Sudan University of Seince and Technology

College of Engineering

School of Electronics Engineering

Face Recognition for Blind People

التعرف علي الوجوة للاشخاص المعاقين بصريا

A research submitted in partial fulfillment for the requirements of the degree of B.Sc. (Honors) in Electronics Engineering

By:

Alargam Mohamed Ahmed Ali.

Noon Ali Nour Salih.

Osman Abd-Elaal Osman Mansour.

Supervisor:

Dr. Jacqueline John George.

September 2015

بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ

﴿ وَهُوَ الَّذِي أَنشَأَ لَكُمُ السَّمْعَ وَالْأَبْصَارَ

وَالْأَفْئِدَةَ قَلِيلًامَّاتَشْكُرُونَ ﴾

صدق الله العظيم

المؤمنون 78

To our beloved parents….

Acknowledgement

In the name of Allah, the Most Gracious and the Most Merciful Alhamdulillah, all praises to Allah, the Almighty, for the strengths and his blessing in completing this thesis.

We would like to express our sincere gratitude and thanks to our supervisor **Dr. Jacqueline John George** for her constant guidance, encouragement and support throughout the course of this thesis. The blessing, help and guidance given by her time to time shall carry us a long way in the journey of life on which we are about to embark.

We are thankful to our batch mates and friends for their support and being such a good company.

We extend our gratitude to researchers and scholars whose papers and thesis have been utilized in our project.

Finally, we thank our parents for their love, support and encouragement without which this would not have been possible.

Abstract

It is estimated that 285 million people globally are visually impaired, and without additional interventions, these numbers are predicted to increase significantly. One of the most difficult tasks faced by the visually impaired is identification of people. The inability to recognize known individuals in the absence of audio or haptic cues severely limits the visually impaired in their social interactions and puts them at risk from a security perspective.

In this thesis Matlab software was used to simulate a system that aids the blind person to recognize his family and friends facial images that are stored in a database, and if a match is found on the database, the system will announce the name of the person via speakers to the blind person. Two face recognition algorithms will be used; Principle Component Analysis (PCA), and Hidden Markov Model (HMM) to compare their performance. The simulation considered the recognition of a static facial image (photo) and a live facial image. The results showed that the PCA algorithm performs better than the HMM. It has a small recognition time and work properly under different face orientation.

المستخلص

يقدر أن 285 مليون شخص على مستوى العالم هم ضعاف البصر،و هذه الأرقام في زيادة كبيرة. واحدة من أصعب المهام التي تواجه المعوقين بصريا تحديد هوية الأشخاص. عدم القدرة على التعرف علي اشخاص معروفين في غياب الإشارات الصوتيه يحد المعاقين بصريا في تعاملاتهم الاجتماعية ويضعهم في خطر من منظور أمني.

في هذا البحث تم استخدام الماتلاب لتصميم نظام يقوم بمساعدة الاشخاص المعاقين بصريا للتعرف علي العائله و الاصدقاء عن طريق التعرف علي الوجوة المخزنه في قاعده البيانات, وعند وجود مطابقة لشخص في قاعدة البيانات سوف يقوم النظام بنطق اسم الشخص المتعرف علية للشخص المعاق بصريا عن طريق سماعات . تم استخدام خوارميتان هما خوارزمية تحليل المكونات الاساسية, و خوارزمية نموذج ماركوف الخفي لمقارنة نتائجهما. يقوم النظام بتطبيق التعرف علي صورة ثابتة و ايضا التعرف المباشر من خلال الكميرا. و النتائج توضح ان خوارزمية تحليل المكونات الاساسية تعطي نتائج افضل من نموذج ماركوف الخفي وفي زمن اقل وتعمل بصورة ناجحه للتوجهات المختلفه للوجه.

Table of Contents

Lists of Figures

Figure **Page**

Abbreviations

WHO	**W**orld **H**ealth **O**rganization
PCA	**P**rincipal **C**omponent **A**nalysis
DLM	**D**ynamic **L**ink **M**atching
SVM	**S**upport **V**ector **M**achine
OSH	**O**ptimal **S**eparating **H**yperplane
LDA	**L**inear **D**iscriminant **A**nalysis
HMM	**H**idden **M**arkov **M**odel
NNs	**N**eural **N**etworks
MCSs	**M**ultiple **C**lassifier **S**ystems
SVD	**S**ingular **V**alue **D**ecomposition
GUI	**G**raphical **U**ser **I**nterface

Chapter One

Introduction

1.1 Background

Visual impairment afflicts approximately 285 million people worldwide according to recent estimates by the World Health Organization (WHO) [1], and without additional interventions, these numbers are predicted to increase significantly. One of the many challenges faced by this population is their inability to recognize the faces of known individuals when they encounter them in their daily lives. One consequence of this is that whenever a visually impaired individual arrives in a social setting (e.g., in a conference room or at a dinner party), the conversation has to be interrupted to announce which people are already present on the scene which may result in some social awkwardness. The importance of being able to view faces in social interactions is also confirmed by several studies which indicate that most of our communication takes place not through words but via non-verbal means, the majority of which consist of facial expressions [2]. Furthermore, the ability to determine if an approaching person is a friend or a stranger is essential from a security perspective and also contributes to a person's general awareness of his context and surroundings [3]. The exponential increase in computing power per volume coupled with the decreasing size of computing elements and sensors in recent years has opened up the possibility of running computationally demanding applications on wearable electronic devices. These advances, in conjunction with the needs specified above, have fueled research into developing wearable face recognition aids for the visually impaired in the past few years. This area of research is still in its infancy with only a few prototype systems being implemented for this purpose so far.

1.2 Problem Statement

It is difficult for people who are blind and visually impaired to recognize people in a variety of social interactions. Sole reliance on voice recognition may be difficult in some circumstances, and impossible in other circumstances, e.g., people within a group who do not speak. The inability to identify people during group meetings is a disadvantage for blind people in many professional and educational situations.

1.3 Proposed Solution

This project proposes to develop a system to aid blind and visually impaired in the aforementioned types of social interactions. By using face recognition technology, the system will identify classmates and colleagues by discretely announcing their names via speakers.

1.4 Objectives

The objectives of this thesis are;

- To design a system to aid blind.
- To use face recognition algorithms.
- To evaluate and test the face recognition system.

1.5 Methodology

In this thesis Matlab software will be used to simulate a system that aids the blind person to recognize his family and friends face images that are stored in a database. When a person approaches the blind person, his face image will be captured and compared to the list of faces on the database. If a match is found on the database, the system will announce the name of the person via speakers to the blind person. Two face recognition algorithms will be used Principle Component Analysis (PCA), and Hidden Markov Model (HMM) to compare their performance.

1.6 Thesis Outline

This thesis consists of five chapters presented as follow;

Chapter Two: presents the Literature Review of biometric and biometrics component and face recognition.

Chapter Three: presents the face recognition system and its components. It also provides the algorithms used to recognize faces.

Chapter Four: Provides the simulation and discusses the results.

Chapter Five: Provides the conclusion of the work done, and a list of recommendations.

Chapter Two

Literature Review

2.1 Introduction

In the last twenty years, the computer-based facial recognition field has expanded rapidly. A facial recognition system is a computer application for automatically identifying or verifying a person from a digital image or video frame from a video source. One of the ways to do this is by comparing selected facial features from the image and a facial database [4].

Some facial recognition algorithms identify facial features by extracting landmarks, or features, from an image of the subject's face. For example, an algorithm may analyze the relative position, size, and/or shape of the eyes, nose, cheekbones, and jaw. These features are then used to search for other images with matching features other algorithms normalize a gallery of face images and then compress the face data, only saving the data in the image that is useful for face recognition. A probe image is then compared with the face data. One of the earliest successful system is based on template matching technique applied to a set of salient facial features, providing a sort of compressed face representation [5].

Recognition algorithms can be divided into two main approaches, geometric, which look at distinguishing features, or photometric, which is a statistical approach that distills an image into values and compares the values with templates to eliminate variances [6].

2.2 Related Works

There is a lot of research that had been done in this field such as;

In [7] Sinha et al. outlined nineteen basic results regarding human facial recognition, including many of the methods that humans use to identify faces. They showed that the study of human processes involved in facial recognition and the artificial algorithms being used for facial recognition systems are inextricably linked together.

In [8] Ross Beveridge evaluated the efficiency and accuracy of the major algorithms. He concluded that Principle Component Analysis (PCA), Linear Discriminant Analysis (LDA), Elastic Graph Matching (EGM), and Bayesian Intrapersonal/Extrapersonal Image Difference Classifier (BIC) were the four major baseline algorithms.

In [9] Song et al. applied a different approach on image preprocessing/enhancement. They calculated the illumination difference between the right and left part of face. If there was a large amount of difference than the mirror of the average illuminated part was considered. After preprocessing, feature extraction was done with PCA and finally classification of feature vector was completed by Euclidian Distance.

In [10] the author developed a system that uses a PCA algorithm and was designed with the visually impaired in mind. The system does not use stereo cameras, nor was it tested on visually impaired users.

In [10] A study by Krishna, Little, Black, and Panchanathan also evaluated PCA and LDA algorithms. With respect to changes in illumination and pose. The LDA and PCA algorithms were found to be superior. LDA was fastest while PCA was the most accurate.

2.3 Overview on Biometric

Biometrics refers to metrics related to human characteristics and traits. Biometrics authentication is used in computer science as a form of identification and access control. It is also used to identify individuals in groups that are under surveillance [11].

Biometric identifiers are the distinctive, measurable characteristics used to label and describe individuals. Biometric identifiers are often categorized as physiological versus behavioral characteristics. Physiological characteristics are related to the shape of the body. Examples include, but are not limited to fingerprint, palm veins, face recognition, DNA, palm print, hand geometry, iris recognition, retina and odor/scent. Behavioral characteristics are related to the pattern of behavior of a person, including but not limited to typing rhythm, gait, and voice [11].

2.4 Biometric System Components

The biometric system consists of the following main components;

Sensor: collects data and convert the information to a digital format.

Template: This is mathematical representation of biometric sample. It is used for matching. It is generated after applying certain feature extraction algorithms. The template created during enrollment phase is known as stored template while during authentication it is a live template [12].

Feature extraction: The process by which distinct features of biometric data are located is known as feature extraction [12].

Matcher: The process of matching live template with stored template to find a degree of similarity is known as matching. A score is generated based on similarity and the user is authenticated based on this score [12].

Data storage: keeps information that new biometric templates will be compared to [19].

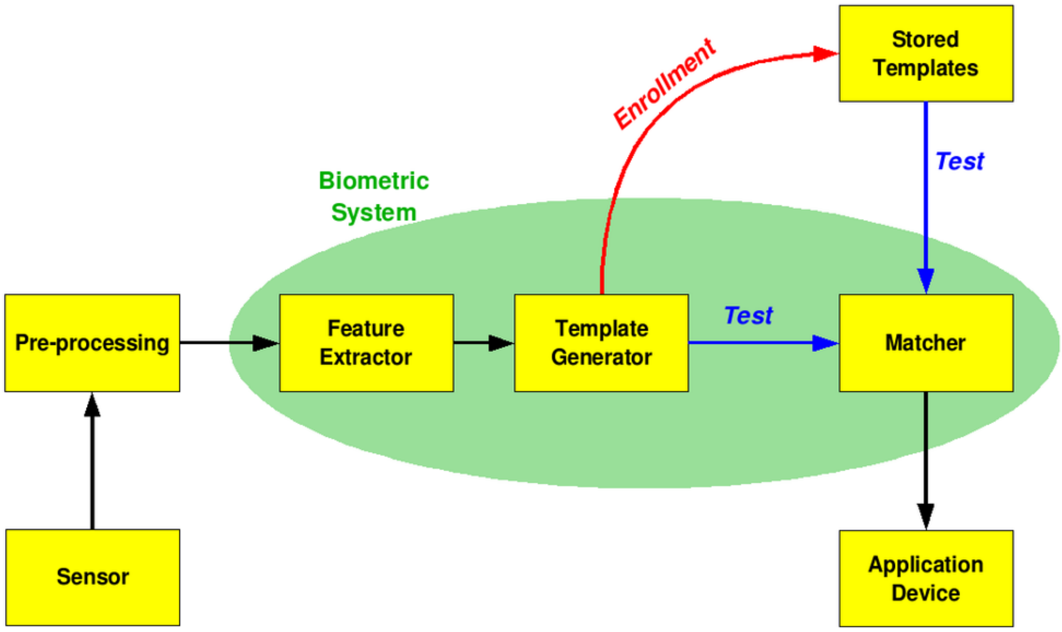


Figure 2. 1: Biometric System

In Figure 2.1, the first time an individual uses a biometric system is called enrollment [13]. During the enrollment, biometric information from an individual is captured and stored. In subsequent uses, biometric information is detected and compared with the information stored at the time of enrollment. Note that it is crucial that storage and retrieval of such systems themselves be secure if the biometric system is to be robust. The sensor is the interface between the real world and the system; it has to acquire all the necessary data. Most of the times it is an image acquisition system, but it can change according to the characteristics desired. The pre-processing performs all the necessary pre-processing: it has to remove artifacts from the sensor, to enhance the input (e.g. removing background noise), to use some kind of normalization, etc. [13].

In the feature extractor necessary features are extracted. This step is an important step as the correct features need to be extracted in the optimal way. A vector of numbers or an image with particular properties is used to create a template. A template is a synthesis of the relevant characteristics extracted from the source. Elements of the biometric measurement that are not used in the comparison algorithm are discarded in the template to reduce the file size and to protect the identity of the enrollee. During the enrollment phase, the template is simply stored somewhere (on a card or within a database or both) [13].

During the matching phase, the obtained template is passed to a matcher that compares it with other existing templates, estimating the distance between them using any algorithm (e.g. Hamming distance). The matching program will analyze the template with the input. This will then be output for any specified use or purpose (e.g. entrance in a restricted area). The selection of biometrics in any practical application depends upon the characteristic measurements and user requirements. Performance, Acceptability, Circumvention, Robustness, Population coverage, Size, Identity theft deterrence should be considered in selecting a particular biometric. Selection of biometric based on user requirement considers sensor availability, device availability, computational time and reliability, cost, sensor area and power consumption [13].

2.5 Biometric Modes

Biometric systems operate in two basic modes, verification and identification**;**

- **Verification**

Verification (or authentication) mode is when the system performs a one-to-one comparison of a captured biometric with a specific template stored in a biometric database in order to verify the individual is the person they claim to be [13] as shown in Figure 2.2. Three steps are involved in the verification of a person; first, reference models for all the users are generated and stored in the model database. Second, some samples are matched with reference models to generate the genuine and impostor scores and calculate the threshold. Finally, is the testing step where a smart card, username or ID number (e.g. PIN) may be used to indicate which template should be used for comparison. 'Positive recognition' is a common use of the verification mode [13].

