Computer Engineering Department
Faculty of Engineering
Deanery of Higher Studies
Islamic University – Gaza
Palestine

Robust Face Recognition

Shady S. Al-atrash

Supervisor

Prof. Ibrahim S. I. Abuhaiba

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Computer Engineering

1432H (2011)

ACKNOWLEDGMENT

All praise and glory are due to Allah the almighty who provided me with the much needed strength and stamina to successfully accomplish this work. The completion of this thesis also owes a great deal to the help and unrelenting support of the people around me. I am especially indebted to my advisor, Prof. Ibrahim Abuhaiba, for his guidance, support and patience with me throughout the course of my research. He taught me the essence and principles of research and guided me through until the completion of this thesis. I could not have asked for a better advisor.

I owe my largest debt to my family, and I wish to express my heartfelt gratitude to all of them for their encouragement, constant prayers, and continued support. My dear parents who have given me all their love and support over the years; I thank them for their unwavering commitment through good times and hard times.

My profound gratitude goes to my wife for her support, endurance and patience. I dedicate this thesis to her.

Finally, I thank all of those whose names are not mentioned here but have helped me in any way to accomplish this work.

TABLE OF CONTENTS

LIST OF ACRONYMS

FD	Face Detection
FR	Face Recognition
RGB	Red, Green, Blue Color Space
ATM	Automated Teller Machine
NNs	Neural Networks
ANN	Artificial Neural Network
HCI	Human Computer Interaction
PDBNN	Probabilistic Decision-Based Neural Network
DBNN	Decision Based Neural Network
SVM	Support Vector Machine
HMM	Hidden Markov Model
PCA	Principal Component Analysis
LDA	Linear Discriminant Analysis
FLD	Fisher Linear Discriminant
AI	Artificial Intelligence
DFT	Discrete Fourier Transform
DCT	Discrete Cosine Transform
FRej.	False Rejection
Facc.	False Acceptance
Crej.	Correct Rejection
Crec.	Correct Recognition

LIST OF FIGURES

LIST OF TABELS

تعرف قوي على الوجه

شادي سهيل الأطرش

الملخص

الهدف من هذه الأطروحة هو اقتراح خوارزميات جديدة في اكتشاف الوجه والتي يكون لها القدرة على الكشف عن الوجه في مختلف الاتجاهات وتحت ظروف مختلفة. هذا الهدف تم الحصول عليه على مراحل مختلفة؛ وباستخدام خوارزميات مقترحة متعددة. أولا: تم اقتراح خوارزمية قوية لتفريغ قطاع الجلد من الصورة . ثانيا: تم تطبيق خطوات مختلفة لتصفية هذه الصورة المفرغة, وذلك للحصول على منطقة الوجه المرشحة فقط. بعد ذلك: تم استخدام النهج القائم على ميزة الكشف عن الملامح (Feature-Based)، وذلك للكشف عن الملامح من الوجه المرشح. هذا النهج يمكن أن يعمل في الوقت الحقيقي مع قدر ضئيل من التدريب, وذلك على عكس المناهج الأخرى التي تعتمد على الصورة ككل والمسمى (Image-Based). وفي النهاية: تم تطبيق بعض القواعد من أجل الحكم ما إذا كان هذا الوجه المرشح هو وجه جانبي أو لا؛ سواء كان هذا الوجه له وضع جانبي إلى اليمين أو اليسار.

ولتعزيز هذا الجهد؛ تم استخدام نهج قالب المطابقة (Template Matching), وذلك للكشف عن الوجه ذو الاتجاه الأمامي. وبعد الكشف عن الوجه بشكل صحيح، تم تطبيق بعض الخطوات لمعالجة صورة الوجه المكتشف؛ وذلك للحصول على صورة الوجه بشكل معتدل ومقلص وبصورة لها تدرج رمادي, مما يساعد في الحد من تعقيد العمليات الحسابية عند تطبيق خوارزمية التعرف على الأوجه. تم تطبيق تقنية (Fisher) على صورة الوجه المعالجة من الخطوات السابقة؛ وذلك للتعرف عليه الوجه تحت ظروف مختلفة.

