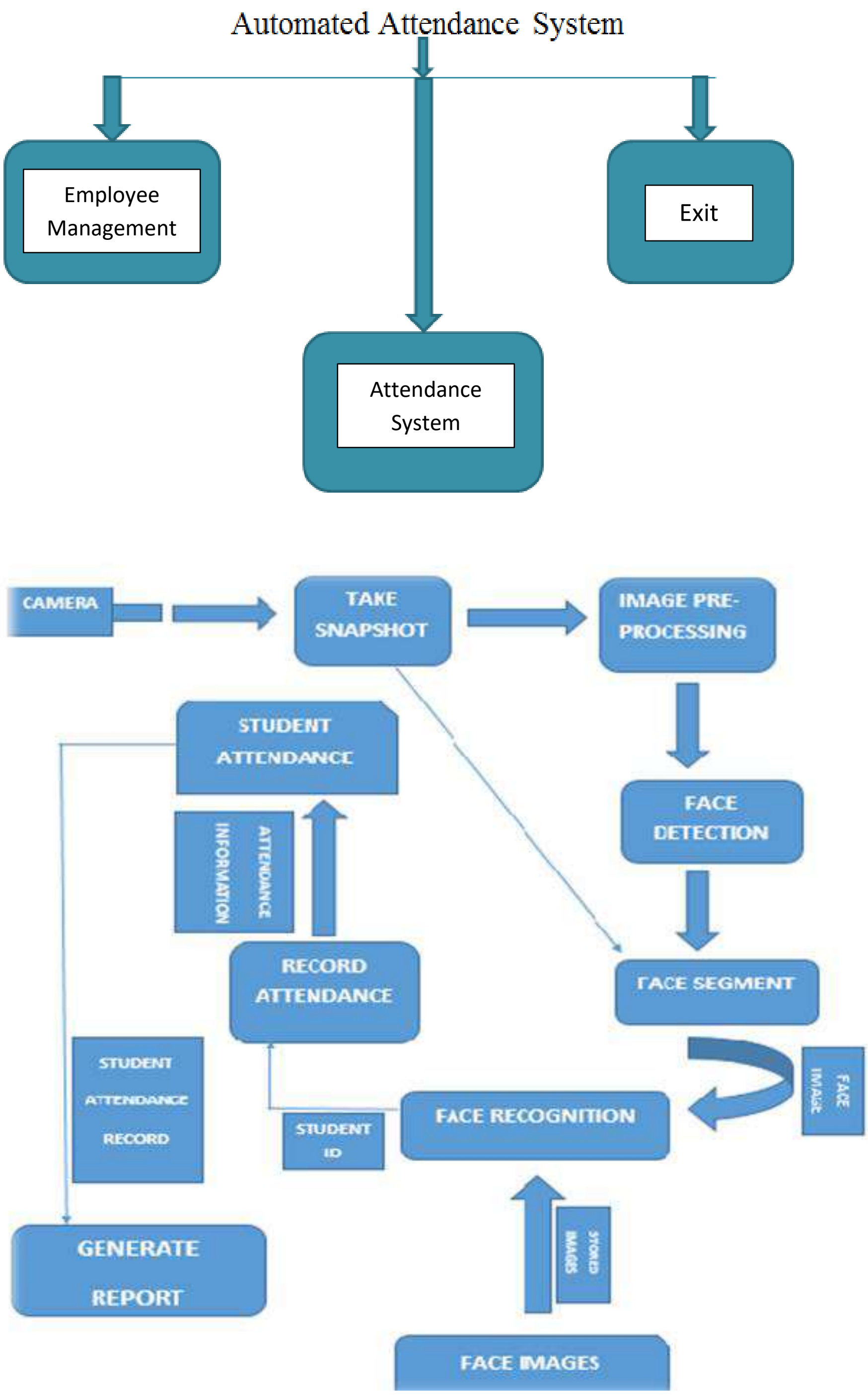


Fig 3.4 System Architecture

3.5.4 Sequence Diagram (Admin)

A sequence diagram depicts a set of actions to be performed when a user uses a system.
Here, an administrator logs in, and is being authenticated. He can update records; generate reports on attendance and payroll. He can add a new user (Employee and Admin) and can choose to repeat all these processes. On completion, he logs out of the system.

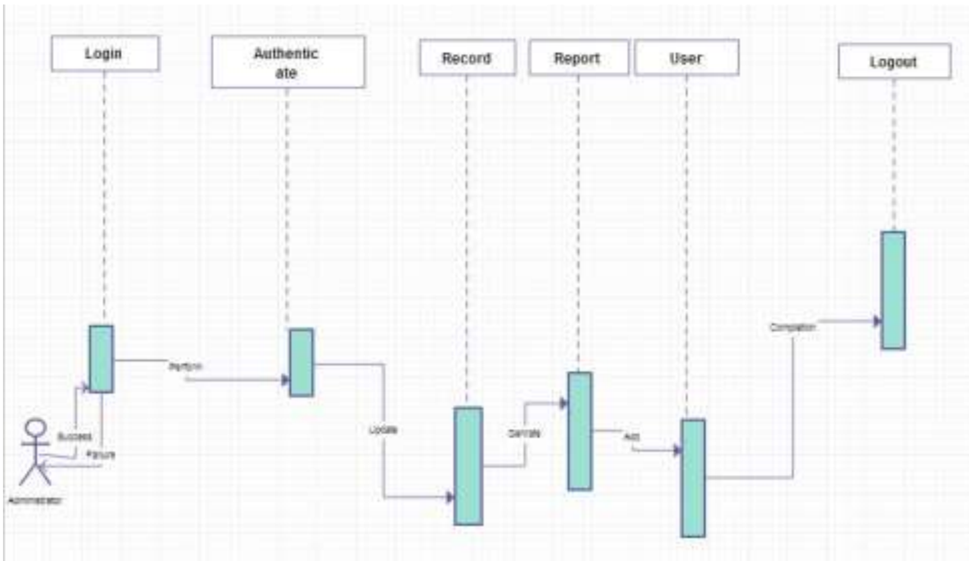


Fig 3.5.1 Sequence Diagram (Admin)

3.5.5 Sequence Diagram (Employee)

A sequence diagram depicts a set of actions to be performed when a user uses a system.
Here, an administrator logs in, and is being authenticated. He can update records; generate reports on attendance and payroll. He can add a new user (Employee and Admin) and can choose to repeat all these processes. On completion, he logs out of the system.