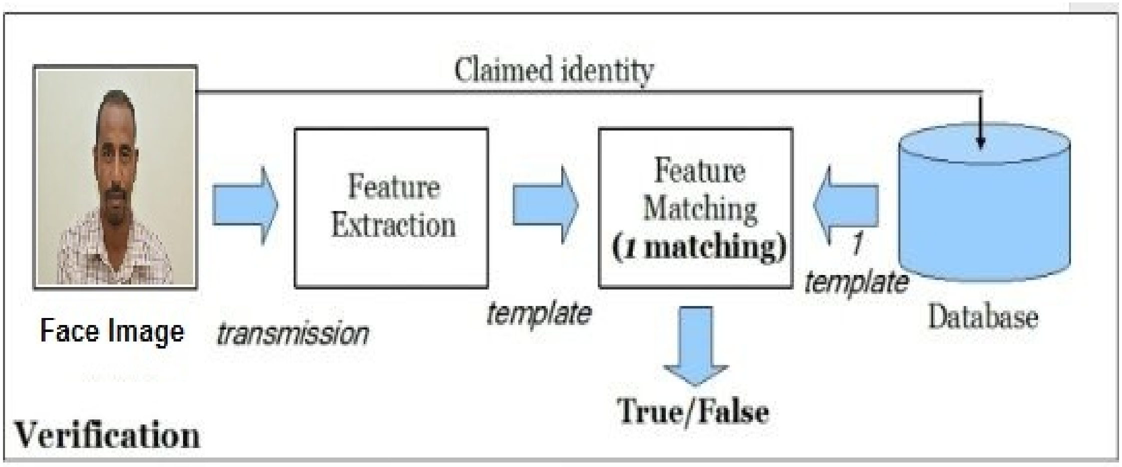


Figure 2.2: Biometric Verification

- **Identification**

In an identification mode, the system performs a one-to-many comparison against a biometric database in an attempt to establish the identity of an unknown individual [13], as shown in Figure 2.3. The system will succeed in identifying the individual if the comparison of the biometric sample to a template in the database falls within a previously set threshold. Identification mode can be used either for 'positive recognition' (so that the user does not have to provide any information about the template to be used) or for 'negative recognition' of the person "where the system establishes whether the person is who he (implicitly or explicitly) denies to be". The latter function can only be achieved through biometrics since other methods of personal recognition such as passwords, PINs or keys are ineffective [13].

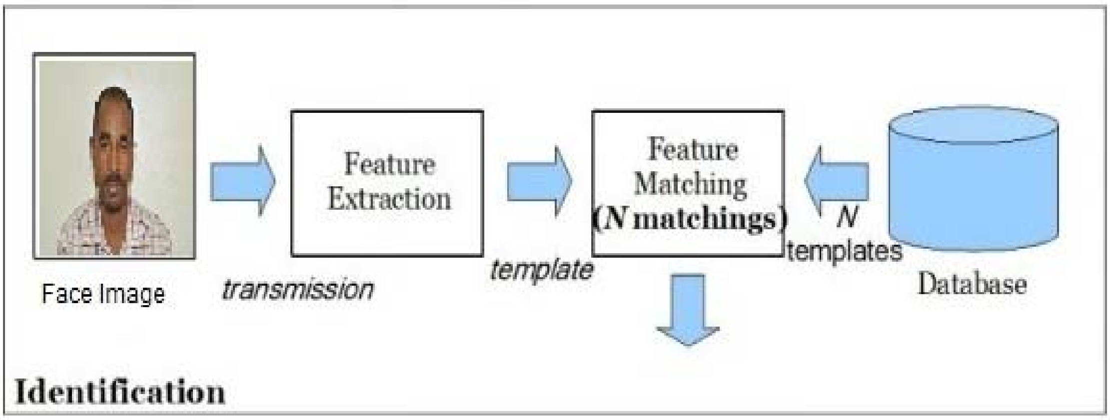


Figure 2.3: Biometric Identification

2.6 Common Biometric Modalities

Biometric systems use a number of individual physical or behavioral characteristics for identification. Each character used depends on the application environment and has its own strengths and weakness. Biometric modalities are defined as in Figure 2.4.

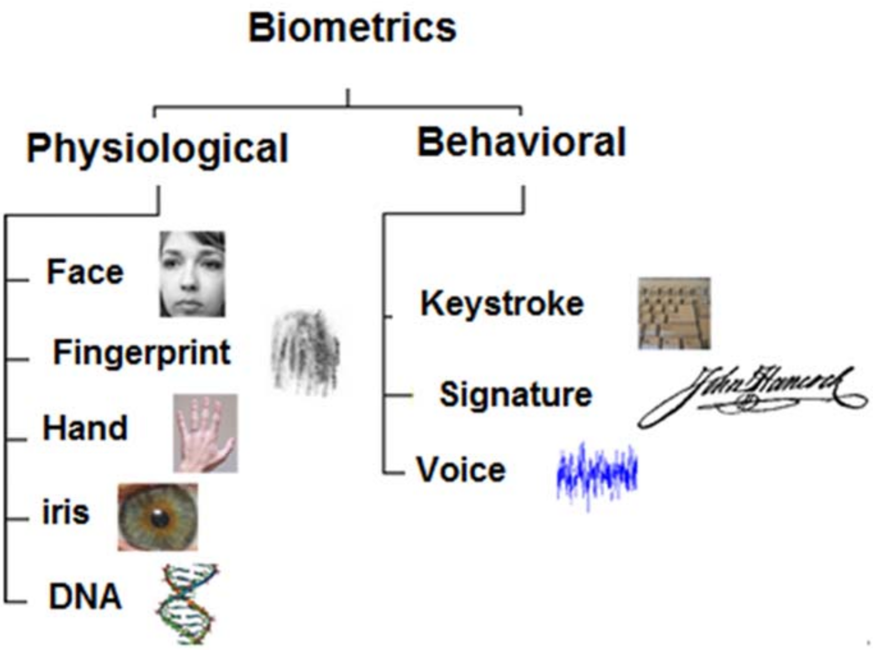


Figure 2.4: Common Biometric Modalities

Fingerprint: The fingerprint biometric is an automated digital version of the old ink-and-paper method used for more than a century for identification, primarily by law enforcement agencies. Among all the biometric techniques, fingerprint-based identification is the oldest method which has been successfully used in numerous applications. Everyone is known to have unique fingerprints [14].

A fingerprint is made of a series of ridges and furrows on the surface of the finger. The uniqueness of a fingerprint can be determined by the pattern of ridges and furrows as well as the minutiae points. Minutiae points are local ridge characteristics that occur at either a ridge bifurcation or a ridge ending.

Advantage of finger print is that; it is easy to use with some training, some systems require little space, and finger print is unique to each finger of each person. The disadvantage of finger print is that; collection of high quality images is required, and individual age and occupation may cause some sensor difficulty in capturing a complete and accurate finger print images [14].

Face Recognition and Detection: Facial recognition is an automated method to record the spatial geometry of distinguishing features of the face. Different methods of facial recognition among various vendors all focus on measures of key features. Non-cooperative behavior by the user and environmental factors, such as lighting conditions, can degrade performance for facial recognition technologies. Facial recognition has been used in projects designed to identify card counters in casinos, shoplifters in stores, criminals in targeted urban areas, and terrorists overseas [14].

Advantages of facial recognition is that; No contact is required, Commonly available sensors (cameras), Large amounts of existing data to allow background and/or watch list checks, and Easy for humans to verify results.

The disadvantages are; Faces change over time, Propensity for users to provide poor-quality video images yet to expect accurate results, and Face can be obstructed by hair, glasses, hats, scarves, etc. [14].

Iris: Working on completely different principles from retinal scanning, iris recognition is far more user friendly and offers very high accuracy. Furthermore, iris scanning has been adopted under license by certain high profile electronics companies who are able to develop good quality, interesting products and have existing marketing options for their distribution.

Advantages of Iris are; no contact required, Protected internal organ; less prone to injury, and believed to be highly stable over lifetime. The disadvantage of Iris are; difficult to capture for some individuals, easily obscured by eyelashes, eyelids, lens and reflections from the cornea, and Public myths and fears related to “scanning” the eye with a light source [14].

Voice: Voice recognition is an automated method of using vocal characteristics to identify individuals using a pass-phrase. The technology itself is not well-developed, partly because background noise affects its performance. Additionally, it is unclear whether the technologies actually recognize the voice or just the pronunciation of the pass-phrase (password) used to identify the user. The telecommunications industry and the National Security Agency (NSA) continue to work to improve voice recognition reliability. A telephone or microphone can serve as a sensor, which makes this a relatively cheap and easily deployable technology [14].

The voice recognition has a lot of advantages such as; Public acceptance, no contact required, and commonly available sensors (telephones, microphones). Its disadvantages are; difficult to control sensor and channel variances that significantly impact capabilities, and not sufficiently distinctive for identification over large databases [14].

Signature: Signature verification is an automated method of examining an individual’s signature. It uses a stylus and surface on which a person writes. This technology examines dynamics, such as speed, direction, and pressure of writing; time that the stylus is in and out of contact with the "paper"; total time of the signature; and where the stylus is raised and lowered onto the "paper." Unfortunately, a signature is one of the least reliable methods of Identification [14].

Hand geometry: Hand or finger geometry is an automated measurement of many dimensions of the hand and fingers. Neither of these methods take prints of the palm or fingers. Rather, only the spatial geometry is examined as the user lays his hand on the sensor's surface and uses guiding poles between the fingers to place the hand properly and initiate the reading [14].

The advantages of finger print that; it's easy to capture, and believed to be a highly stable pattern over the adult lifespan. The disadvantage are; its use requires some training, not sufficiently distinctive for identification over large databases, usually used for verification of a claimed enrollment identity, and the system requires a large amount of physical space [14].

Keystroke: Keystroke dynamics is an automated method of examining an individual's keystrokes on a keyboard. This technology examines such dynamics as speed and pressure, the total time of typing a particular password, and the time a user takes between hitting certain keys. This technology's algorithms are still being developed to improve robustness and distinctiveness. One potentially useful application that may emerge is computer access, where this biometric could be used to verify the computer user's identity continuously [14].

DNA: DNA identification techniques look at specific areas within the long human DNA sequence, which are known to vary widely between people. The accuracy of this technique is thus very high, and allows both identification and verification. Enrolment can be done from any cell that contains a nucleus; for instance taken from blood, semen, saliva or hair samples which is considered intrusive by many users. However, DNA as a biometric for identification uses a very small amount of non-coding genetic information which does not allow deciphering a person's initial genetic heritage [14].

2.7 Applications of biometric systems

Biometric applications fall into three main groups:

• **Commercial**: applications, such as computer network logins, electronic data security, e-commerce, Internet access, ATMs, credit cards, physical access control, cellular phones, PDAs, medical records management, and distance learning [15].

Commercial applications have used knowledge-based systems employing PINs and passwords.

• **Government:** applications such as national ID cards, correctional facilities, driver's licenses, social security, border control, passport control, and welfare-disbursement; and

Government applications have utilized systems based on tokens such as ID cards and badges [15].

• **Forensic**: applications such as corpse identification, criminal investigation, terrorist identification, parenthood determination, and missing children.

Forensic applications have relied on human experts to match biometric features [15].

Chapter Three

Face Recognition System

3.1 Introduction

Face recognition has become a very active area of research in recent years mainly due to increasing security demands and its potential commercial and law enforcement applications. The last decade has shown dramatic progress in this area, with emphasis on such applications as human-computer interaction (HCI), biometric analysis, content-based coding of images and videos, and surveillance [16]. Although a trivial task for the human brain, face recognition has proved to be extremely difficult to imitate artificially, since although commonalities do exist between faces, they vary considerably in terms of age, skin, color and gender. The problem is further complicated by differing image qualities, facial expressions, facial furniture, background, and illumination conditions [17].

Face recognition is an important research problem spanning numerous fields and disciplines. This is because face recognition, in additional to having numerous practical applications such as bankcard identification, access control, Mug shots searching, security monitoring, and surveillance system, is a fundamental human behavior that is essential for effective communications and interactions among people [18].

3.2 Face Recognition System

A face recognition system is a computer application for automatically identifying or verifying a person from a digital image or a video frame from a video source. One of the ways to do this is by comparing selected face features from the image and a facial database [19]. Figure 3.1 shows the face recognition system process.

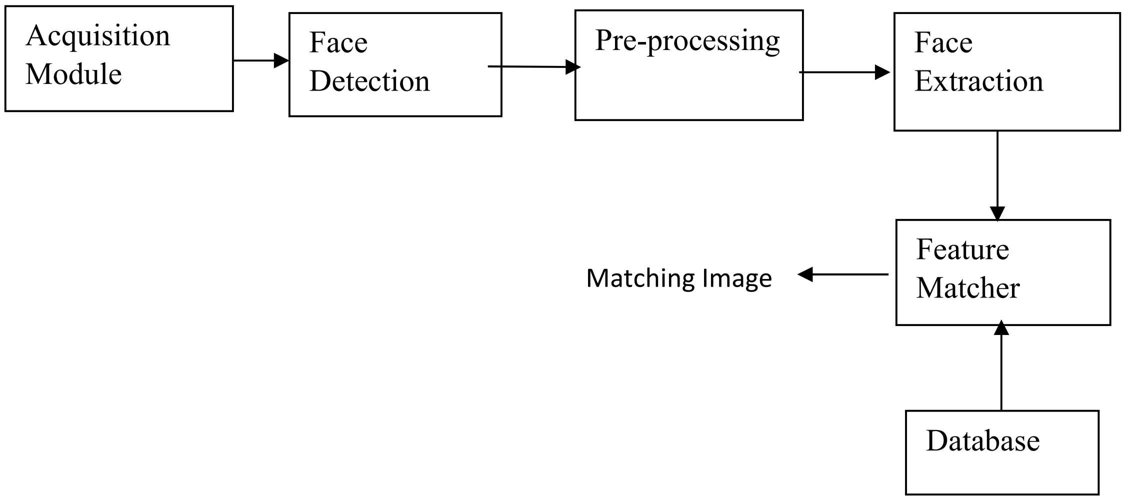


Figure 3.1 Face Recognition System

Figure 3.1 outlines the basic architecture of a face recognition system, which takes images of people, and returns the possible identity of that person.

The detection stage is the first stage; it includes identifying and locating a face in an image. The Feature Matcher is the recognition stage and it is the second stage of the system; where important information for discrimination is saved, and the matching, where the recognition result is given with the aid of a face database.

3.2.1 The Acquisition Module

This is the entry point of the face recognition process. It is the module where the face image under consideration is presented to the system. In other words, the user is asked to present a face image to the face recognition system in this module. An acquisition module can request a face image from several different environments. The face image can be an image file that is located on a magnetic disk, it can be captured by a frame grabber or it can be scanned from paper with the help of a scanner [20].

3.2.2 Face Detection

Face detection is a computer technology that determines the locations and sizes of human faces in digital images. It detects face and ignores anything else, such as buildings, trees and bodies. Face detection can be regarded as a more general case of face localization. In face localization, the task is to find the locations and sizes of a known number of faces (usually one). In face detection, face is processed and matched bitwise with the underlying face image in the database.

Face detection is the first step of face recognition system. Output of the detection can be location of face region as a whole, and location of face region with facial features (i.e. eyes, mouth, eyebrow, nose etc.) [20]. Detection methods in the literature are difficult to classify strictly, because most of the algorithms are combination of methods for detecting faces to increase the accuracy. Mainly, detection can be classified into two groups as Knowledge-Based Methods and Image-Based Methods as shown in the figure 3.2.