هذا العمل له العديد من المزايا على أنهج أخرى، وذلك ينبع من قدرة الخوارزمية المقترحة من كشف الأوجه التي تكون في اتجاهات مختلفة (يسار، يمين، أمام)، وتحت ظروف إضاءة مختلفة، وبوجود أعراق متعددة. ولقد تم الحصول على ذلك عبر دمج بعض الخوارزميات المعروفة مثل (Template Matching) مع خوارزمية جديدة تم اقتراحها؛ وذلك لتقوية عملية اكتشاف الأوجه، والحصول على خوارزمية قوية, وتعطي نتائج جيدة.

وقد أظهرت النتائج أن الأسلوب المقترح هو قوي في ظل طائفة واسعة من ظروف الإضاءة واختلاف وضعية الوجه وتعدد الأعراق. هذه النتائج أخذت من ثلاثة قواعد بيانات أوجه مختلفة وهي GTAV وFEI وChampion. هذه القواعد تحتوي على أشخاص من أعراق مختلفة, وتحت ظروف إضاءة مختلفة، وبوضعيات أوجه مختلفة. هذه الخوارزمية المقترحة نفذت باستخدام برنامج Matlab نسخة 7.6, وقيمت باستخدام لابتوب من نوع Dell-Inspiron، يعمل باستخدام معالج 2- جيجا هرتز، ورام 3-جيجا بايت من نوع DDR2 . من هذه النتائج تبين أن معدل الإكتشاف الصحيح لقاعدة بيانات الأوجه GTAV هو 95.04%. بينما بالنسبة لقاعدة بيانات FEI فكان المعدل هو 94.5%. أما أخيرا, بالنسبة لقاعدة بيانات Champion فإن معدل الاكتشاف الصحيح هو 93.3% .

الخوارزميات المقترحة تعطي نتائج جيدة مقارنة مع خوارزميات باحثين آخرين. وبمقارنة هذه الخوارزمية المقترحة مع خوارزميات اكتشاف أوجه أخرى فإنها تعطي تحسين للأداء بمعدل 2.87%. بينما عند مقارنة خوارزمية التعرف على الأوجه التي استخدمت؛ وذلك بعد تطبيق بعض خطوات المعالجة المقترحة، بخوارزميات أخرى يعطي تحسن في الأداء بمعدل 0.2 % .

Robust Face Recognition

Shady S. Al-atrash

ABSTRACT

The objective of this thesis is to propose new algorithms in face detection which have the capabilities of detecting the face with different poses and under different conditions. This objective is obtained in different stages and using different proposed algorithms. Firstly, a robust segmentation algorithm is proposed to segment the skin region from the image. Secondly, different filtering steps are applied to this segmented image to obtain the face candidate region only. After that, Feature-Based approach is used to detect the features from this candidate face which can work in real-time with minimal training in contrast to other approaches such as image-based approach. Finally, some rules is applied in order to judge if this candidate is profile face or not either the profile face is right or left.

To strengthen this work, Template Matching approach is used to detect a frontal face. After detecting the face correctly, some pre-processing steps is applied to the detected face to obtain a down scaled normalized gray face image that help in reduce the computation complexity when applying the face recognition algorithm. The Fisher faces technique is applied to the processed face in order to recognize it in different conditions.

This work has advantages over other approaches flow out from the ability of the proposed algorithm on detecting faces with different poses (Left, Right and Frontal) in varying illumination conditions and different races. This is done by merging some already defined algorithms such as Template Matching with new proposed algorithm to strength the detection process and obtaining robust algorithm with good results.

Experimental results show that the proposed method is robust under a wide range of lighting conditions, different poses, and different races. These results are taken from three different face databases which are GTAV, FEI and Champions. These databases contain peoples from different races under different illumination conditions. The proposed method is implemented using Matlab version 7.6 software and they evaluated using Delll-inspiron laptop run with a 2-GHz CPU and has a 3-GB DDR2 RAM. The correct detection rate for the GTAV face database is 95.04%. While, for FEI face database it has 94.5% rate. Finally, For Champions face database the detection rate is 93.3%.

The proposed face detection algorithm give good results compared to other researchers algorithms. The comparison of this proposed algorithm with other face detection algorithms gives an improved performance rate estimated by 2.87%. While, the comparison of our used face recognition algorithm after applying some proposed pre-processing steps with other algorithms gives an improved performance rate estimated by 0.2% .

KeyWords: Face Detection, Face Recognition, Fisher Face, Skin Segmentation, Profile Face, Frontal Face, Complex Background, Different Lightening Conditions, Face Pose.