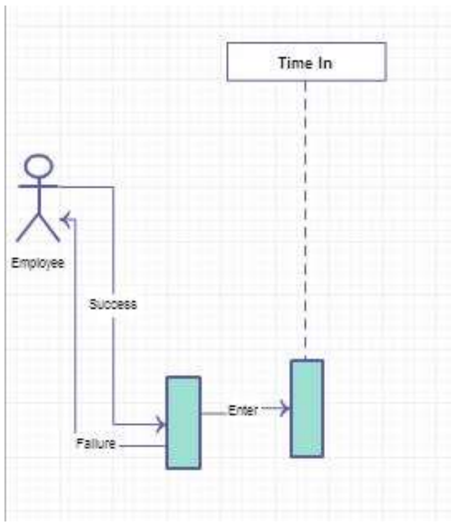


Fig 3.5.2 Sequence Diagram (Employee)

3.5.6 Database Schema

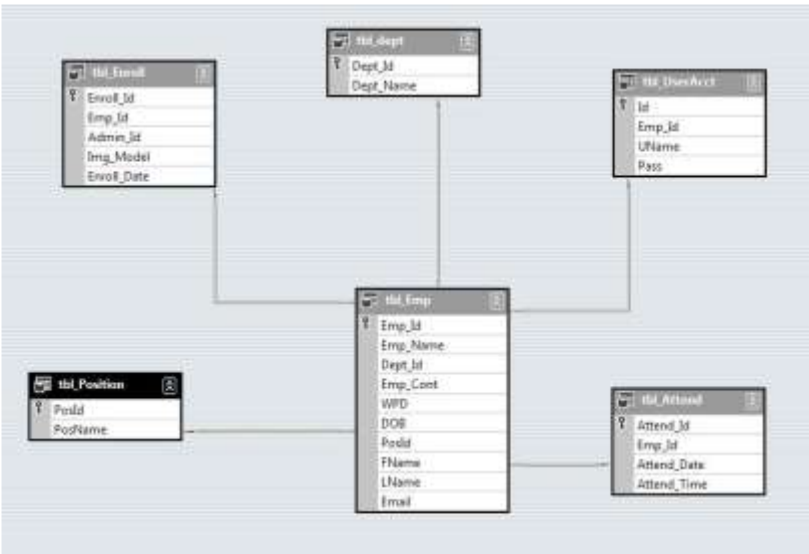


Fig 3.6 Database Schema

CHAPTER 4

IMPLEMENTATION AND RESULTS

4.1 Software and Hardware Requirements

4.1.1 Hardware Requirements

- Computer System
- Screen
- 1GHz or Greater Processor
- 1 GB RAM
- 500MB disk space

4.1.2 Software Requirements

User have to install various Software and Libraries of Python which are listed below:

I. Applications:
   1. python-2.7.9.amd64.msi
   2. opencv-2.4.10.exe
   3. PyQt4-4.11.3-gpl-Py2.7-Qt4.8.6-x64.exe
   4. py2exe-0.6.9.win64-py2.7.amd64.exe

II. Libraries:
   1. Pillow-2.1.0.win-amd64-py2.7.exe
   2. numpy-MKL-1.9.1.win-amd64-py2.7.exe
   3. scikit_learn-0.15.2-cp27-none-win_amd64.whl
   4. matplotlib-1.4.3.win-amd64-py2.7.exe
      a. pyparsing-2.0.3-py2-none-any.whl
      b. python_dateutil-2.4.2-py2.py3-none-any.whl
      c. pytz-2015.4-py2.py3-none-any.whl
      d. scipy-0.15.1-cp27-none-win_amd64.whl
      e. setuptools-17.1.1-py2.py3-none-any.whl
      f. six-1.9.0-py2.py3-none-any.whl

III. Files:
   1. haarcascade_frontalface_alt.xml

IV. Database:
   1. SQLite3

V. Others:
   1. Windows 7 or Later.

4.2 Implementation Details

4.2.1 Face Detection

4.2.1.1 Code

```
import cv2
import sys
cascPath = sys.argv[1]
faceCascade = cv2.CascadeClassifier(cascPath)
video_capture = cv2.VideoCapture(0)
while True:
 # Capture frame-by-frame
 ret, frame = video_capture.read()
 gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
 faces = faceCascade.detectMultiScale(
     gray,
     scaleFactor=1.1,
     minNeighbors=5,
     minSize=(30, 30),
     flags=cv2.cv.CV_HAAR_SCALE_IMAGE
    )
    # Draw a rectangle around the faces
    for (x, y, w, h) in faces:
        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0),
2)
    # Display the resulting frame
    cv2.imshow('Video', frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
   # When everything is done, release the capture
   video_capture.release()
   cv2.destroyAllWindows()
```

4.2.1.2 Explanation of Code

```
import cv2
import sys
cascPath = sys.argv[1]
faceCascade = cv2.CascadeClassifier(cascPath)
```

You first pass in the image and cascade names as command-line arguments. We'll use the Abba image as well as the default cascade for detecting faces provided by OpenCV.