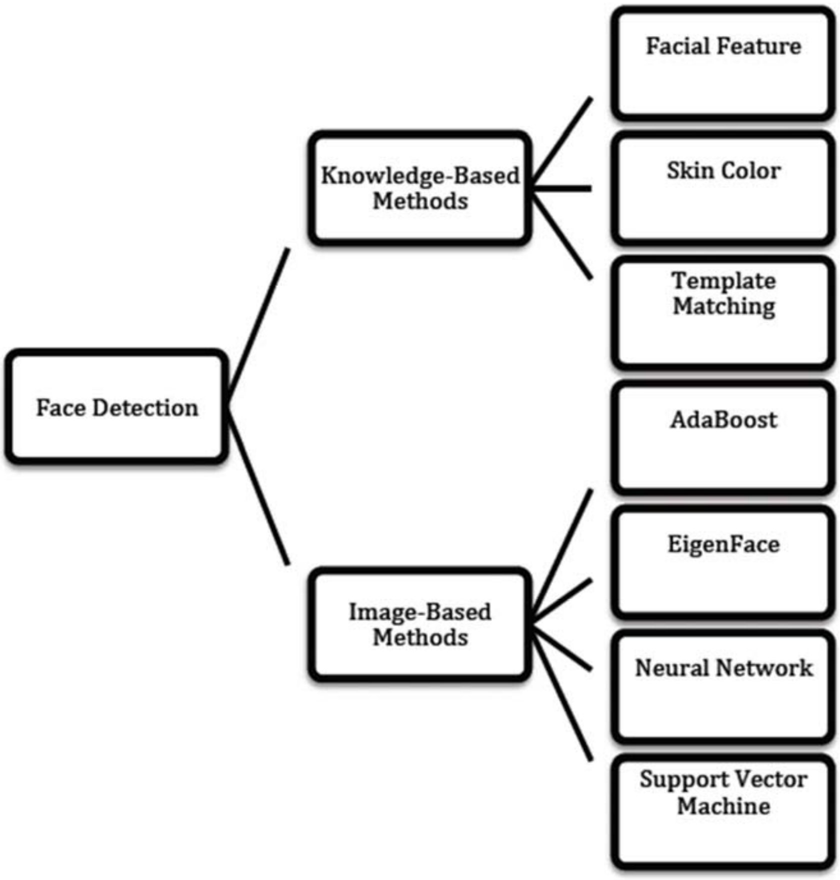


Figure 3.2: Methods for Face Detection

Knowledge-Based methods use information about Facial Features, Skin Color or Template Matching. Facial Features are used to find eyes, mouth, nose or other facial features to detect the human faces. Skin color is different from other colors and unique, and its characteristics do not change with respect to changes in pose and occlusion. Skin color is modeled in each color spaces like RGB (Red-Green-Blue), YCbCr (Luminance-Blue Difference Chroma-Red Difference Chroma), and HSV (Hue- Saturation-Value), YUV (Luminance-Blue Luminance Difference-Red Luminance Difference), and in statistical models. Face has a unique pattern to differentiate from other objects and hence a template can be generated to scan and detect faces [21].

Image-Based methods use training/learning methods to make comparison between face and non-face images. For these methods, large number of images of face and non-face should be trained to increase the accuracy of the system. AdaBoost, EigenFace, Neural Networks and Support Vector Machines are kind of methods that are used commonly in face detection algorithms [21].

Principal Component Analysis (PCA) is used to generate the feature vector of face and non-face image in EigenFace method. Also, PCA is used to compress the given information vector.

AdaBoost is an algorithm that constructs strong classifier from weak classifiers. Face candidates are found by applying AdaBoost algorithm. Then, verification is done with Cascade Classifier. This algorithm can handle of faces; left, left+45, front, and right+45, right pose [21].

Neural Networks (NNs) are good at pattern recognition and classification problem, while face and non-face images are two classes in face detection.

Support Vector Machines (SVM) is trained with face and non-face images to construct a kernel function that classifies the face and non-face images. Also, SVM can be applied to detect faces with scanning technique [21].

3.3.3 The Pre-processing Module

In this module, by means of filters, face images are normalized and if desired, they are enhanced to improve the recognition performance of the system [20].

3.3.4 The Feature Extraction Module

After performing some pre-processing (if necessary), the normalized face image is presented to the feature extraction module in order to find the key features that are going to be used for classification. This module is responsible for composing a feature vector that is well enough to represent the face image [20]. There is a lot of feature extraction module such as, Principal Component Analysis (PCA), Kernel PCA, Linear Discriminant Analysis (LDA), Semi-supervised Discriminant Analysis (SDA), and Neural Network based methods. These algorithms have formed the basis for most of the research in the area of face recognition. The statistical approaches (including PCA, LDA and BIC) work on the face image as a whole, treating each face image as a point in a multidimensional space [23].

3.3.5 Feature Matcher Methods

Feature matcher and selection algorithm's aim is to select a subset of the extracted features that cause the smallest classification error. The importance of this error is what makes feature selection dependent to the classification method used. The most straightforward approach to this problem would be to examine every possible subset and choose the one that fulfills the criterion function [23]. However, this can become an unaffordable task in terms of computational time. Some effective approaches to this problem are based on algorithms such as [23]:

- Exhaustive search: Evaluate all possible subsets of features, it is an optimal method but too complex.
- Branch and bound: Use branch and bound algorithm,
- Sequential Forward Selection: Evaluate growing feature sets (starts with best feature).

- Sequential Backward Selection: Evaluate shrinking feature sets (starts with all features).

Recently more feature selection algorithms have been proposed. Feature selection is a hard problem, so researchers make an effort towards a satisfactory algorithm, rather than an optimum one. The idea is to create an algorithm that selects the most satisfying feature subset, minimizing the dimensionality and complexity. Some approaches have used resemblance co-efficient or satisfactory rate as a criterion and quantum genetic algorithm (QGA) [23].

3.2.5 Data Base

While existing publicly-available face databases contain face images with a wide variety of poses, illumination angles, gestures, face occlusions, and illuminant colors, these images have not been adequately annotated, thus limiting their usefulness for evaluating the relative performance of face detection algorithms. For example, many of the images in existing databases are not annotated with the exact pose angles at which they were taken [18].

- **Type of Database**

AT&T (formerly ORL): Contains face images of 40 persons, with 10 images of each. For most subjects, the 10 images were shot at different times and with different lighting conditions, but always against a dark background.

The limitation of this type are, Limited number of people, Illumination conditions are not consistent from image to image, and the images are not annotated for different facial expressions, head rotation, or lighting conditions [18].

XM2VTS: Consists of 1000 GBytes of video sequences and speech recordings taken of 295 subjects at one-month intervals over a period of 4 months (4 recording sessions). Significant variability in appearance of clients (such as changes of hairstyle, facial hair, shape and presence or absence of glasses) is present in the recordings. During each of the 4 sessions a “speech” video sequence and a “head rotation” video sequence was captured. This database is designed to test systems designed to do multimodal (video + audio) identification of humans by facial and voice features [18].

This database does not include any information about the image acquisition parameters, such as illumination angle, illumination colour, or pose angle.

Oulu Physics: Includes frontal color images of 125 different faces. Each face was photographed 16 times, using 1 of 4 different illuminants (horizon, incandescent, fluorescent, and daylight) in combination with 1 of 4 different Camera calibrations (color balance settings). The images were captured under dark room conditions, and a gray screen was placed behind the participant. The spectral reflectance (over the range from 400 nm to 700 nm) was measured at the forehead, left cheek, and right cheek of each person with a Spectrophotometer. The spectral sensitivities of the R, G and B channels of the camera, and the spectral power of the four illuminants were also recorded over the same spectral range [18].

The Oulu Physics database has a lot of limitation such as although this database contains images captured under a good variety of illuminant colors, and the images are annotated for illuminant, there are no variations in the Lighting angle, and all of the face images are basically frontal (with some variations in pose angle and distance from the camera) [18].

3.3 Face Recognition Algorithms

Face recognition algorithms consist of feature extraction and feature matcher methods. There is a lot of face recognition algorithms such as, Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Semi-supervised Discriminant Analysis (SDA), Dynamic Link Matching (DLM), Support Vector Machine (SVM), and Neural Network based methods.

3.3.1 Dynamic Link Matching DLM

Dynamic Link Matching is a neural dynamics for translation invariant object recognition that is robust against distortion. Dynamic Link Matching overcomes these draw-backs by creating a set of one-tone correspondences between image and model during the matching process. This is achieved with the help of temporal binding and fast synaptic plasticity [24].

3.3.2 Support Vector Machine (SVM)

SVM is a learning technique that is considered an effective method for general purpose pattern recognition because of its high generalization performance without the need to add other knowledge. Intuitively, given a set of points belonging to two classes, a SVM finds the hyper plane that separates the largest possible fraction of points of the same class on the same side, while maximizing the distance from either class to the hyper plane. This hyper plane is called Optimal Separating Hyper plane (OSH) which minimizes the risk of misclassifying not only the examples in the training set but also the unseen example of the test set [25]. SVM can also be viewed as a way to train polynomial neural networks or Radial Basis function classifiers. The training techniques used here are based on the principle of Structure Risk Minimization (SRM), which states that better generalization capabilities are achieved through a minimization of the bound on the generalization error [26].

3.3.3 Linear Discriminant Analysis (LDA)

LDA and the related Fisher's linear discriminant are methods used in statistics, pattern recognition and machine learning to find a linear combination of features which characterizes or separates two or more classes of objects or events. The resulting combination may be used as a linear classifier, or, more commonly, for dimensionality reduction before later classification [27].

LDA is closely related to ANOVA (analysis of variance) and regression analysis, which also attempt to express one dependent variable as a linear combination of other features or measurements. ANOVA uses categorical independent variables and a continuous dependent variable, whereas discriminant analysis has continuous independent variables and a categorical dependent variable (i.e. the class label).[3] Logistic regression and probit regression are more similar to LDA, as they also explain a categorical variable by the values of continuous independent variables. These other methods are preferable in applications where it is not reasonable to assume that the independent variables are normally distributed, which is a fundamental assumption of the LDA method [28].

3.3.4 Multiple Classifier Systems (MCSs)

Recently, MCSs based on the combination of outputs of a set of different classifiers have been proposed in the field of face recognition as a method of developing high performance classification systems. Traditionally, the approach used in the design of pattern recognition systems has been to experimentally compare the performance of several classifiers in order to select the best one. However, an alternative approach based on combining multiple classifiers has emerged over recent years and represented a departure from the traditional strategy. This approach goes under various names such as MCS or committee or ensemble of classifiers, and has been developed to address the practical problem of designing automatic pattern recognition systems with improved accuracy [18].

3.3.5 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 [29]. 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 [29].

Principal component analysis (PCA) is one of the most popular methods for reducing the number of variables in face recognition. In PCA, faces are represented as a linear combination of weighted eigenvectors called as Eigen faces. These eigenvectors are obtained from covariance matrix of a training image set called as basis function. The number of Eigen faces obtained would be equal to the number of images in the training set. Eigen faces takes advantage of the similarity between the pixels among images in a dataset by means of their covariance matrix. These eigenvectors defines a new face space where the images are represented.

If there are N face images in the training set, The Gallery Set contains M images.

From a theoretical point of view, a face image Γ_i can be seen as a vector is a huge dimensional space, concatenating the columns. The algorithm steps are as follow [30]:

- **Training steps:**

Step1: prepare the training faces

Obtain face images $I_1, I_2, \ldots, I_M$ (training faces). The face images must be centered and of the same size.

Step2: prepare the data

Each face image I_i in the database is transformed into a vector and placed into a training set S.

$$S = [\Gamma_1, \Gamma_2, \ldots., \Gamma_M]$$

Each image is transformed into a vector of size MN × 1 and placed into the set. For simplicity, the face images are assumed to be of size N × N resulting in a point in N^2 dimensional space. An ensemble of images, then, maps to a collection of points in this huge space.

Step3: compute the average face vector

The average face vector (Ψ) has to be calculated by using the following formula:

$$\Psi_{train} = \frac{1}{N}\sum_{n=1}^{N} \Gamma_n \qquad 3.1$$

Step4: subtract the average face vector

The average face vector Ψ is subtracted from the original faces Γ_i and the result is stored in the variable Φ_i

$$\Phi_i = \Gamma_i - \Psi_{train} \quad , \ i = 1 \ldots. N \qquad 3.2$$

Step5: calculate the covariance matrix

The covariance matrix C is obtained in the following manner,

$$C = \frac{1}{M}\sum_{n=1}^{M} \Phi_n \ \Phi_n^T \qquad 3.3$$

$$= AA^T$$

Step6: calculate the eigenvectors and eigenvalues of the covariance matrix

The covariance matrix C in step 5 has a dimensionality of $N^2 X N^2$, so one would have N^2 eigenface and eigenvalues.

Compute the eigenvectors V_i of $L = AA^T$, where the eigenface is u_i

$$u_i = \sum_{n=1}^{N} V_i \Phi_i \quad 3.4$$

Step7: Feature Extraction

The feature weight for the training images can be calculated by the following formula:

$$\omega_i = u_i^T(\Gamma_i - \Psi_{train}) \quad , \; i = 1 \dots k \quad 3.5$$

Where, is the i^{th} Eigenfaces and i=1, 2, 3 . . . . . .K. The weight is obtained as above form a vector as follows

$$\Omega_i^T = [\, \omega_1, \omega_2, \dots, \omega_k] \quad 3.6$$

Eigenfaces with low eigenvalues can be omitted, as they explain only a small part of Characteristic features of the faces.

- **Testing steps**

Testing of the trained image is performed as follow:

a) Read the test image and separate face from it.
b) Calculate the feature vector of the test face. The test image is transformed into its eigenface components. First we compare line of our input image with our mean image and multiply their difference

with each eigenvectors. Each value would represent a weight and would be saved on a vector Ω^T

$$\omega_{test} = u_i^T(\Gamma_{test} - \Psi_{train}) \quad , \; n = 1 \ldots M \qquad 3.7$$

Where, u_i is the i^{th} Eigenfaces and i=1, 2, 3 . . . . . .K.

$$\Omega_{test}^T = [\, \omega_1, \omega_2, \ldots, \omega_k] \qquad 3.8$$

c) Compute the average distance (Euclidean distance) between test feature vector and all the training feature vectors.

Mathematically, recognition is finding the minimum Euclidean distanceϵ_k, between a testing point and a training point given in the following equation

$$\epsilon_k = \sqrt{||\Omega_{test} - \Omega_i||^2} \qquad 3.9$$

Where, i = 1, 2, 3. . . . . . K. the Euclidean distance between two weight vectors thus provides a measurement of similarity between the corresponding images.

d) The face class with minimum Euclidian distance shows similarity to test image.

3.3.6 Hidden Markov Models (HMMs)

Mathematical theory of Hidden Markov Models (HMMs) was originally described during the 1960's and early 1970's. HMM is a stochastic model which can be used to characterize statistical properties of a signal [Rab89]. An HMM is associated with a stochastic process that is not directly observable (thus hidden), but can be indirectly observed through another set of stochastic processes generating the sequence of observations [31].

Faces were intuitively divided into regions such as the eyes, nose, mouth, etc., which can be associated with the states of a hidden Markov model. Since HMMs require a one-dimensional observation sequence and images are two-dimensional, the images should be converted into either 1D temporal sequences or 1D spatial sequences [18].

HMMs are usually used to model one dimensional data but in the recent years, they have been used in vision: texture segmentation, face finding, object recognition and face recognition. Every HMM is associated with non-observable (hidden) states and an observable sequence generated by the hidden states individually [32].

The recognition process is based on a frontal face view where the facial regions like hair, forehead, eyes, nose and mouth come in a natural order from top to bottom. In the HMM the image faces is divided into seven regions which each is assigned to a state in a left to right one dimensional HMM. Figure 3.3 shows the mentioned seven face region [32].

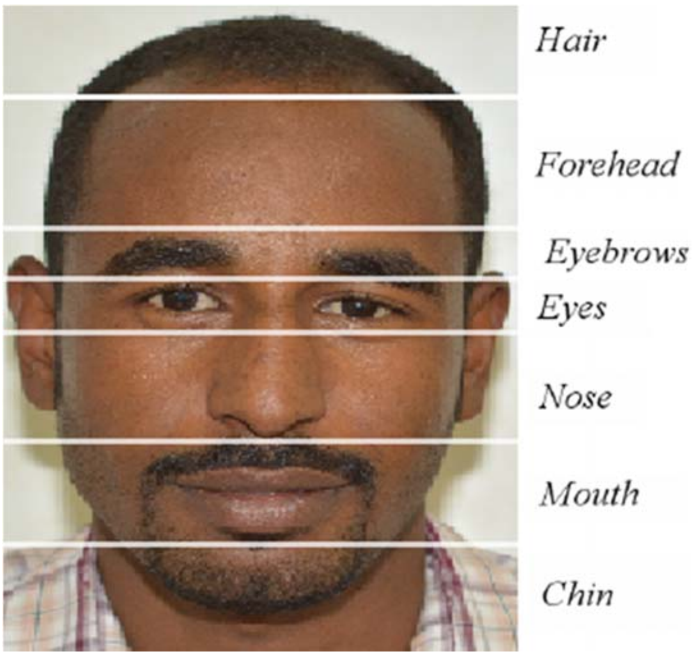


Figure 3.3: Seven regions of face coming from top to down in the natural order

Figure 3.4 shows equivalent one-dimensional HMM model for a partitioned image into seven distinct regions like figure 3.3.