Chapter 1

INTRODUCTION

1.1 Background

The face plays a main role in carrying identity of persons. One of things that really admire the viewer is the human ability to recognize faces. Humans can recognize thousands of faces and identify familiar faces despite large changes in the visual stimulus due to viewing conditions, expression, aging, sex, and distractions such as glasses, or changes in hair style [1]. When we pays attention to human ability in Face Recognition (FR), we don't know how faces are decoded by the human brain.

FR has been studied for over two decades [2, 3, 4], in order to make a noticeable advance in this admire field and it is still an active subject due to extensive practical applications. Many recent events, such as terrorist attacks, exposed serious weakness in most sophisticated security systems. Various government agencies are now more motivated to improve security data systems based on body or behavioral characteristics, often called biometrics. Biometrics is a very attractive technology, because it can be integrated into any application requiring security or access control, effectively eliminating risks associated with less advanced technologies that are based on what a person have or know rather than whom a person really is.

Perhaps the most common biometrics are fingerprints and iris, but many other human characteristics have been studied in last years such as finger/palm geometry, voice, signature, face. Figure 1.1 shows the spreading of the most popular biometrics in the last years from a commercial point of view.

However, biometrics have drawbacks. Iris recognition is extremely accurate, but expensive to implement and not very accepted by people. Fingerprints are reliable and non-intrusive, but not suitable for non-collaborative individuals. On the contrary, FR seems to be a good compromise between reliability and social acceptance and balances security and privacy well.

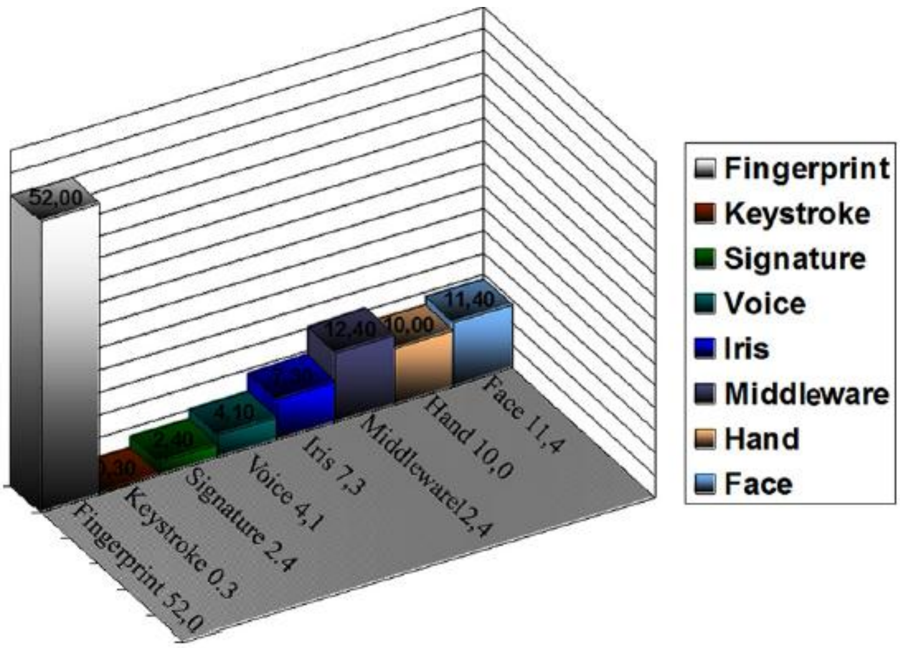


Figure 1.1: The spreading of most popular biometrics [23].

FR is done naturally by humans. However, developing a computer algorithm to do the same thing is difficult. Assume for the moment we start with images, and we want to distinguish between images of different people. Many FR have been developed to construct a set of "images" that provides the best approximation of the overall image data set. The training set is then projected onto this subspace. To query a new image, we simply project the image onto this subspace and seek a training image whose projection is closest to it.

Face Detection (FD) is one of the main biometric features that many researches concentrate on developing algorithms to apply it in different systems. FD is one of the fundamental techniques that enable a Human Computer Interaction (HCI) [5]. So, FD is the main step stone to all facial analysis algorithms, including face alignment, face verification/authentication, gender/age recognition, and especially in FR which is the area of our interest. The following sections will hold the light on the approaches that can be used in FD and FR fields.