Now we create the cascade and initialize it with our face cascade. This loads the face cascade into memory so it's ready for use. Remember, the cascade is just an XML file that contains the data to detect faces.

```
video_capture = cv2.VideoCapture(0)
```

This line sets the video source to the default webcam, which OpenCV can easily capture.

NOTE: You can also provide a filename here, and Python will read in the video file. However, you need to have ffmpeg installed for that since OpenCV itself cannot decode compressed video. Ffmpeg acts as the front end for OpenCV, and, ideally, it should be compiled directly into OpenCV. This is not easy to do, especially on Windows.

```
while True:
    # Capture frame-by-frame
    ret, frame = video_capture.read()
```

Here, we capture the video. The `read()` function reads one frame from the video source, which in this example is the webcam. This returns:

1. The actual video frame read (one frame on each loop)
2. A return code

The return code tells us if we have run out of frames, which will happen if we are reading from a file. This doesn't matter when reading from the webcam, since we can record forever, so we will ignore it.

```
ret, frame = video_capture.read()
```

```
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

faces = faceCascade.detectMultiScale(
    gray,
    scaleFactor=1.1,
    minNeighbors=5,
    minSize=(30, 30),
    flags=cv2.cv.CV_HAAR_SCALE_IMAGE
)

# Draw a rectangle around the faces
for (x, y, w, h) in faces:
    cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)

# Display the resulting frame
cv2.imshow('Video', frame)
```

This function detects the actual face – and is the key part of our code, so let's go over the options.

The detectMultiScale function is a general function that detects objects. Since we are calling it on the face cascade, that's what it detects. The first option is the grayscale image.

The second is the scaleFactor. Since some faces may be closer to the camera, they would appear bigger than those faces in the back. The scale factor compensates for this.

The detection algorithm uses a moving window to detect size of each

This function returns 4 values: the x and y location of the rectangle,

.

4.2.1.3 Snapshots of Face Detection

Fig 4.1 A colored Image

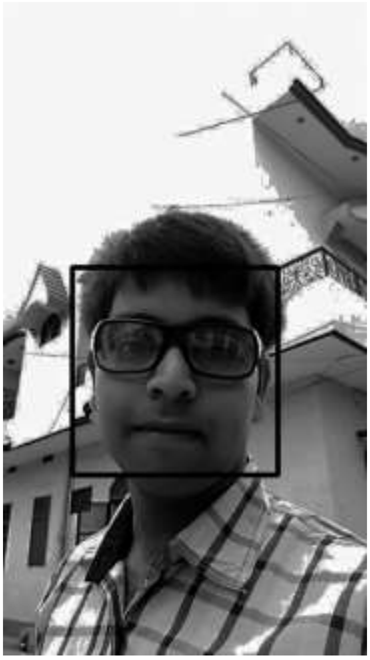


Fig 4.2 Face Detected Image

Fig 4.3 Face Cropped Image

4.2.2 Face Recognition

4.2.2.1 Code

```
from sklearn.decomposition import RandomizedPCA
import numpy as np
import glob
import cv2
import math
import os.path
import string

#function to get ID from filename
def ID_from_filename(filename):
    part = string.split(filename, '\\')
    return part[1].replace("s", "")

#function to convert image to right format
def prepare_image(filename):
    img_color = cv2.imread(filename)
    img_gray = cv2.cvtColor(img_color, cv2.cv.CV_RGB2GRAY)
    img_gray = cv2.equalizeHist(img_gray)
    return img_gray.flat

IMG_RES = 92 * 112 # img resolution
NUM_EIGENFACES = 10 # images per train person
NUM_TRAINIMAGES = 400 # total images in training set

#loading training set from folder train_faces
folders = glob.glob('train_faces/*')

# Create an array with flattened images X
# and an array with ID of the people on each image y
X = np.zeros([NUM_TRAINIMAGES, IMG_RES], dtype='int8')
y = []

# Populate training array with flattened imags from subfolders of
train_faces and names
c = 0
for x, folder in enumerate(folders):
train_faces = glob.glob(folder + '/*')
for i, face in enumerate(train_faces):
X[c,:] = prepare_image(face)
        y.append(ID_from_filename(face))
        c = c + 1

# perform principal component analysis on the images
pca = RandomizedPCA(n_components=NUM_EIGENFACES, whiten=True).fit(X)
X_pca = pca.transform(X)

# load test faces (usually one), located in folder test_faces
            test_faces = glob.glob('test_faces/*')
```

```
# Create an array with flattened images X
X = np.zeros([len(test_faces), IMG_RES], dtype='int8')

# Populate test array with flattened imags from subfolders of
train_faces
for i, face in enumerate(test_faces):
    X[i,:] = prepare_image(face)

# run through test images (usually one)
for j, ref_pca in enumerate(pca.transform(X)):
    distances = []
    # Calculate euclidian distance from test image to each of the known
images and save distances
    for i, test_pca in enumerate(X_pca):
        dist = math.sqrt(sum([diff**2 for diff in (ref_pca -
test_pca)]))
        distances.append((dist, y[i]))

    found_ID = min(distances)[1]
    print "Identified (result: "+ str(found_ID) +" - dist - " +
str(min(distances)[0])  + ")"
```

4.2.2.2 Explanation

Python is used for coding. A colored face image is converted to grey scale image as grey scale images are easier for applying computational techniques in image processing.[Fig 4.4]

Figure 4.4: A colored face image

A grey scale face image is scaled for a particular pixel size as 250x250 because many input images can be of different size whenever we take an input face for recognition.