In the model shown by figure 3.4, only transitions between adjacent states in a single direction (cyclic manner) is allowed. This model can be defined by the following elements:

(i) N is the number of hidden states in the model.

(ii) M is the number of observation features.

(iii) S =$[s_1, s_2, \dots, s_N]$ is the set of all possible states. The state of the model at time t is given by $q_t \epsilon S$, $1 \le t \le T$, where T is the length of the observation sequence.

(iv) A=$\{a_{ij}\}$ is the state transition probability matrix, where: $a_{ij} = P[q_{t+1} = s_j | q_t = s_i]$, $1 \le i, j \le N$ with $0 \le a_{ij} \le 1$ and $\sum_{j=1}^{N} a_{ij} = 1, 1 \le i \le N$

This cyclic HMM will segment the image into statistically similar regions, each of which is represented by a facial band.

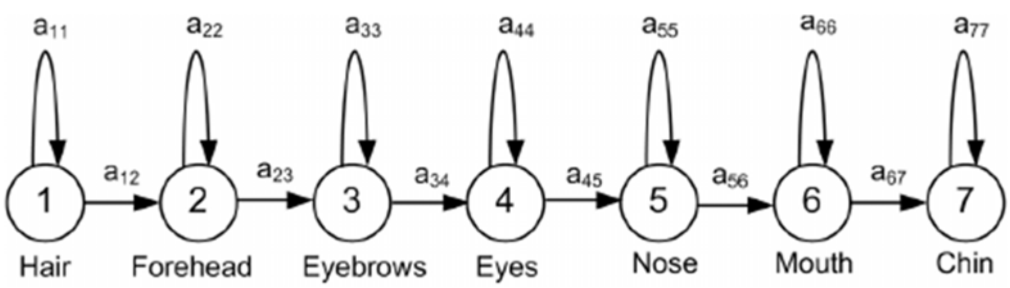


Figure 3.4: A one dimensional HMM model with seven states for a face image with seven regions.

The main advantage of the model is its simple structure and small number of parameters to adjust.

- **HMM training steps**

Step 1- Filtering

Filtering is use to improve the performance of the system, Specific filter were used which directly affects the speed and recognition rate of the algorithm. Order-statistic filters are nonlinear spatial filters. Their operations are as follows; a sliding window moves from left to right and top to down with steps of size one pixel, at each situation the centered pixel is replaced by one of pixels of the window based on the type of filter. For example minimum, maximum and median of pixels of the window may replace the centered pixel. A two dimensional order statistic filter, which replaces the centered element of a 3×3 window with the minimum element in the window, was used in the proposed system. It can simply be represented by the following equation.

$$f(x,y) = min_{(s,t)\epsilon S_{xy}}\{g(s,t)\}$$

$g(s,t)$ Is the grey level of pixel (s,t) and S_{xy} is the mentioned window.

Step 2- The Observation Vectors

The observation sequence is generated by dividing each face image of width W and height H into overlapping blocks of height L and width W. A L×W window is slid from top to bottom on the image and creates a sequence of overlapping blocks. The number of blocks extracted from each face image is given by:

$$T = \frac{\text{H}-\text{L}}{\text{L}-\text{P}} + 1 \qquad 3.10$$

Where P is overlap size of two consecutive blocks. A high percent of overlap between consecutive blocks significantly increases the performance of the system consequently increases the computational complexity.

Step 3- Feature Extraction

Feature extraction is used to achieve high recognition rate. It depend on Singular Value Decomposition (SVD) coefficients.

The Singular Value Decomposition (SVD) has been an important tool in signal processing and statistical data analysis. Singular values of given data matrix contain information about the noise level, the energy, the rank of the matrix, etc. As singular vectors of a matrix are the span bases of the matrix, and orthonormal, they can exhibit some features of the patterns embedded in the signal. SVD provides a new way for extracting algebraic features from an image [32].

A singular value decomposition of a m×n matrix X is any function of the form:

$$X = U\Sigma V^T$$

Where U (m×m) and V (m×m) are orthogonal matrix, and Σ is and m×n diagonal matrix of singular values with components $\sigma_{ij} = 0, i \neq j$ and $\sigma_{ij} > 0.$ Furthermore, it can be shown that there exist non-unique matrices U and V such that $\sigma_1 \geq \sigma_2 \ldots \geq 0.$ The columns of the orthogonal matrices U and V are called the left and right singular vectors respectively; an important property of U and V is that they are mutually orthogonal. The main theoretical property of SVD relevant to face image recognition is its stability on face image. Singular values represent algebraic properties of an image.SVD is a robust feature extraction technique for face images [32].

Step 4- Quantization

The SVD coefficients have innately continuous values. These coefficients build the observation vectors. If they are considered in the same continuous type, the infinite number of possible observation vectors that can't be modeled by discrete HMM will be encounter.

$$\Delta_i = \frac{X_{imax} - X_{imin}}{D_i} \qquad 3.11$$

X_{imax} and X_{imin} are the maximum and minimum values that X_i gets in all possible observation vectors respectively.

Knowing $\Delta_i$component X_i will be replaced with its quantized value computed as below:

$$X_{iQuantized} = \frac{X_i - X_{imin}}{\Delta_i} \qquad 3.12$$

Thus all components of X will be quantized. At last each quantized vector is associated with a label that here is an integer number

$$label = q_{t1} * 10 * 7 + q_{t2} * 7 + q_{t3} + 1$$

Where q_{t1}, q_{t2}, q_{t3} are the quantized values. Note that if the coefficients are all zero the label will be 1 and if they are 18, 10 and 7, the label will have the maximum value of 1260.

So each block of image is mapped to an integer.

Considering all blocks of an image, the image is mapped to a sequence of integer numbers that is considered as an observation vector.

Step 5- Modeled by a seven -state HMM

A HMM is trained for each person in the database using the Baum-Welch algorithm. At the first step $\lambda = (A,B,\pi)$ is initialized. The initial values for A and π are set due to the left to right structure of the face model. The initial values for A and π are as follows:

$$a_{i,i} = a_{i,i+1} = 0.5 \quad 1 < i < 7 \quad a_{7,7} = 1, \pi_0 = 1$$

Initial estimates of the observation probability matrix B are obtained as following:

$$B = \frac{1}{M} \text{Ones}(N, M) \qquad 3.13$$

Where M is the number of all possible observation symbols obtained from quantization procedure and N is the number of states. After representing the images by observation vectors, the parameters of the HMM are estimated using the Baum-Welch algorithm which finds

$$\lambda^* = \max_\lambda \ p(O|\lambda)$$

In the computed model the probability of the observation O associated to the learning image is maximized.

- **HMM Testing steps:**

After learning process, each class (face) is associated to a HMM. For a K-class classification problem, a K distinct HMM models were found. Each test image experiences the block extraction, feature extraction and quantization process as well. Indeed each test image like training images is represented by its own observation vector [32].

Calculate the probability of the observation vector (current test image) given each HMM face model for incoming face images. A face image m is recognized as face d if:

$$p\left(O^{(m)}\middle|\lambda_d\right) = \max_n p\left(O^{(m)}\middle|\lambda_d\right) \qquad 3.14$$

3.4 Application of Face Recognition

The application of face recognition technique can be categorized into two main parts: law enforcement application and commercial application. Face recognition technology is primarily used in law enforcement applications, especially Mug shot albums (static matching) and video surveillance (real-time matching by video image sequences) [18].

The commercial applications, range from static matching of photographs on credit cards, ATM cards, passports, driver's licenses, and photo ID to real-time matching with still images or video image sequences for access control. Each application presents different constraints in terms of processing [18].

Chapter Four

Results and Discussion

4.1 Introduction

Visual impairment afflicts approximately 285 million people worldwide according to recent estimates by the World Health Organization (WHO) [1], and without additional interventions these numbers are predicted to increase significantly. One of the many challenges faced by this population is their inability to recognize the faces of known individuals when they encounter them in their daily lives.

The importance of being able to view faces in social interactions is also confirmed by several studies which indicate that most of our communication takes place not through words but via non-verbal means, the majority of which consist of facial expressions [2].

4.2 System Block Diagram

Figure 4.1 shows the architecture of a face recognition system for a visually impaired person. The system detects the face of the person approaching the blind person by a camera. The detected face will be processed by one of the face recognition algorithms either PCA or HMM to find if it belongs to the database of friends and family or not. After the recognition process the system will announce to the blind person the name of that person.

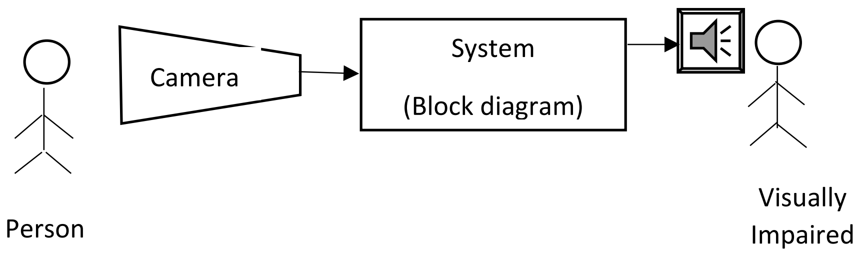
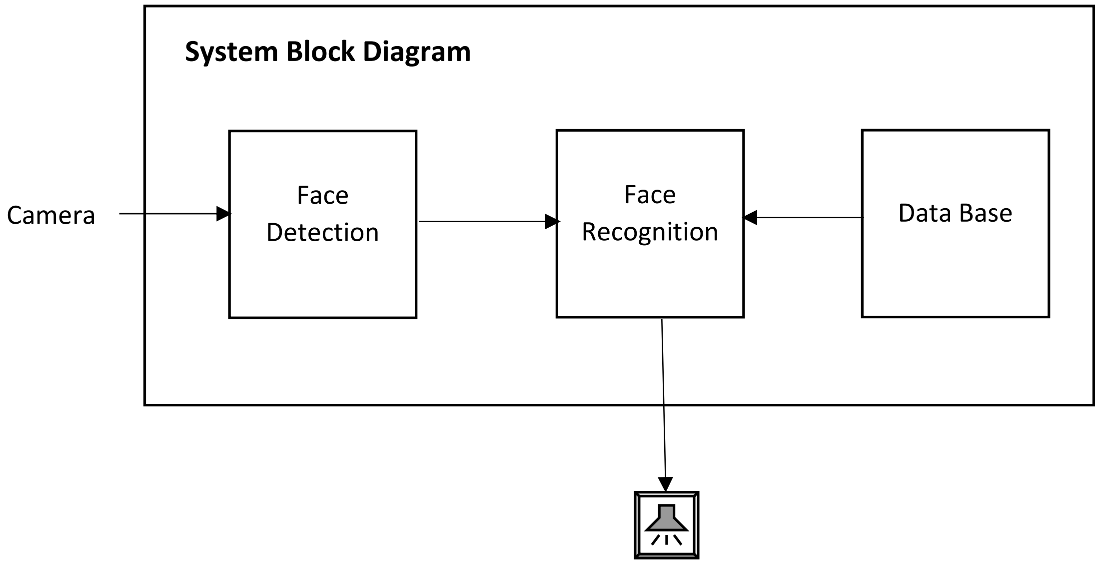


Figure 4.1: System Block Diagram



Speaker

Figure 4.2: System Components

4.3 System Database

Two databases where designed for each algorithm; the PCA and the HMM. They both differ in size and in the way the face was cropped.

- **PCA Databases**

Two databases were designed for both static facial images and live facial images. Figure 4.3 shows the static facial image database. It consist of 6 facial images of 3 people. Each person has two images having different facial expression. The images are all of size 180X200 pixels.

Figure 4.4 shows the live facial images database. It consist of 3 facial images of 3 people captured be PC webcam. Each person has one image, and the images are all size of 180X200 pixels.

Figure 4.3: PCA static Database

Figure 4.4: PCA live Database

- **HMM Database**

Two databases were designed for both static facial images and live facial images. Figure 4.5 shows the static facial image database. It contains 6 facial images of 3 people. Each person has two facial images having different facial expression. The images have different sizes of pixels depending on the manual cropping done on the image to represent the faces.

Figure 4.6 shows the live facial images database. It consist of 3 facial images of 3 people captured be PC webcam. Each person has one image, and the images are all size of 180X200 pixels. This database designed because the camera resolution is not high enough to recognize facial image from static database.

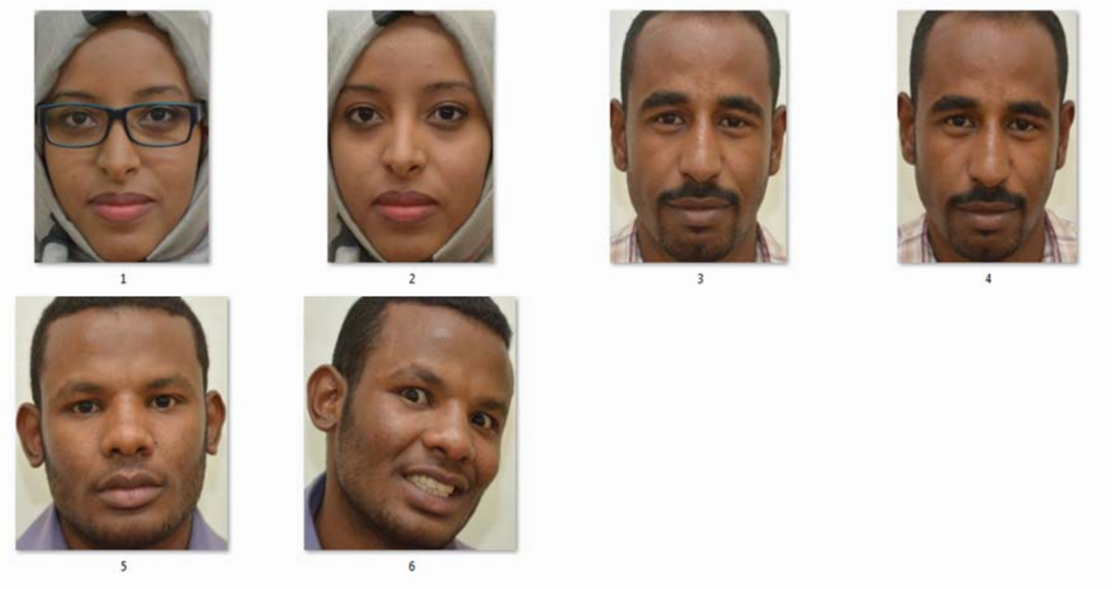


Figure 4.5: HMM static Database

Figure 4.6: HMM live Database

4.4 System Flow Charts

4.4.1 PCA flow chart

The flowchart of the PCA algorithm is divided into two parts; the training process and the testing process.

- **Training Process Flow chart**

The PCA algorithm training process starts by preparing the facial training database. This process converts the facial images in the database into matrices. The face vector will then be computed and the covariance matrix and the eigenvectors will be calculated from it. The eigenfaces with low eigenvalues will then be omitted. From the remaining eigen faces matrices, the feature weight of each will be calculated and stored as shown in Figure 4.7.

- **Testing Process Flowchart**

In the testing process an image is captured and inserted into the system to find a match from the database. The eigenface, eigenvalues, and feature vectors of that face are then calculated. The Euclidean distance will then be calculated and the minimum Euclidean distance value will be found from it. The minimum Euclidean distance represents the matching value.The minimum Euclidean value together with a threshold will determine if the image is in the database or not as shown in Figure 4.8.