1.2 Face Detection Obstacles

The goal of FD is to determine whether or not there are any faces in the image and, if present, locate the image face. While this appears as a trivial task for human beings, it is a very challenging task for computers, and has been one of the top studied research topics in the past few decades [6,7]. Early efforts in FD have presented as early as the beginning of the 1970s, where simple heuristic techniques [8] were used.

These techniques are largely rigid since they assume ideal conditions such as plain background and frontal face. Some of the factors that make FD such a difficult task are:

- ***Face orientation*:** A face can appear in many different poses. For instance the face can appear in a frontal or a profile (i.e. sideways) position. Furthermore a face can be rotated in plane (e.g. it appears under an angle of 45'), so the near profile face position can appear in the image. Therefore, a face appears in many different shapes in an image.
- ***Face size*:** The size of the human face can vary a lot. Not only do different persons have different sized faces, also faces closer to the camera appear larger than faces that are far away from the camera.
- ***Different facial expression*:** The appearance of a person who is laughing is totally different than the appearance of a person who is angry. Therefore facial expressions directly affect the appearance of the face in the image.
- ***Different facial feature*:** Some people wear glasses, some have a beard or a mustache, others have a scar. These types of features are called facial features. There are countless examples of facial features and they all vary in shape, size and color.
- ***Occlusion*:** Faces in images may be partially occluded. For instance a person standing in front of another or an object that is placed in front of the face. Therefore only part of the facial image is present in the image.
- ***Lighting condition*:** Faces appear totally different when different lighting conditions are used. For instance when side lighting is used, a part of the face is very bright while the other part is very dark.

1.3 Face Detection Approaches

FD techniques require a priori information of the face, so, they can be effectively organized into two broad categories which can be used to determine the face knowledge from one person to other. The first approach is **feature-based**, while the second approach is **image-based** approach. The following sections discuss each category alone.

1.3.1 FD Feature-Based Approach

This approach apply many techniques in FD that make explicit use of face knowledge and follow the classical detection methodology in which low level features are obtained prior to knowledge-based analysis. Typically, in these techniques FD tasks are accomplished by manipulating distance, angles, and area measurements of the visual features derived from the image. Since features are the main components, these techniques are termed the *feature-based* approach [9]. This approach has the majority of interest in FD research starting as early as the 1970s, and we base on it in my research.

It is easy to come up with simple rules to describe the features of a face and their relationships. For example, a face usually appears with two eyes that are symmetric to each other, a nose and mouth. The relationships between these features can be represented by their relative distances and positions. Faces are then detected by applying these codes rules to find features. Face candidates are detected after applying all the rules.

One problem with this approach is that, it is difficult to translate our knowledge about faces into rules effectively. If the rules are strict it may fail to detect faces that don't pass all the rules. If the rules are too general, it may give many false positives [7].

In any FD problem use feature-based approach for locating face firstly, it will start with *low level feature analysis* in which segmentation of visual features using pixel properties such as gray-scale and color will be done. After that, Features generated from this are likely to be ambiguous which can be solved by *higher level feature analysis* such as Template Matching approach that use knowledge of face geometry to characterize and subsequently verify various features from their ambiguous state [10,11].

1) Low Level Analysis

The following section shows the features that can be detected from low level feature analysis step:

- Color Information

In spite of obtaining important feature information from gray information, color is a more powerful means of describing object appearance. Detection of skin color in color images is a very popular and useful technique for FD process. Many techniques [12, 13] have reported for locating skin color regions in the input image. There are many color spaces used to classify the skin color. We go through the most popular color spaces that can be used in FD which are RGB, YCbCr.

- RGB Color Space

One of the most widely used color spaces (models) is RGB, which consists of three additive primaries colors: red, green and blue. Spectral components of these colors combine additively to produce a resultant color [14]. The RGB model is represented by a 3-dimensional cube with red green and blue at the corners on each axis as shown in Figure 1.2. Black is at the origin. White is at the opposite end of the cube. The gray scale follows the line from black to white [14]. In a 24-bit color graphics system with 8 bits per color channel, red is (255, 0, 0) and blue is (0, 0,255). On the color cube, it is respectively (1, 0, 0) and (0, 0, 1).