Figure 4.5: Grey scale face image

4.2.2.3 Training Set

Database for different set of conditions is maintained. Ten different expressions for ten different people thus creating a 10x10 that is equal to 100 different set of face images. Rotated images in left and right direction and different illumination conditions are also considered while making the training set. Size variations in a input face image can also change the output therefore input images by varying their size are also taken for recognition.[Fig 4.6]

Figure 4.6: A single face image for ten different expressions

4.2.2.4 Testing Conditions

Expression- When an expression of a person is changed the orientation of face organs are changed according to it thus changing the feature vectos accordingly. Therefore changed expressions alters the recognition procedure.

Illumination- Different intensity of light on face may change the recognition just as bright light causes image saturation.

Size variation- If the size of image is varied the recognition may alter accordingly.

Figure 4.7: Image in reduced light intensity

4.2.2.5 Face Image Testing

A test image for recognition is tested by comparing to the stored dataset.

Figure 4.8: 200 × 200 image as input

Figure 4.9: 5 × 5 training set

4.2.2.6 Mean Face

Mean face is obtained by $\Psi /M)\sum_{i=1}^{M}\Gamma_i$= (1 where $\Gamma_1,\Gamma_2,\Gamma_3,\ldots,\Gamma_n$ are training set images and hence mean centered images are also evaluated by $\Phi_i = \Gamma_i - \Psi$ for further computations.[Fig 4.10]

Figure 4.10: Mean face

4.2.2.7 Eigen Face

The eigenvectors corresponding to the covariance matrix define the Eigen face which has a ghostly face like appearance and a match is found if new face is close to these images.[Fig 4.11]

Figure 4.11: Eigenface ranked according to usefulness

4.2.3 Snapshots Of Interfaces

“ATTENDANCE AUTOMATION USING FACE DETECTION AND RECOGNISATION SYSTEM” is completely Python Based Software which has built in python 2.7. Algorithm and Backend architecture is designed in Python 2.7 and Front End GUI (Graphic User Inter face) has been designed in PyQT4.(see in fig 4.12,4.13,4.14)

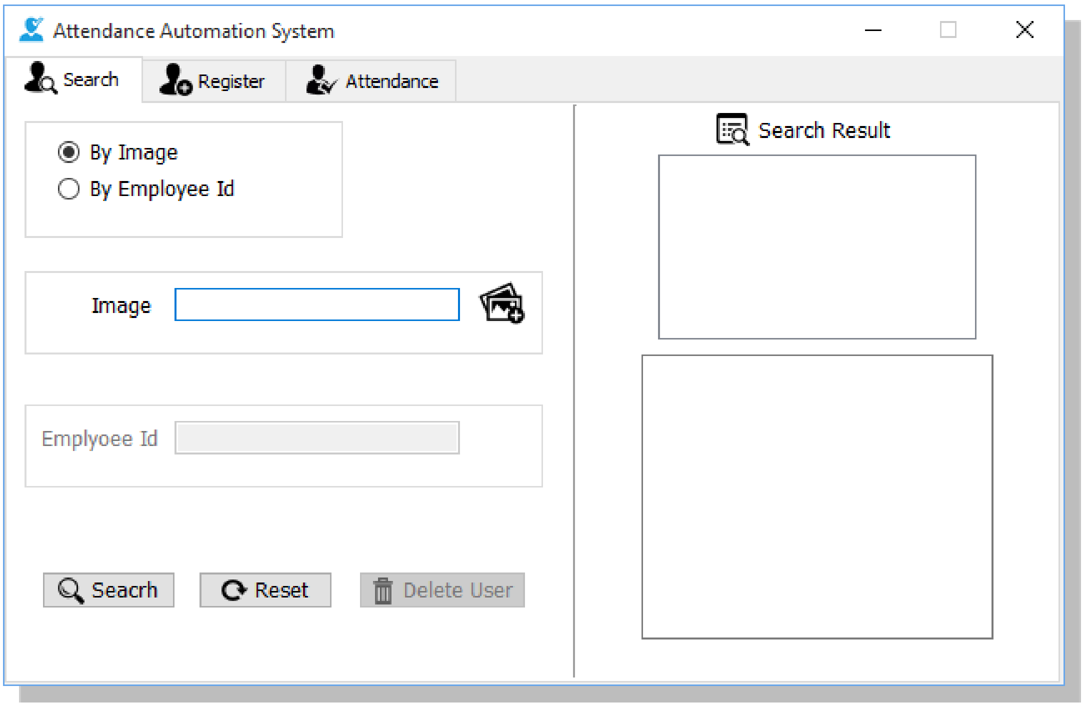


Figure 4.12 Home Screen (Search Screen)