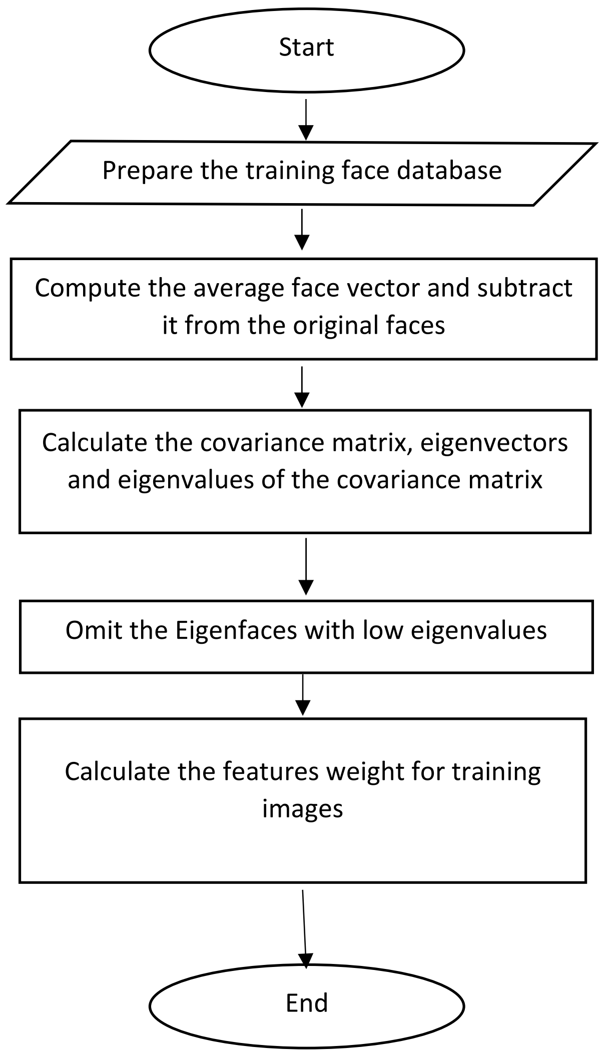


Figure 4.7: PCA Training Process Flowchart

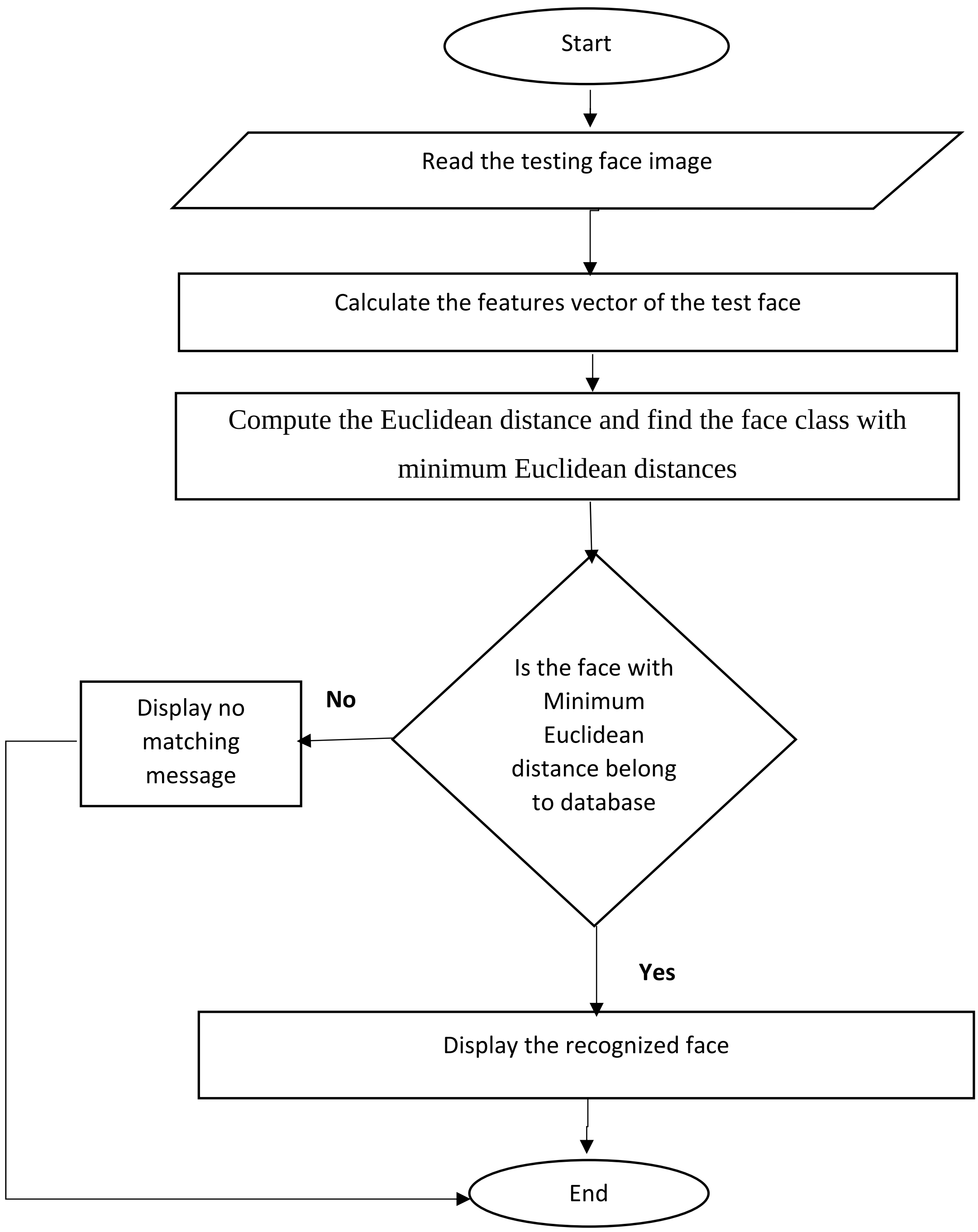


Figure 4.8: PCA Testing Process Flowchart

4.4.2 HMM Flowchart

- **Training Process Flowchart**

The HMM algorithm training process starts by preparing the facial training database. This process filters the facial images in the database and segments it into blocks. The features are then extracted from the segmented blocks and quantized. The seven-state model will then be initialized for the first time, and after each quantitation process the model will be re-estimated. From the model the HMM will then be estimated and stored, as shown in Figure 4.9.

- **Testing flow chart**

In the testing process an image is captured and inserted into the system to find a match from the database. The image will then be filtered and segmented into blocks; the features and quantization of that face are then calculated. The probability of matching will then be calculated, and the max probability will be determined form it. The max probability represents the matching value. The max probability will then be used to determine the closer match in the database, as shown in Figure 4.10.

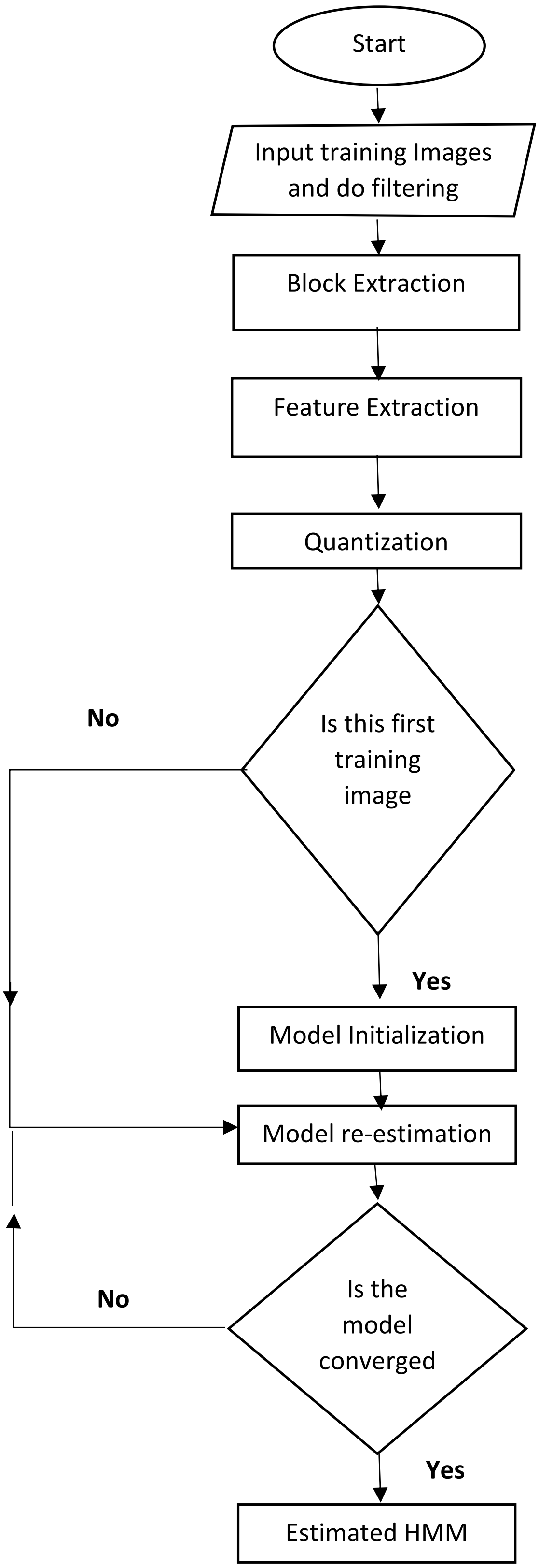


Figure 4.9: HMM Testing process Flowchart

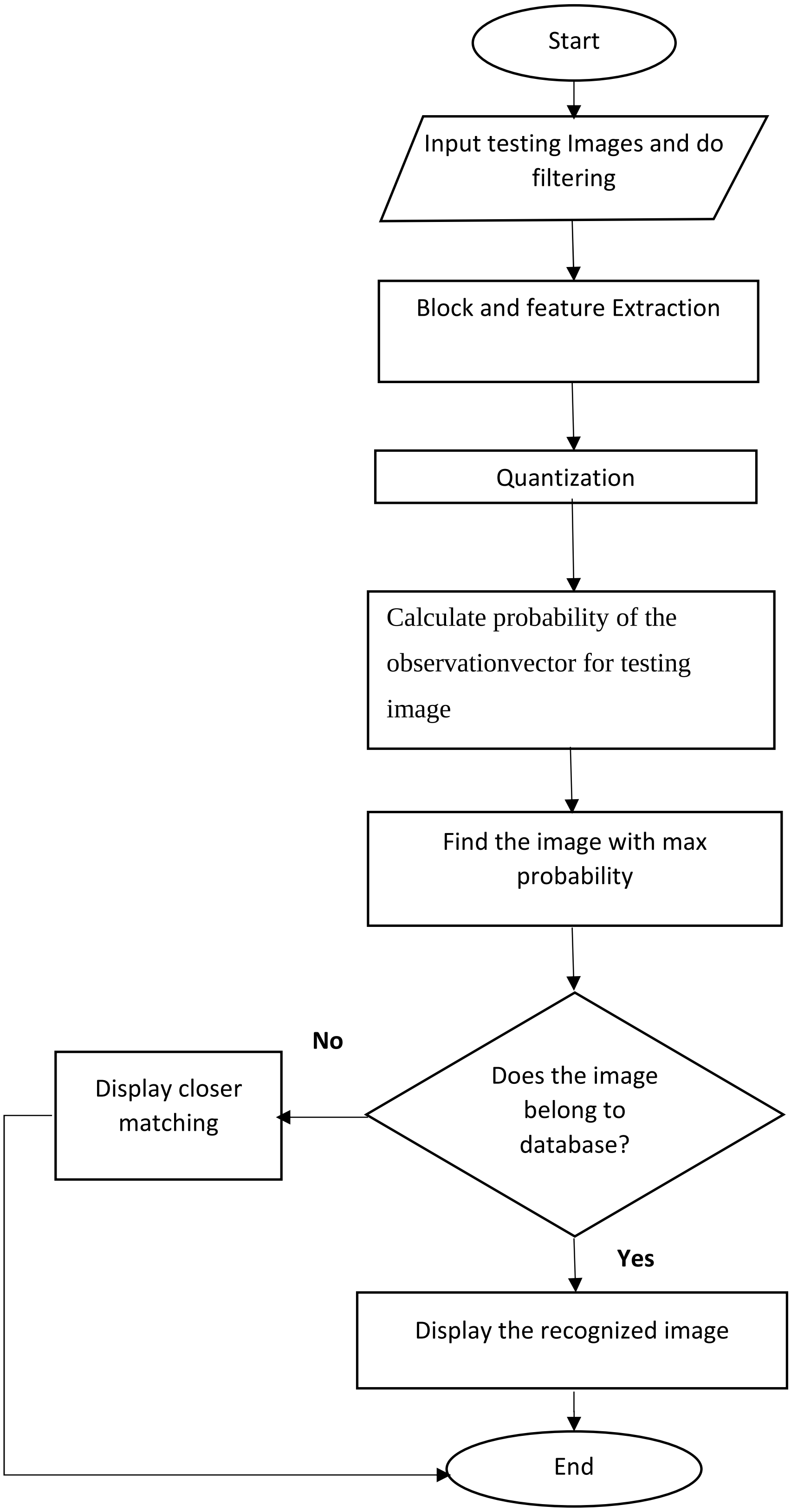


Figure 4.10: HMM Testing process Flowchart

4.5 Graphical User Interface (GUI)

A graphical user interface (GUI) was designed using Matlab to test the performance of the two algorithm; PCA and HMM. The GUI uses two databases the PCA and HMM database. It allows the user to enter the tested facial image either from a pre-specified database or from a webcam. After performing the recognition process the GUI provides the user with the name of the matched person from the database. Figure 4.11 shows the GUI of the system.

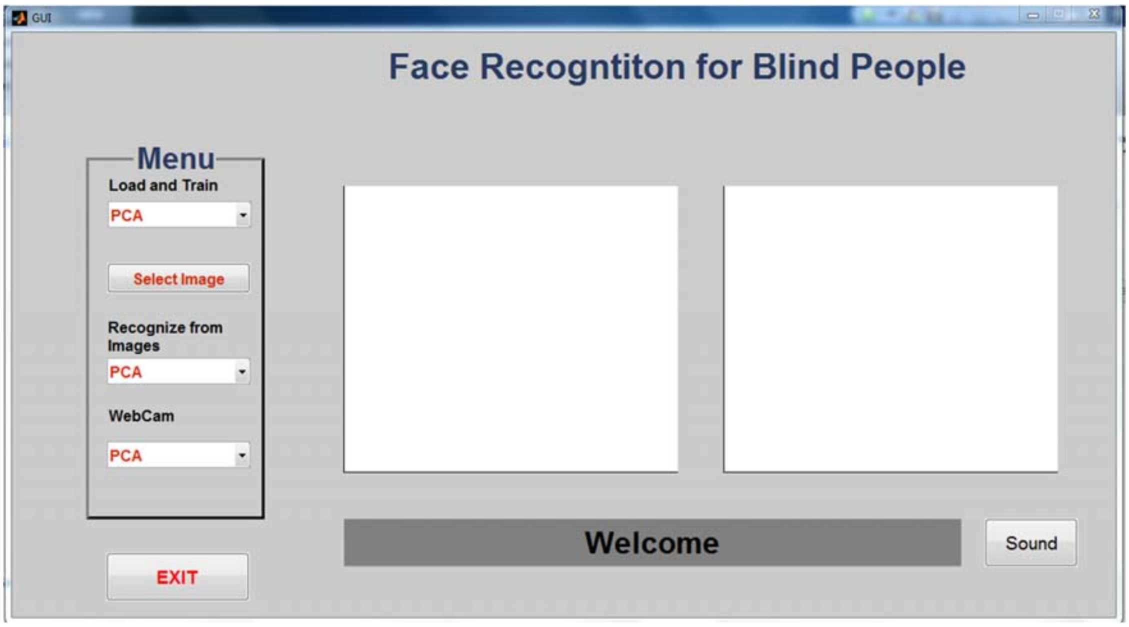


Figure 4.11: System GUI

4.6 Result and Discussions

In this section the performance of both the PCA and HMM algorithm was tested. A facial image is inserted to the system, to find a match from the database. It can be inserted either from a pre-specified database or captured by the webcam.