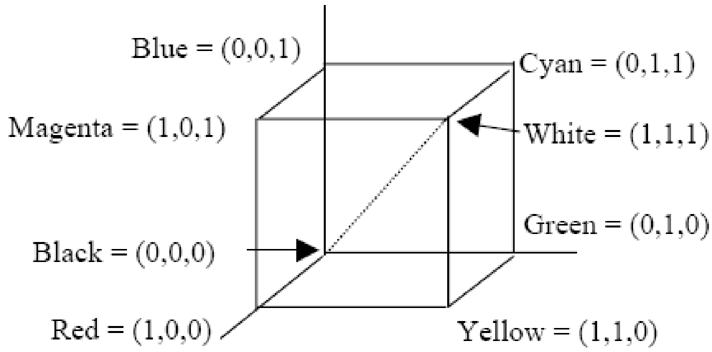


Figure 1.2: RGB Color Cube [14].

Though RGB model is used in computer graphics, it is not ideal for all applications. Because the red, green and blue color components are correlated, it is difficult to use RGB color space in some image processing algorithms for skin color classifying. However, RGB is one of the most common color spaces used in this field; but this color space components (Red, Green, Blue) doesn't represent only the color but also luminance that may vary across a person's face because of different lighting conditions so, this model is not a reliable for separating skin pixels from non-skin pixels.

- RGB Color Value Normalization

Since the main variation in skin appearance is largely due to luminance change (brightness) [15], *normalized RGB* colors are generally preferred, so that the effect of luminance can be filtered out. The normalized colors can be derived from the original RGB components as follows:

$$r = \frac{R}{R+G+B} \qquad \text{.................. (1.1)}$$

$$g = \frac{G}{R+G+B} \qquad \text{.................. (1.2)}$$

$$b = \frac{B}{R+G+B} \qquad \text{.................. (1.3)}$$

Where, R, G and B are the three components of the RGB color space. From above equations, it can be seen that (r + g + b= 1). The normalized colors can be effectively represented using only r and g values as b can be obtained by noting (b=1- r- g) so, the third component (b) does not hold any significant information and can be omitted, reducing the space dimensionality.

- YCbCr Color Space

YCbCr color space has been defined in response to increasing demands for digital algorithms in handling video information, and has become a widely used model in a digital video. YCbCr is from the family of television transmission color spaces that includes others formats such as YUV and YIQ so; it is a digital color system while YUV and YIQ are analog spaces.

YCrCb is an encoded nonlinear RGB signal, commonly used by European television studios and for image compression work. YCbCr separates RGB (Red-Green-Blue) color space into luminance and chrominance information which are useful in detecting face skin color applications [15]. Color is represented by luma (Y) (i.e. luminance), computed from nonlinear RGB , constructed as a weighted sum of the RGB values, and two color difference values chrominance red (Cr) and chrominance blue (Cb) that are formed by subtracting luma from RGB red and blue components.

$$Y = 0.299R + 0.587G + 0.114B \text{ .... (1.4)}$$

$$Cr = R - Y \text{ .... (1.5)}$$

$$Cb = B - Y \text{ .... (1.6)}$$

The Y component describes brightness; the other two values describe a color difference rather than a color, making the color space unintuitive. The transformation simplicity and explicit separation of luminance and chrominance components makes this color space attractive for skin color modeling. In contrast to RGB, the YCbCr color space is luma independent, resulting in a better performance.

2) High Level Analysis

After analyzing the image in low level and extracting the skin tone areas and the edges in an image, it is required to determine whether the patches are faces or not. To achieve this goal, Template Matching algorithm is used as the main face detector in the system.

- FD Template Matching Approach

In Template Matching, a standard pattern of face (normally frontal view) is stored first. Given an input image, the correlation values of several sizes of the standard pattern are calculated for the face contour, eyes, nose and mouth independently. The final determination for the existence of a face is done based on these correlation values. The approach has the advantage of being simplistic but has several disadvantages. One of them is it cannot deal with variation of faces in scale, shape and pose. Other approaches have been proposed to achieve scale and shape invariance such as deformable Template Matching [16].

1.3.2 Image-Based Approach

Taking advantages of the current advances in pattern recognition theory, the techniques in this group address FD as a general recognition problem. Image-based representations of faces, for example in 2D intensity arrays, are directly classified into a face group using training algorithms without feature derivation and analysis. Unlike the feature-based approach, these relatively new techniques incorporate face knowledge implicitly into the system through mapping and training schemes.

The feature-based FD techniques still have many problems in unpredictability of face appearance and environmental conditions that affect the system robustness. Although some of the recent feature-based attempts have improved the ability to cope with the unpredictability, most are still limited to head and shoulders.