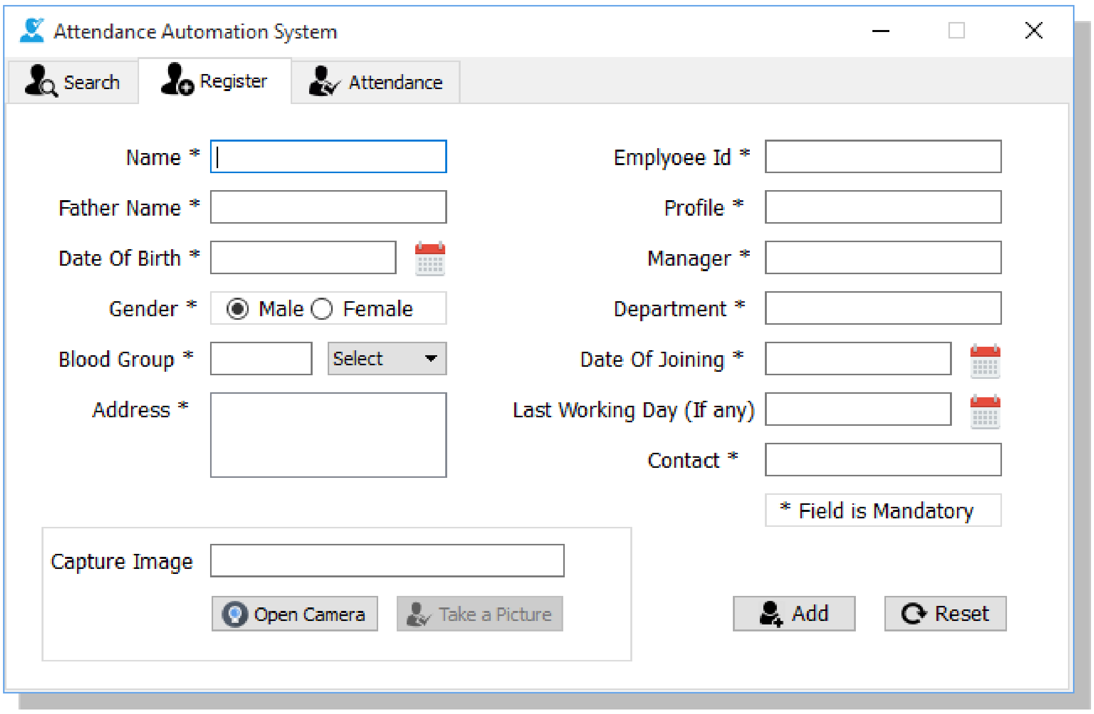


Figure 4.13 New User Screen (Register a User)

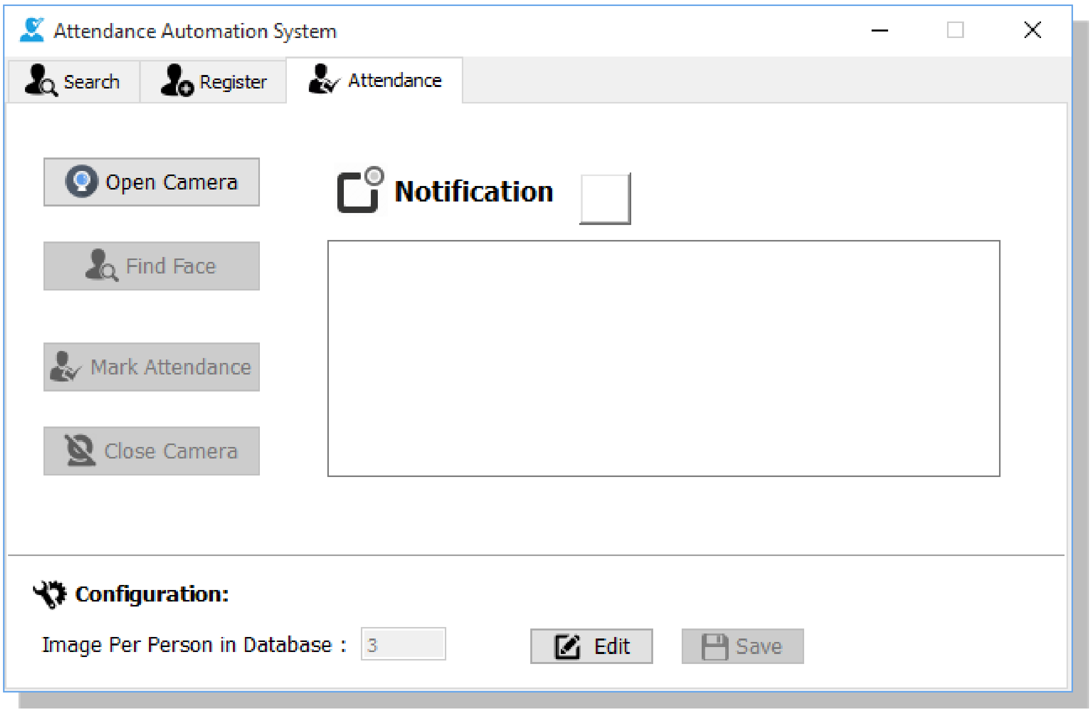


Figure 4.14 Attendance Screen

4.3 How to Use

1. By Image: User have to input the path of the image to be search in database by clicking on browse button.(see in fig 4.15)