1. PCA Algorithm

The performance of the system was tested considering the two cases; inserting a facial image from a pre-specified database and capturing the facial image using a webcam.

Case1: A facial image from a pre-specified database

To start the process, the PCA database is chosen. A facial image is then selected from the database as shown in Figure 4.12.

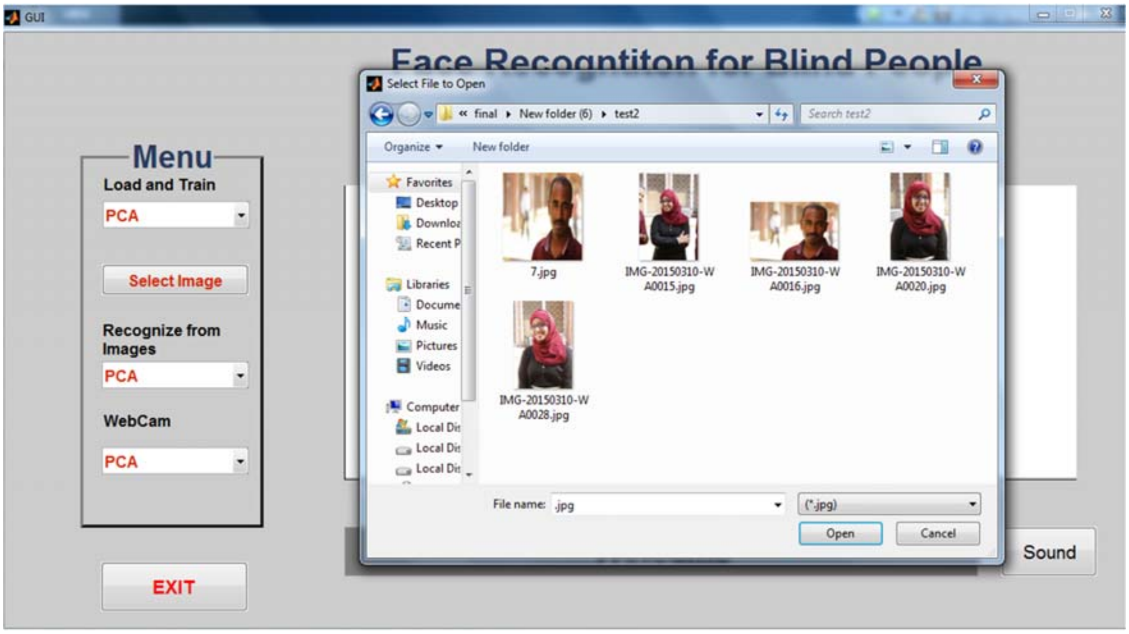


Figure 4.12: Select Image for recognition with PCA

The selected facial image will be displayed on the GUI as shown in Figure 4.13.

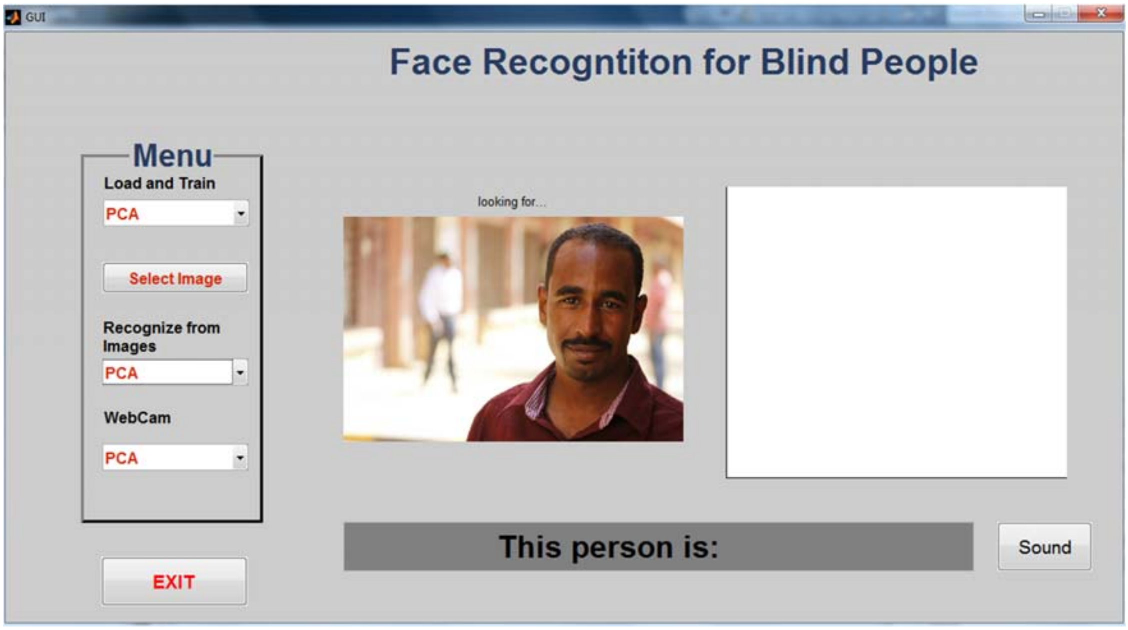


Figure 4.13: Display the selected Image for PCA

Pressing the recognize button will run the PCA algorithm. If a match is found in the database then the person's name will be displayed and announced to the user as shown in Figure 4.14.

Figure 4.14: PCA recognition result

If the facial image entered is not present in the PCA database, a message will be displayed and announced to the user to notify him that this facial image does not exist in the database as shown in Figure 4.15.

Figure 4.15: PCA result for person does not in database

Case 2: A facial image captured by a webcam

In this case the webcam is activated to capture a live facial image to test its presence in the PCA database. Figure 4.16 shows the captured facial image.

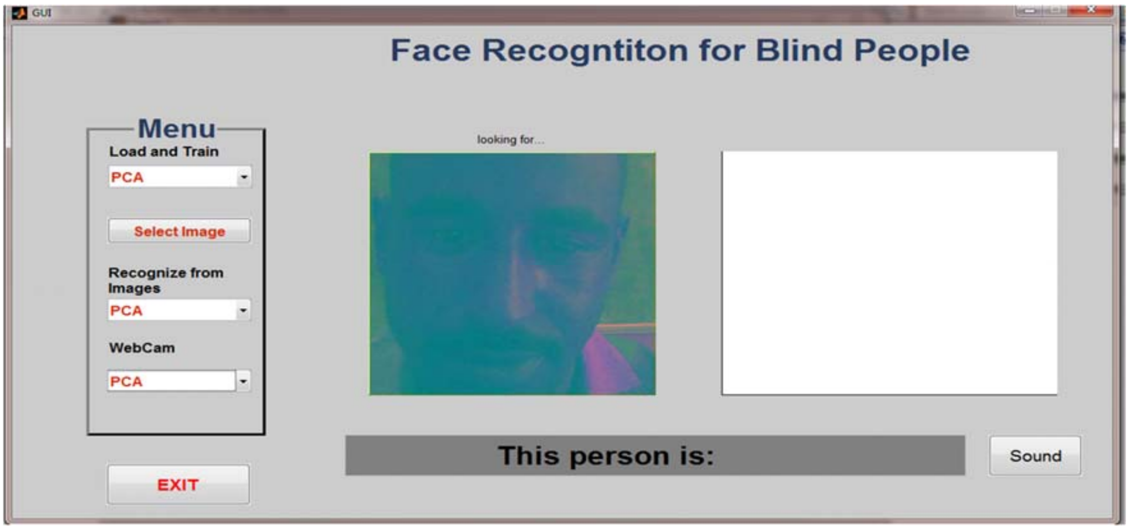


Figure 4.16: Display the captured facial image for PCA

After detecting the face the recognition process will start automatically. If a match is found in the database then the person's name will be displayed and announced to the user as shown in Figure 4.17.

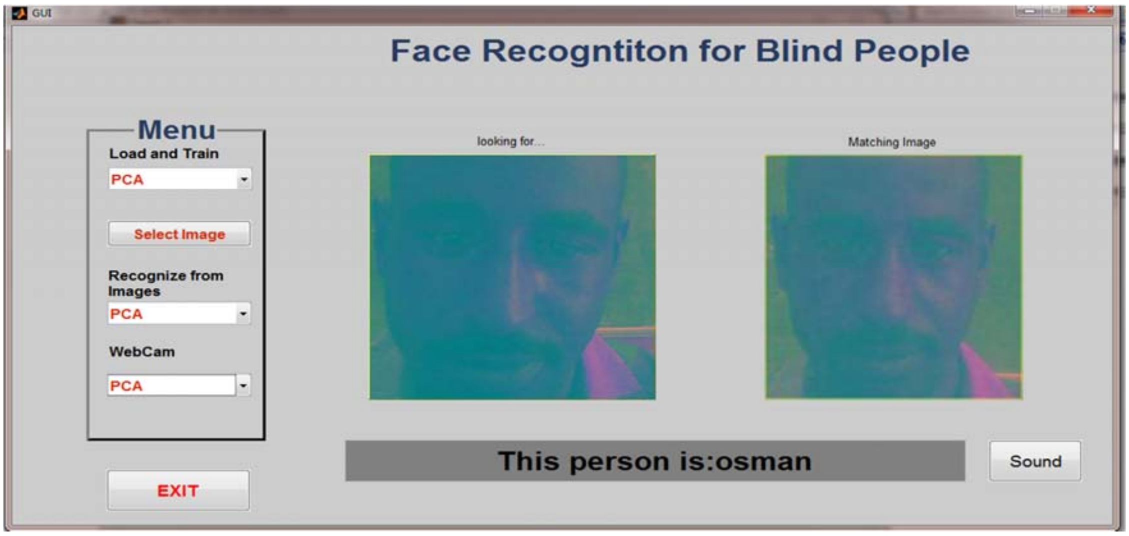


Figure 4.17: PCA live camera recognition result

If the detected face is not present in the PCA database, a message will be displayed and announced to the user to notify him that this facial image does not exist in the database as shown in Figure 4.18.

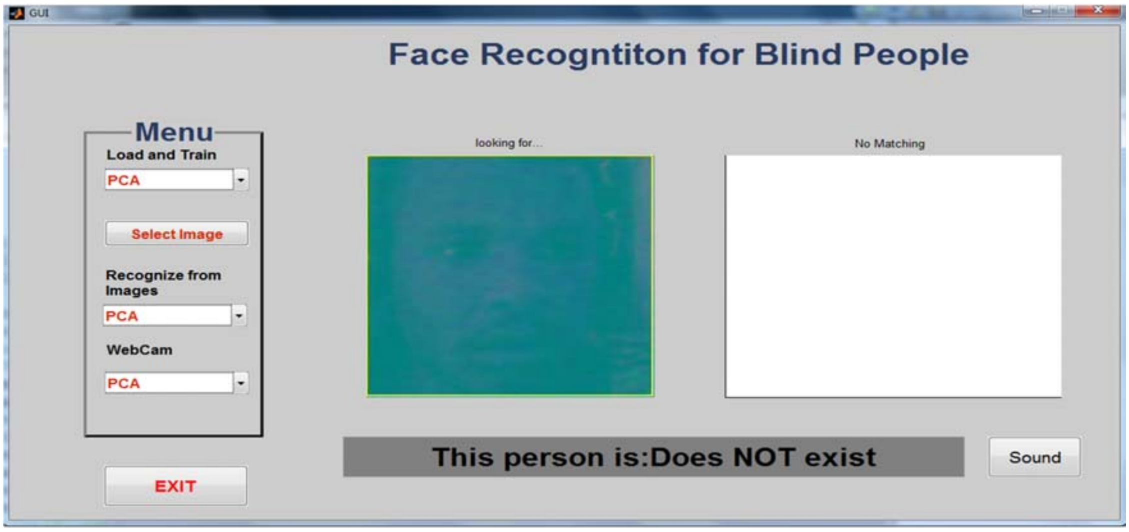


Figure 4.18: PCA camera recognition result for person does not present in PCA database

2. HMM algorithm

The performance of the system was tested considering the two cases; inserting a facial image from a pre-specified database and capturing the facial image using a webcam.

Case1: A facial image from a pre-specified database

To start the process, the HMM database is chosen. A facial image is then selected from the database as shown in Figure 4.19.

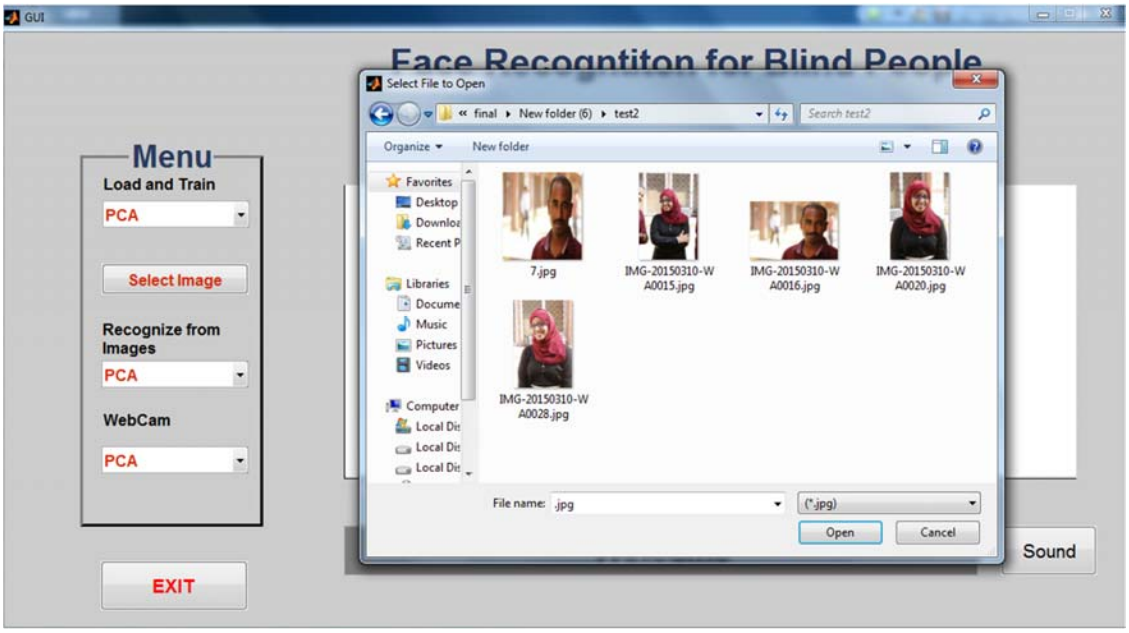


Figure 4.19: select image for HMM recognition

The selected facial image will be displayed on the GUI as shown in Figure 4.20.

Figure 4.20: Display Selected Image for HMM

Pressing the recognize button will run the PCA algorithm. If a match is found in the database then the person's name will be displayed and announced to the user as shown in Figure 4.21.

Figure 4.21: HMM result for person in database

If the facial image entered is not present in the HMM database, the name of the person with the closest matching facial image in the database will be displayed an announced to user as shown in Figure 4.22.

Figure 4.22: HMM result for person does not in database

Case 2: A facial image captured by a webcam

In this case the webcam is activated to capture a live facial image to test its presence in the PCA database. Figure 4.23 shows the captured facial image.