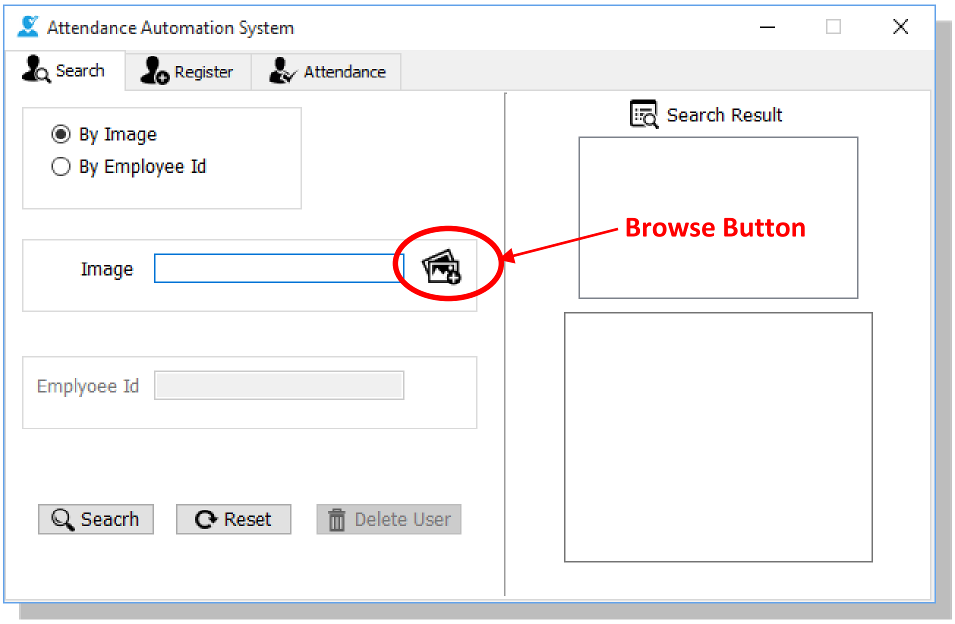
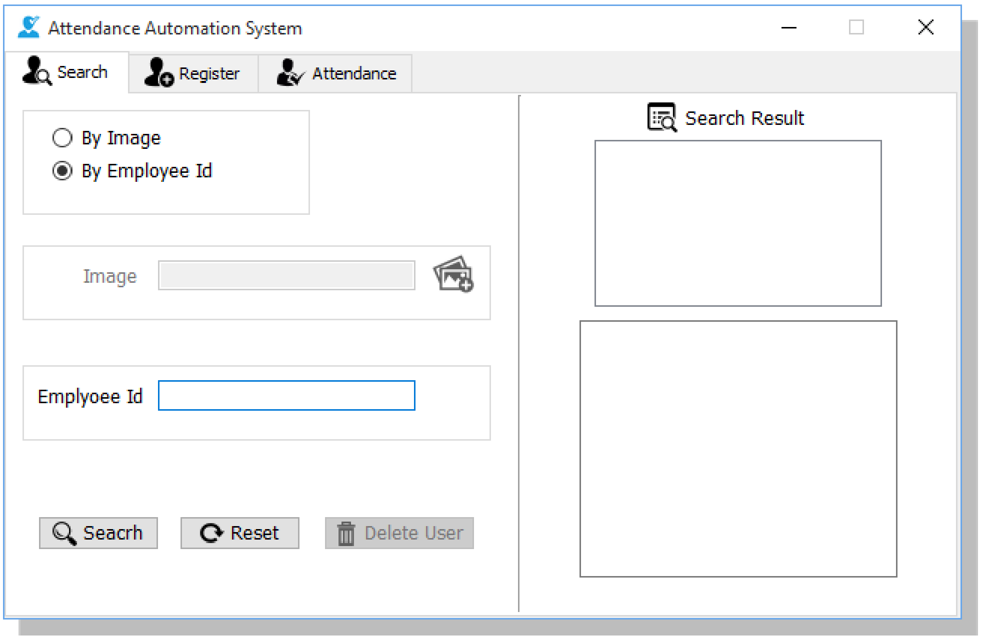
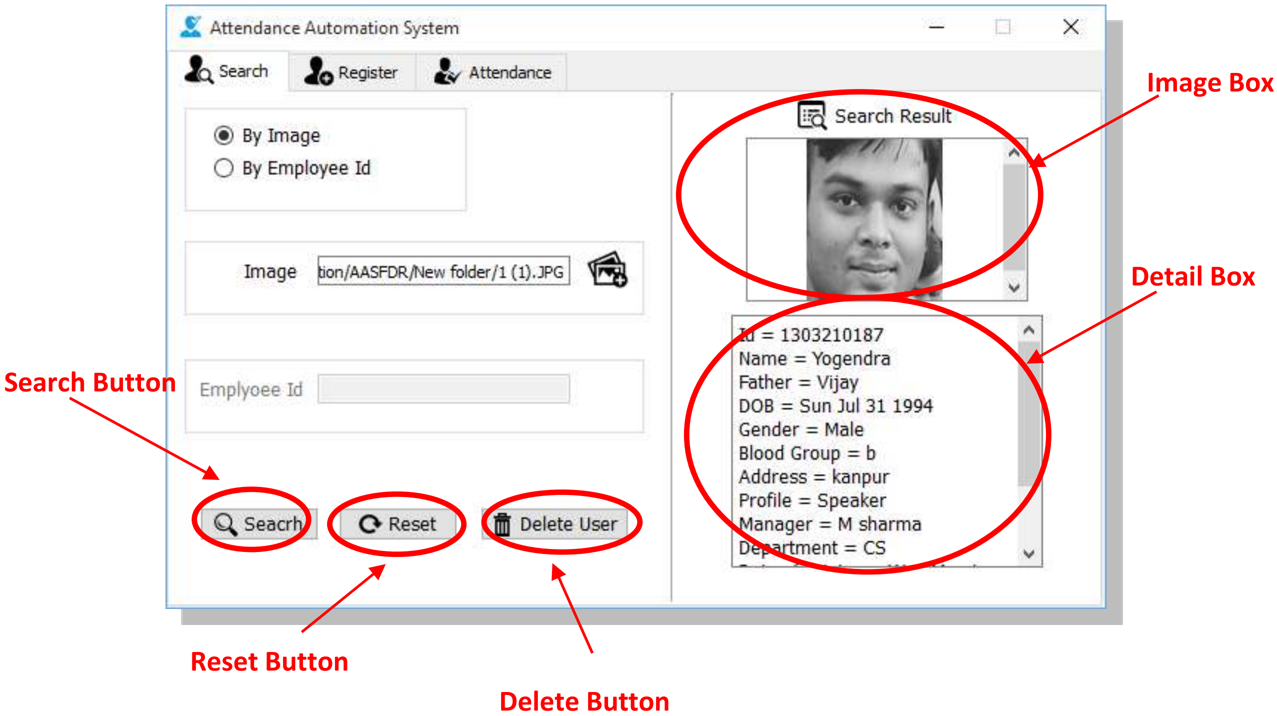


Figure 4.15 Image search

2.By Employee: User can also enter the employee id to search an employee, his details and his image by selecting By Employee ID (Radio Button) :



3.Search: After Click on Search Result will be shown in Image Box and Detail Box



2. Reset: By clicking on Reset all fields became Blank.
3. Delete: By clicking on Delete, Use will be deleted from the database.

4.New User: A new Employee can be added by entering all his details and giving the path of folder containing his/her images by clicking on Open Camera /Take a Picture button. 'Last Working Day' field is optional and all others are Mandatory.

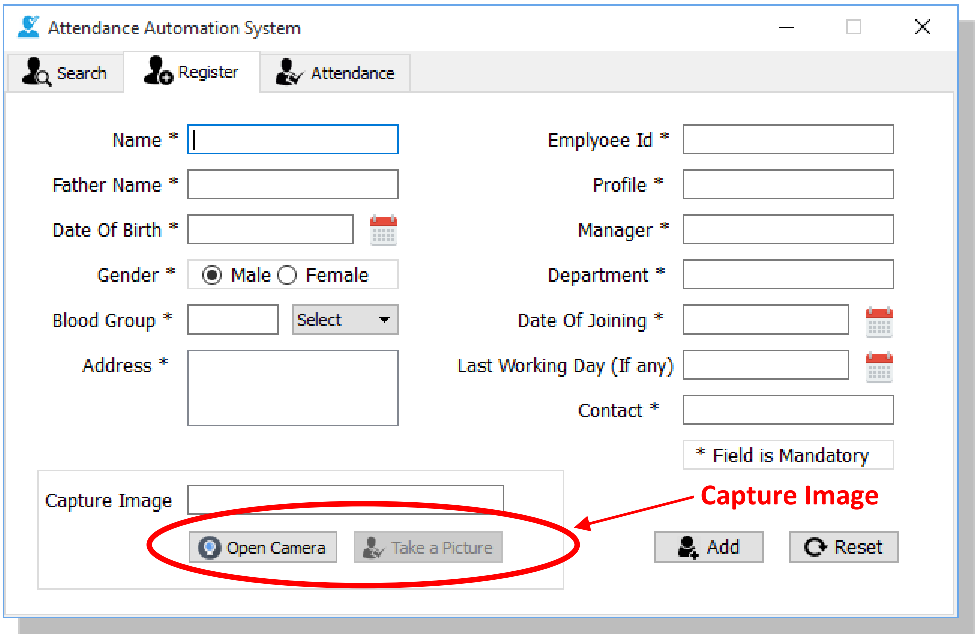
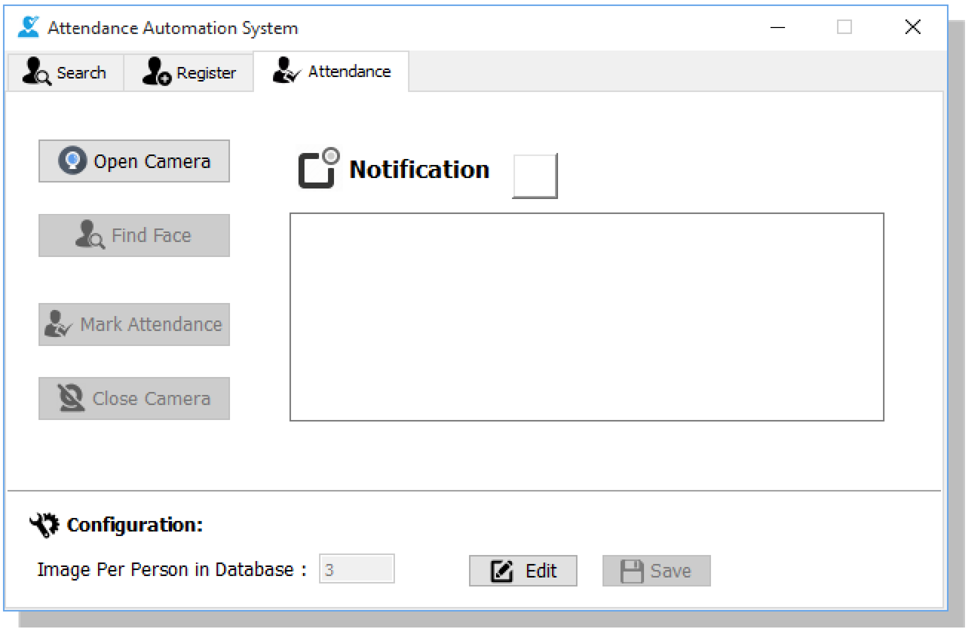


Figure 4.16 Register tab

4.3.1 Attendance System:



8. Open Camera: By clicking on this button Webcam will be open and if there would be a face in Webcam, a green rectangle would be drawn around the face in the webcam.

9. Find Face: By clicking on this button face which is appearing in webcam is searched in the data database and employee id is shown in the message box.

10. Mark Attendance: By clicking on this attendance is mark on the database with the current timestamp and employee id whose face is appearing in the webcam and message “Attendance has been Marked Successfully Employee Id = XXXXXX” is shown in Message Box.

11. Close Camera: By clicking on this button camera is closed.

12. Configuration:

13. Image per Person in Database: This is the maximum no of images any employee have in database. By default this value is 10 which can be edit using Edit and Save button. It is suggested that it should keep large (5<X<25).

4.4 Result

In this Project we implemented the face recognition system using Principal Component Analysis and Eigen face approach. The system successfully recognized the human faces and worked better in different conditions of face orientation.

We took 250 image of 25 person as a training data set. While making attendance of 25 person it correctly recognized 23 person from data set.

Thus success rate of AASFDR is

(23/25)*100 = 92%

CHAPTER 5
CONCLUSION

5.1 Summary

Attendance is a key factor in employee salary processing for every institutions. In this project, the potential benefits of implementing e-attendance environment were investigated. This is done in order to eliminate the challenges and limitations of the current automated attendance system. This project is centered on how to enhance attendance marking.