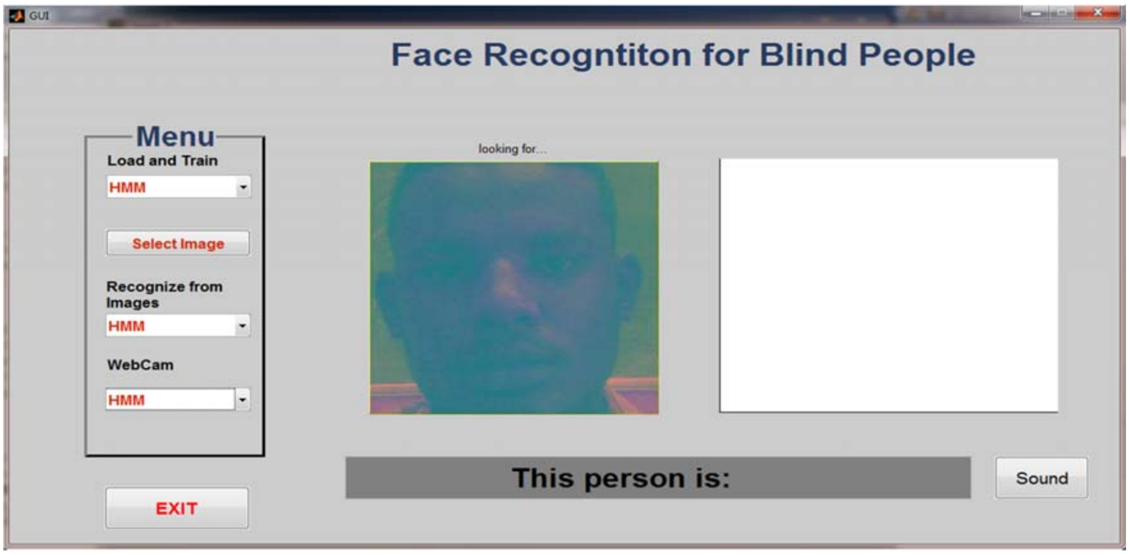


Figure 4.23: Display the captured facial image for HMM

After detecting the face the recognition process will start automatically. If a match is found in the database then the person's name will be displayed and announced to the user as shown in Figure 4.24.

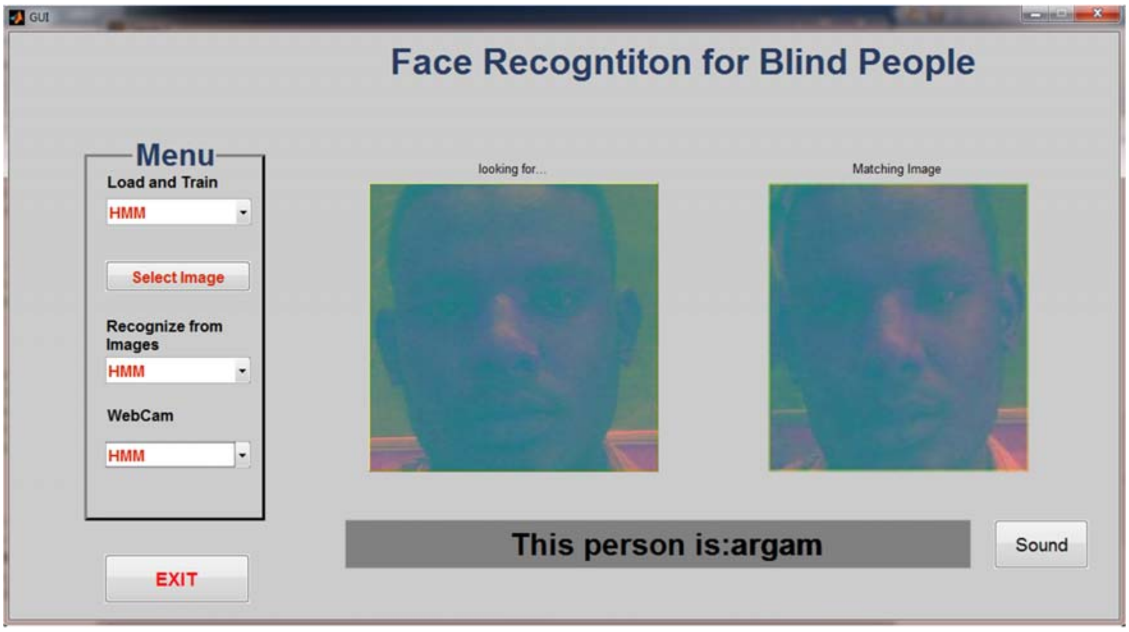


Figure 4.24: PCA live camera recognition result

If the detected face is not present in the HMM database, Then the name of the person with closest matching among the database will be displayed an announced to user as shown in Figure 4.25.

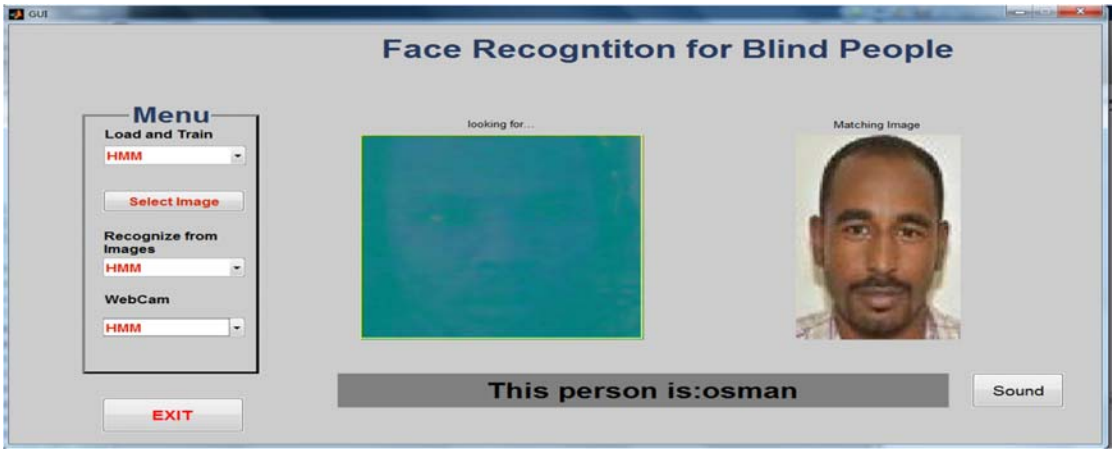


Figure 4.25: HMM camera recognition result

Chapter Five

Conclusion and Recommendations

5.1 Conclusion

Visual impairment afflicts approximately 285 million people worldwide according to recent estimates by the World Health Organization (WHO), and without additional interventions, these numbers are predicted to increase significantly. One of the many challenges faced by this population is their inability to recognize the faces of known individuals when they encounter them in their daily lives. One consequence of this is that whenever a visually impaired individual arrives in a social setting (e.g., in a conference room or at a dinner party), the conversation has to be interrupted to announce which people are already present on the scene which may result in some social awkwardness.

In this thesis Matlab software was used to simulate a system that aids the blind person to recognize his family and friends facial images that are stored in a database. When a person approaches the blind person, his facial image will be captured and compared to the list of faces on the database. If a match is found on the database, the system will announce the name of the person via speakers to the blind person. Two face recognition algorithms were used; Principle Component Analysis (PCA), and Hidden Markov Model (HMM) to compare their performance. The simulation considered the recognition of a static facial image (photo) and a live facial image. The results showed that the PCA is more accurate than the HMM. It has a small recognition time and work properly under different face orientation.

5.2 Recommendations

A lot of work has been done in this thesis but still there is room for more improvement;

- In this thesis, PCA and HMM algorithms were used. It is recommended to use other algorithms such as; support Vector Machine (SVM), Linear Discriminant Analysis (LDA), Dynamic Link Matching (DLM), etc. and compare the results to come up with the optimum recognition rate and time.
- In this project, the camera used is able to recognize one person successfully, it is recommended to carry out more modifications and make it possible to recognize more than one person at the same time.
- This thesis is only based on software simulation, it is recommended to implement a hardware device as a glass to be used as face recognition gadget by visually impaired individuals.

References

Reference

[1] D. Pascolini and S. P. Mariotti, "Global estimates of visual impairment: 2010," British Journal Ophthalmology, 2012.

[2] Rabia Jafri1, Syed Abid Ali2, and Hamid R. Arabnia3 “Face Recognition for the Visually Impaired” ,Department of Information Technology, King Saud University, Riyadh, Saudi Arabia, 2013.

[3] S. Krishna, G. Little, J. Black, and S. Panchanathan, "A wearable face recognition system for individuals with visual impairments," in Proceedings of the 7th international ACM SIGACCESS conference on Computers and accessibility, Baltimore, MD, USA, 2010.

[4] Kandla Arora “Real Time Application of Face Recognition Concept”, International Journal of Soft Computing and Engineering (IJSCE), November 2012.

[5] R. Brunelli and T. Poggio, “Fundamentals” - Computer Vision for Human-Computer Interaction, IEEE Trans, 2014.

[6] Yoon Young Park, YongKeum Choi “A Study on the Design and Implementation of Facial Recognition Application System”, Department of Computer Engineering, Sunmoon University, Korea Department of Dental hygiene, Sunmoon University, Korea, 2014.

[7] Sinha, P., Balas, B., Ostrovsky, Y., & Russell, R. “Face recognition by humans: Nineteen results all computer vision researchers should know about. Proceedings of the IEEE, 2007.

[8] C. Zhang, and Z. Zhang, "Boosted Multi-Task Learning for Face Verification With Applications to Web Image and Video Search", in Proc.

IEEE International Conference on Computer Vision and Pattern Recognition (CVPR2009), Miami, Florida, USA, June 2009.

[9] Rahul Yogi, 2Ashish Goud Purushotham “Review on Side View Face Identification”, International Journal of Emerging Technologies in Computational and Applied Sciences (IJETCAS), 2014

[10] Krishna, S., Little, G., Black, J., & Panchanathan, S. “A wearable face recognition system for individuals with visual impairments”, Paper presented at the ACM SIGACCESS Conference on Assistive Technologies, Baltimore, Maryland, 2012.

[11] Aastha Bhargava and Priya Jain “Biometric; an Effective Shield in Today’s Scenario”, International Conference on Cloud, Big Data and Trust, 2015.

[12] Gursimarpreet Kaur, Dr.Chander Kant Verma “Comparative Analysis of Biometric Modalities”,Department of Computer Science and Application, Kurukshetra University Kurukshetra, India, International Journal of Advanced Research in Computer Science and Software Engineering, 2014.

[13] Sahoo, SoyujKumar; Mahadeva Prasanna, SR, Choubisa, Tarun "Multimodal Biometric Person Authentication: A Review". IETE Technical Review, 2012.

[14] Sushma Jaiswal et al., “Biometric case study”, Journal of Global Research in Computer Science, 2011.

[15] Prashant Kumar Gautam et al, “A Review on Biometric Recognition Systems”, Dronacharya College of Engg. Greater Noida, India, 2007.

[16] A. Abdallah, M. Abou El-Nasr, and A. Lynn Abbott, “A New Face Detection Technique using 2D DCT and Self Organizing Feature Map” in Proc. of World Academy of Science, Engineering and Technology, 2007.

[17] J. Nagi, “Design of an Efficient High-speed Face Recognition System”, Department of Electrical and Electronics Engineering, College of Engineering, Universiti Tenaga Nasional, 2007.

[18] A. S. Tolba, A.H. El-Baz, and A.A. El-Harby “Face Recognition: A Literature Review”, World Academy of Science, Engineering and Technology, 2008.

[19] Dr.S.B.Thorat et al “Facial Recognition Technology: An analysis with scope in India”, International Journal of Computer Science and Information Security, 2010.

[20] Siti Fatimah “face recognition developing using matalb”, Fakulti Kejuruteraan Elektronik dan Kejuruteraan Komputer, Universiti Teknikal Malaysia Melaka, June 2012.

[21] Cahit gurel, “developing of a face recognition system”, altim university, the department of mechatronics engineering, July 2011.

[22] D. Anijiantis, E. Dernzatas and G. Kukkinakis, 1999, “A Neural Network Method for Accurate Face Detection on Arbitrary Images”, Proc. The 6th IEEE International Conference on Electronics, Circuits and Systems, pp. 109 – 112 vol.1, Pafos, Cyprus.

[23] Ion Marqu´es, "Face Recognition Algorithms", Proyecto Fin de Carrera June 16, 2010.

[24] Abhishek Singh, Saurabh Kumar, “Face Recognition Using PCA and Eigen Face Approach”, Department of Computer Science and Engineering National Institute of Technology Rourkela Rourkela, India, 2012.

[25] Laurenz Wiskott and Christoph von der Malsburg, “Recognizing Faces by Dynamic Link Matching”, Ruhr-University at Bochum Institute for Neuro informatics D {44780 Bochum, Germany.

[26] Sanjeev R. Kulkarni and Gilbert Harman, “Statistical Learning Theory: A Tutorial”, 2011

[27] C.-Y. Lin, C.-H. Tsai, C.-P. Lee, and C.-J. Lin. Large-scale logistic regression and linear support vector machines using Spark. IEEE International Conference on Big Data, 2014.

[28] Yoon Young Park, YongKeum Choi “A Study on the Design and Implementation of Facial Recognition Application System”, International Journal of Bio-Science and Bio-Technology, pp.1-10 Korea, 2014.

[29] McLachlan, G. J. (2004). Discriminant Analysis and Statistical Pattern Recognition. Wiley Interscience.

[30] Liton Chandra Paul, Abdulla Al Sumam “Face Recognition Using Principal Component Analysis Method”, Electronics and Telecommunication Engineering, Rajshahi University of Engineering and Technology, November 2012.

[31] Samer Kais JAmeel, “Face Recognition System Using PCA and DCT in HMM”, International Journal of Advanced Research in Computer and Communication Engineering, 2015.

[32] H. Miar-Naimi and P. Davari “A New Fast and Efficient HMM-Based Face Recognition System Using a 7-State HMM Along With SVD Coefficients”, Department of Electrical and Computer Engineering,, Babol University of Technology, Babol, Iran, 2007.

Appendixes

- **Graphical User Interface**

```
function varargout = GUI(varargin)
% GUI MATLAB code for GUI.fig
%      GUI, by itself, creates a new GUI or raises the existing
%      singleton*.
%      H = GUI returns the handle to a new GUI or the handle to
%      the existing singleton*.
%      GUI('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in GUI.M with the given input
arguments.
%      GUI('Property','Value',...) creates a new GUI or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before GUI_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to GUI_OpeningFcn via varargin.
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows
only one
%      instance to run (singleton)".
% See also: GUIDE, GUIDATA, GUIHANDLES
% Edit the above text to modify the response to help GUI
% Last Modified by GUIDE v2.5 12-Jun-2015 12:32:39
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
```

```
                 'gui_Singleton',  gui_Singleton, ...
                 'gui_OpeningFcn', @GUI_OpeningFcn, ...
                 'gui_OutputFcn',  @GUI_OutputFcn, ...
                 'gui_LayoutFcn',  [] , ...
                 'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
% End initialization code - DO NOT EDIT
% --- Executes just before GUI is made visible.
function GUI_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to GUI (see VARARGIN)
% Choose default command line output for GUI
handles.output = hObject;
% Update handles structure
guidata(hObject, handles);
axes(handles.axes1)
cla(handles.axes1,'reset');
set(handles.axes1,'xtick',[],'ytick',[])
cla(handles.axes2,'reset');
```

```
set(handles.axes2,'xtick',[],'ytick',[])
set(handles.edit1, 'String', 'Welcome');

% UIWAIT makes GUI wait for user response (see UIRESUME)
% uiwait(handles.figure1);
% --- Outputs from this function are returned to the command line.
function varargout = GUI_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Get default command line output from handles structure
% --- Executes on selection change in popupmenu1.
function popupmenu1_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu1
contents as cell array
%        contents{get(hObject,'Value')} returns selected item from
popupmenu1
val = get(hObject,'Value');
   case 1  %%%%%%PCA loading
       global TrainDatabasePath m A Eigenfaces T
       pause(0.1);
            choice2 = questdlg('Generating a new database will remove any
previous trained database. Are you sure?', ...
```

```
                    'Warning...',...
                    'Yes', ...
                    'No','No');
        switch choice2
           case 'Yes'
              pause(0.1);
              set(handles.edit1, 'String', 'Loading PCA database...');
              [ T, TrainDatabasePath,m,A] = CreateDatabase();
              case 'No'
        end
        set(handles.edit1, 'String', 'Done...');
      %global TrainDatabasePath m A Eigenfaces T
     % [ T, TrainDatabasePath,m,A,Eigenfaces] = CreateDatabase();
  case 2  %%%%%HMM loading
     global  TrainDatabasePathHMM myDatabaseHMM minmax
        choice2 = questdlg('Generating a new database will remove any
previous trained database. Are you sure?', ...
                    'Warning...',...
                    'Yes', ...
                    'No','No');
        switch choice2
           case 'Yes'
              pause(0.1);
              set(handles.edit1, 'String', 'Loading HMM database...');
              [myDatabaseHMM,minmax,TrainDatabasePathHMM] =
gendata();
           case 'No'
```

```
        end

       %global  TrainDatabasePathHMM myDatabaseHMM minmax
       % [myDatabaseHMM,minmax,TrainDatabasePathHMM] = gendata();
       set(handles.edit1, 'String', 'Done...');
end
% Save the handles structure.
guidata(hObject,handles)
% --- Executes during object creation, after setting all properties.
function popupmenu1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
% --- Executes on selection change in popupmenu2.
function popupmenu2_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu2
contents as cell array
%        contents{get(hObject,'Value')} returns selected item from
popupmenu2
```

```
global TestImage
val = get(hObject,'Value');
switch val
  case 1  %%%%%%PCA Recognition
        global   m  TrainDatabasePath Eigenfaces A
        global SelectedImage b
        [b,str, SelectedImage] =
Recognizefromimage(TestImage,TrainDatabasePath,m,A,Eigenfaces);
        if SelectedImage ~=0
        SelectedImage = imread(SelectedImage);
        axes(handles.axes2)
        %axesimage2=imread(SelectedImage)
        imshow(SelectedImage), title('Matching Image');
        set(handles.edit1, 'String', str);
        else
          set(handles.edit1, 'String', str);
          axes(handles.axes2)
          title('No Matching');
        end
   case 2  %%%%HMM recognition
        global image_index maxlogpseq minmax SelectedImageHMM
        global myDatabaseHMM TrainDatabasePathHMM
        [strHMM, image_index,maxlogpseq, SelectedImageHMM] =
facerec(TestImage,myDatabaseHMM,minmax,TrainDatabasePathHMM);
        if SelectedImageHMM ~=0
        SelectedImageHMM = imread(SelectedImageHMM);
        axes(handles.axes2)
```

```
    imshow(SelectedImageHMM), title('Matching Image');
    set(handles.edit1, 'String', strHMM);
    else
     set(handles.edit1, 'String', strHMM);
     axes(handles.axes2)
     title('No Matching ');
    end
end
% --- Executes during object creation, after setting all properties.
function popupmenu2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
end
% --- Executes on selection change in popupmenu3.
function popupmenu3_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu3
contents as cell array
%        contents{get(hObject,'Value')} returns selected item from
popupmenu3
```

```
val = get(hObject,'Value');
    case 1        %%%%%PCA CAM
        global TestImage TrainDatabasePath m A Eigenfaces
        set(handles.edit1, 'String', 'Camera ....');
        axes(handles.axes1)
        cla(handles.axes1,'reset');
        set(handles.axes1,'xtick',[],'ytick',[])
        axes(handles.axes2)
        cla(handles.axes2,'reset');
        set(handles.axes2,'xtick',[],'ytick',[])
        [TestImage] = PCAcam();
        TestImage = TestImage;
        im = imread(TestImage);
        axes(handles.axes1)
        imshow(im), title('looking for...')
        set(handles.edit1, 'String', 'This person is....');
        axes(handles.axes2)
        cla(handles.axes2,'reset');
        set(handles.axes2,'xtick',[],'ytick',[])
Recognizefromimage(TestImage,TrainDatabasePath,m,A,Eigenfaces);
         if SelectedImage ~=0
         SelectedImage = imread(SelectedImage);
         axes(handles.axes2)
         set(handles.edit1, 'String', str);
         else
           set(handles.edit1, 'String', str);
           axes(handles.axes2)
```

```
    end

  case 2      %%%%%HMM CAM
    global image_index maxlogpseq minmax SelectedImageHMM
    global myDatabaseHMM TrainDatabasePathHMM
    set(handles.edit1, 'String', 'Camera ....');
    axes(handles.axes1)
    cla(handles.axes1,'reset');
    set(handles.axes1,'xtick',[],'ytick',[])
    cla(handles.axes2,'reset');
    set(handles.axes2,'xtick',[],'ytick',[])
    [TestImage] = PCAcam();
    TestImage = TestImage;
    im = imread(TestImage);
    axes(handles.axes1)
    imshow(im), title('looking for...')
    set(handles.edit1, 'String', 'This person is....');
    axes(handles.axes2)
    cla(handles.axes2,'reset');
    set(handles.axes2,'xtick',[],'ytick',[])
    [strHMM, image_index,maxlogpseq, SelectedImageHMM] =
facerec(myDatabaseHMM,minmax,TrainDatabasePathHMM);
     SelectedImageHMM = imread(SelectedImageHMM);
     axes(handles.axes2)
     imshow(SelectedImageHMM),title('Matching Image');
     set(handles.edit1, 'String', strHMM);
guidata(hObject,handles)
```

```
% --- Executes during object creation, after setting all properties.
function popupmenu3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
clc
close all
% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global TestImage TestDatabasePath
[FileName,PathName] = uigetfile({'.jpg'});
                TestDatabasePath = [PathName,FileName];
            end
```

```
TestImage = (TestDatabasePath);
axes(handles.axes1)
imshow(im), title('looking for...')
set(handles.edit1, 'String', 'This person is....');
axes(handles.axes2)
cla(handles.axes2,'reset');
set(handles.axes2,'xtick',[],'ytick',[])
function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1 as a double
% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
% --- Executes on button press in pushbutton4.
```

```
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global b
 [argam, fs]=wavread('C:\Users\7x86\Desktop\New folder
(3)\recording2015_05_10_11_03_53.wav');
fs=fs/2;
if b==1 || b==5
   sound(Noon,fs)
elseif  b==2 || b==4
   sound(osman, fs)
else b==3 || b==6
   sound(argam, fs)
end
```

- **PCA Lading database**

```
function [T, TrainDatabasePath,m,A,Eigenfaces] = CreateDatabase()
TrainDatabasePath = uigetdir('Select training database path' );
TrainFiles = dir(TrainDatabasePath);
Train_Number = 0;
for i = 1:size(TrainFiles,1)
if
not(strcmp(TrainFiles(i).name,'.')|strcmp(TrainFiles(i).name,'..')|strcmp(Trai
nFiles(i).name,'Thumbs.db'))
```

```
Train_Number = Train_Number + 1;
end
end
T = [];
for i = 1 : Train_Number
str = strcat('\',str,'.jpg');
str = strcat(TrainDatabasePath,str);
img = rgb2gray(img);
   [a b] = size(img);
temp = reshape(img',a*b,1);
   T = [T temp];
end
[m, A, Eigenfaces] = EigenfaceCore(T);
save DATABASE-PCA m A Eigenfaces T
function [m, A, Eigenfaces] = EigenfaceCore(T)
m = mean(T,2);
Train_Number = size(T,2);
A = [];
for i = 1 : Train_Number
temp = double(T(:,i)) - m;
   A = [A temp];
end
L = A'*A;
[V D] = eig(L);
L_eig_vec = [];
for i = 1 : size(V,2)
if( D(i,i)>1 )
```

```
L_eig_vec = [L_eig_vec,V(:,i)];
end
end
```

- **PCA Recognition**

```
function  [b,str, SelectedImage] =
Recognizefromimage(TestImage,TrainDatabasePath,m,A,Eigenfaces)
 (3)\recording2015_05_10_11_03_46.wav');
[argam, fs]=wavread('C:\Users\7x86\Desktop\New folder
(3)\recording2015_05_10_11_03_53.wav');
global Recognized_index OutputName
[OutputName, Recognized_index] = Recognition(TestImage, m, A,
Eigenfaces);
 if Recognized_index~=0
SelectedImage = strcat(TrainDatabasePath,'\',OutputName)
b = Recognized_index;
str = strcat('This person is: ',s);
fs=fs/2;
if b==1 || b==5
   sound(Noon,fs)
elseif  b==2 || b==4
   sound(osman, fs)
else b==3 || b==6
```

```
   sound(argam, fs)
end
else
SelectedImage=0;
b=0;
 end
end

function [OutputName, Recognized_index] = Recognition(TestImage, m, A,
Eigenfaces)
ProjectedImages = [];
Train_Number = size(Eigenfaces,2);
for i = 1 : Train_Number
temp = Eigenfaces'*A(:,i);
ProjectedImages = [ProjectedImages temp];
end
InputImage = imresize(InputImage,[200 180]);
temp = InputImage(:,:,1);
[a b] = size(temp);
InImage = reshape(temp',a*b,1);
Difference = double(InImage)-m;
ProjectedTestImage = Eigenfaces'*Difference;
Euc_dist = [];
for i = 1 : Train_Number
   temp = ( norm( ProjectedTestImage - q ) )^2;
Euc_dist = [Euc_dist temp];
end
```

```
[Euc_dist_min ,Recognized_index] = min(Euc_dist)
if(Euc_dist_min<6.7290e+14)
if(Euc_dist_min<6.6181e+14)
   OutputName = strcat(int2str(Recognized_index),'.jpg')
   Recognized_index=0;
   OutputName=0;
end
else
   Recognized_index=0;
   OutputName=0;
end
end
```

- **HMM loading database**

```
function [myDatabaseHMM,minmax,TrainDatabasePathHMM] = gendata()
eps=.000001;
TrainDatabasePathHMM = uigetdir('Select training database path' );
TrainFiles = dir(TrainDatabasePathHMM);
myDatabaseHMM = cell(0,0);
person_index = 0;
max_coeffs = [-Inf -Inf -Inf];
min_coeffs = [ Inf  0  0];
for i = 1:size(TrainFiles,1)
not(strcmp(TrainFiles(i).name,'.')|strcmp(TrainFiles(i).name,'..')|strcmp(Trai
nFiles(i).name,'Thumbs.db'))
```

```
person_index = person_index + 1;
end
end
for person=1:person_index
    blk_cell = cell(0,0);
        str = int2str(person);
        str = strcat('\',str,'.jpg');
        str = strcat(TrainDatabasePathHMM,str);
        I = imread(str);
        I = imresize(I,[56 46]);
        blk_index = 0;
        for blk_begin=1:52
            blk_index=blk_index+1;
            blk = I(blk_begin:blk_begin+4,:);
            [U,S,V] = svd(double(blk));
            blk_coeffs = [U(1,1) S(1,1) S(2,2)];
            max_coeffs = max([max_coeffs;blk_coeffs]);
            blk_cell{blk_index,person} = blk_coeffs;
        end
        myDatabaseHMM{1,person} = blk_cell;
end
delta = (max_coeffs-min_coeffs)./([18 10 7]-eps);
minmax = [min_coeffs;max_coeffs;delta];
for person=1:person_index
        for block_index=1:52
            blk_coeffs = myDatabaseHMM{1,person}{block_index,person};
            qt = floor((blk_coeffs-min_coeffs)./delta_coeffs);
```

```
         myDatabaseHMM{2,person}{block_index,person} = qt;
         label = qt(1)*10*7+qt(2)*7+qt(3)+1;
         myDatabaseHMM{3,person}{block_index,person} = label;
      end
      myDatabaseHMM{4,person} =
cell2mat(myDatabaseHMM{3,person});
end
TRGUESS = ones(7,7) * eps;
TRGUESS(7,7) = 1;
EMITGUESS = (1/1260)*ones(7,1260);
fprintf('\nTraining ...\n');
for person=1:person_index
   seqmat = cell2mat(myDatabaseHMM{3,person})';
[ESTTR,ESTEMIT]=hmmtrain(seqmat,TRGUESS,EMITGUESS,'Tolerance
',.01,'Maxiterations',10,'Algorithm', 'BaumWelch');
   ESTTR = max(ESTTR,eps);
   ESTEMIT = max(ESTEMIT,eps);
   myDatabaseHMM{5,person}{1,1} = ESTTR;
   myDatabaseHMM{5,person}{1,2} = ESTEMIT;
end
save DATABASE myDatabaseHMM minmax person_index
TrainDatabasePathHMM
```

- **HMM recognition**

```
function  [strHMM, image_index,maxlogpseq,SelectedImageHMM] =
facerec(TestImage,myDatabaseHMM,minmax,TrainDatabasePathHMM)
 [argam, fs]=wavread('C:\Users\7x86\Desktop\New folder
(3)\recording2015_05_10_11_03_53.wav');
I = imread(TestImage);
try
   I = rgb2gray(I);
end
I = imresize(I,[56 46]);
max_coeffs = minmax(2,:);
delta_coeffs = minmax(3,:);
seq = zeros(1,52);
for blk_begin=1:52
   blk = I(blk_begin:blk_begin+4,:);
   [U,S,V] = svd(double(blk));
   blk_coeffs = [U(1,1) S(1,1) S(2,2)];
   blk_coeffs = max([min_coeffs;blk_coeffs]);
   qt = floor((blk_coeffs-min_coeffs)./delta_coeffs);
   label = qt(1)*7*10+qt(2)*7+qt(3)+1;
   seq(1,blk_begin) = label;
end
number_of_persons_in_database = size(myDatabaseHMM,2);
results = zeros(1,number_of_persons_in_database);
for i=1:number_of_persons_in_database
```

```
   TRANS = myDatabaseHMM{5,i}{1,1};
   EMIS = myDatabaseHMM{5,i}{1,2};
   [ignore,logpseq] = hmmdecode(seq,TRANS,EMIS);
   P=exp(logpseq);
   results(1,i) = P;
end
[maxlogpseq,image_index] = max(results)
if(maxlogpseq>1.0002e-312)
if(maxlogpseq>1.0000e-312)
   OutputName = strcat(int2str(image_index),'.jpg');
SelectedImageHMM = strcat(TrainDatabasePathHMM,'\',OutputName);
strHMM = strcat('This person is: ',S);
fs=fs/2;
if X==1

   sound(Noon,fs)
elseif  X==2 || X==4
   sound(osman, fs)
else X==3 || X==5
   sound(argam, fs)
end


else
   strHMM = strcat('This person Does NOT exist ');
   image_index=0;
   SelectedImageHMM =0
   X=0;
```

```
end
else
       strHMM = strcat('This person Does NOT exist ');
   image_index=0;
    SelectedImageHMM =0;
    X=0;
end
end
```

- **Camera**

```
function [TestImage] = PCAcam()
faceDetector = vision.CascadeObjectDetector();

videoFileReader = imaq.VideoDevice('winvideo', 1,
'YUY2_320x240','ROI',[1 1 320 240]);
bbox            = step(faceDetector, videoFrame);
while(size(bbox,1)<1)
   videoFrame= step(videoFileReader);
    bbox= step(faceDetector, videoFrame);
end
 for i=1:size(bbox,1)
videoFrame = insertShape(videoFrame, 'Rectangle', bbox(i,:));
faceImage = imcrop(videoFrame,bbox(1,:));
end
```

```
figure(1); imshow(videoFrame); title('Detected face');
TestImage=('C:\Users\7x86\Desktop\video\matlabFaceRecognitionRealTim
e-master\Test.jpg');



% Draw the returned bounding box around the detected face.
  %  for i=1:size(bbox,1)
   %     videoFrame = insertShape(videoFrame, 'Rectangle', bbox(i,:));
   % facelmage = imcrop(videoFrame,bbox(1,:));
     %   figure(i); imshow(videoFrame); title('Detected face');
  % figure(2); imshow(facelmage); title('Detected face');
    % Display the annotated video frame using the video player object
   % step(videoPlayer, videoFrame);
    % Detect feature points in the face region.
    %points = detectMinEigenFeatures(rgb2gray(videoFrame), 'ROI', bbox);
%end
%Clean up
%release(videoFileReader);
%release(videoPlayer);
%release(pointTracker);

end
```