Furthermore, the essential benefit of e-attendance was highlighted. We went further by investigating how face recognition can solve the disturbing challenges manual attendance systems. Based on the knowledge derived from this project, it was concluded that implementing an automated attendance system will not only eliminate the challenges that are faced by any company but also provide a rich, effective, and more flexible environment that will have a positive effect on attendance.

5.2 Conclusion

Automated attendance systems are more efficient than manual systems as it prevents employees from falsifying entries.

5.3 Recommendation

For the purpose of growth that will lead to a rich, efficient, and result driven mode of attendance in any industry, hospitals, colleges, government sections, banks we recommend that the face recognition algorithm should be used in order to enhance the e-attendance system.

5.4 Personal Experience

The project really helped to sharpen our programming skills and SQL querying. It served as a tough challenge as writing algorithms for face detection and recognition was a huge step from our previous knowledge. It also gave an insight to our short comings and abilities.

5.5 Future Research Directions

The project is just a blueprint for implementing a full functional automated attendance system. After this blueprint, a thorough research should be carried out on this system, laying more emphasis on the impact it has on attendance and how it has enhanced company's employee-hour-output. The impact that this system have on the business sector should be studied and compared to that of the manual system

We successfully designed and implemented an automated attendance system. The entire system should be studied properly so as to detect the faults of the system and ways of improving it. Research should also be carried out on how to integrate this system into a fully functional employee management system and payroll.

5.6 Future Work

Employee Management Systems should be implemented

A leave management system should be implemented also.

References

E referencing (Internet):

1. **iFace** **“**http://www.projecttopics.info/Computer/Face_Recognition_in_e-attendance.phpAccessed: August 2016]
2. **Multiple face recognition** The Code Project, “EMGU Multiple Face Recognition using PCA and Parallel Optimization", 05 October 2016. Available http://www.codeproject.com/Articles /261550/EMGU-Multiple-FaceRecognition-using-PCA-and-Paral[Accessed: September 2016]
3. **Face detection** https://developers.keylemon.com/welcome[Accessed: June 2014]...
4. **OpenCV API reference**. Available: http://docs.opencv.org/modules/refman.html [Accessed: September 2016]
5. **Biometric fingerprints** http://www.fbi.gov/about-us/cjis/fingerprints_biometrics/biometric-center-of-excellence/modalities/facial-recognition
6. **Jump up Protefil** Group case study, “Touch screen terminals used to record staff attendance[http://www.sescoi.com/casestudies/pdf/wplan-protecmo.pdf] Retrieved 8 December 2016.
7. **Face detection** http://eyalarubas.com/face-detection-and-recognition.html [Accessed: January 2017]
8. **Webcam images**https://codeplasma.com/2012/12/03/getting-webcam-images-with-python-and-opencv-2-for-real-this-time/ [Accessed: January 2017]
9. **Facerecognitionopencv** http://www.probableanswer.com/anwsers/369259/Tutorial-Simple-Face-Recognition-OpenCV-Python [Accessed: March 2017]
10. **Histogram**http://docs.opencv.org/3.1.0/d5/daf/tutorial_py_histogram_equalization.html [Accessed: Februry 2017]

11. **PCA** http://lijiancheng0614.github.io/scikitlearn/modules/generated/sklearn.decomposition.RandomizedPCA.html[Accessed: March 2017]

12. **Eigenvalue** https://www.youtube.com/watch?v=SaEmG4wcFfg [Accessed: January 2017]

Examples of Journal Article referencing:

1. SMART ATTENDANCE USING REAL TIME FACE RECOGNITION (SMART - FR) J. G.. RoshanTharanga 1*, S. M. S. C. Samarakoon 2, T. A. P. Karunarathne 2, K. L. P. M. Liyanage 2,
2. M. P. A. W. Gamage2, D. Perera. 2
3. H. K. Ekenel, J. Stallkamp, H. Gao, M. Fischer, R. Stiefelhagen, "FACE RECOGNITION FOR SMART INTERACATINONS", interACT Research, Computer Science Department, Universit¨atKarlsruhe (TH).
4. K. Kendall, J. Kendall (2011). system analysis and design 8th edition, Pearson Education, Inc.,
5. Phillip Ian Wilson, and Dr. John Fernandez, "FACIAL FEATURE DETECTION USING HAAR CLASSIFIERS", Texas A&M University.
6. Viola, P. and M. Jones (2001). Rapid Object Detection Using a Boosted Cascade of Simple Features, in: Proc. IEEE Conf.
7. Computer Vision and Pattern Recognition.
8. JA.C. Weaver, "Biometric authentication", Computer, 39(2), pp. 96-97 (2006).
9. K. Kendall, J. Kendall (2011). system analysis and design 8th edition, Pearson Education, Inc.,
10. "Time Attendance Device Aflak Electronics".[http://aflak.com.sa/newweb/html/eng/timeAttendanceSoftware.html] Retrieved 8 December 2014.
11. H. K. Ekenel, J. Stallkamp, H. Gao, M. Fischer, R. Stiefelhagen, "FACE RECOGNITION FOR SMART INTERACATINONS", interACT Research, Computer Science Department, Universit¨atKarlsruhe (TH).

12. Phillip Ian Wilson, and Dr. John Fernandez, “FACIAL FEATURE DETECTION USING HAAR CLASSIFIERS”, Texas A&M University.

13. Viola, P. and M. Jones (2001). Rapid Object Detection Using a Boosted Cascade of Simple Features, in: Proc. IEEE Conf.

14. Computer Vision and Pattern Recognition.

15. JA.C. Weaver, “Biometric authentication”, Computer, 39(2), pp. 96-97 (2006).

16. Jump up Protefil Group case study, “Touch screen terminals used to record staff attendance[http://www.sescoi.com/casestudies/pdf/wplan-protecmo.pdf] Retrieved 8 December 2014.