Many researches have been presented in this approach such as Eigenfaces [17,18], Neural Networks (NNs) [19], Support Vector Machine (SVM) [20], Bayesian Approach [21] and Hidden Markov Model (HMM) [22] .

1.4 Skin Color Segmentation

Skin color is an important part in any FD system, so defining robust skin color segmentation algorithm is an important step and it will affect the next FD process. The skin-color information can be used as a pre-processing step to segment skin pixels in the face image from non-skin pixels and this will help in reducing the difficulties that can appears due to the effects of different human races, varying light sources and confusing backgrounds.

Detecting skin-color accurately is not an easy task, for this purpose a robust and reliable skin color model must be proposed which can be used in segmenting and detecting the skin region in different conditions and with different faces.

There are different approaches that can be used for color segmentation and here we mention to two widely used approaches:

(a) Skin color threshold based segmentation: Color segmentation can basically be performed using appropriate skin color thresholds where skin color is modeled through histograms or charts.

(b) Skin color Statistical measure based segmentation: More complex methods make use of statistical measures that model face variation within a wide user spectrum.

1.5 Face Recognition Applications

FR has many applications which are centered long two main primary tasks:

1. **Verification (one-to-one matching):** When presented with a face image of an unknown individual along with a claim of identity, ascertaining whether the individual is who he/she claims to be [23].
2. **Identification (one-to-many matching):** Given an image of an unknown individual, determining that person's identity by comparing (possibly after encoding) that image with a database of (possibly encoded) images of known individuals [23].

There are numerous application areas in which FR can be exploited for these two purposes, a few of which are outlined below.

- **Security:** access control to buildings, airports/seaports, ATM machines and border checkpoints [24]; computer/ network security [25]; email authentication on multimedia workstations.

- **General identity verification**: Electoral registration, banking, electronic commerce, identifying newborns, national IDs, passports, drivers' licenses, employee IDs.
- **Criminal justice systems**: Mug-shot/booking systems, post-event analysis, forensics.
- **Image database investigations**: Searching image databases of licensed drivers benefit recipients, missing children, immigrants and police bookings.
- **"Smart Card" applications**: In lieu of maintaining a database of facial images, the face-print can be stored in a smart card, bar code or magnetic stripe, authentication of which is performed by matching the live image and the stored template [26].

1.6 Face Recognition: Structure and Procedure

Given a picture taken from a digital camera, if we like to know if there is any person inside, where his/her face locates at, and who he/she is. Towards this goal, we generally separate the FR procedure into three steps: **Face Detection**, **Feature Extraction**, and **Face Recognition** (see Figure 1.4).

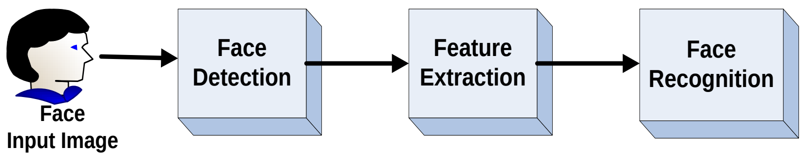


Figure 1.4: Configuration of a general FR structure.

1.6.1 Face Detection

The main function of this step is to determine (1) whether human faces appear in a given image, and (2) where these faces are located at. The expected outputs of this step are patches containing each face in the input image. In Chapter 4, we present a new technique for detecting and locating a face in different circumstances. The more important issue that must be mentioned in this chapter is that FD serves as the pre-processing for FR; FD could be used for region-of-interest detection, retargeting, video and image classification, etc.

1.6.2 Feature Extraction

After the FD step, human-face patches are extracted from images. Directly using these patches for FR have some disadvantages, *first*, each patch usually contains over

1000 pixels, which are too large to build a robust recognition system [3]. *Second*, face patches may be taken from different camera alignments, with different face expressions, illuminations, and may suffer from occlusion and clutter. To overcome these drawbacks, feature extractions are performed to do information packing, dimension reduction, salience extraction, and noise cleaning. After this step, a face patch is usually transformed into a *vector with fixed dimension* or a set of *fiducial points and their corresponding locations*.

1.6.3 Face Recognition

After formulizing the representation of each face, the last step is to recognize the identities of these faces. In order to achieve automatic recognition, a face database is required to build. For each person, several images are taken and their features are extracted and stored in the database. Then when an input face image comes in, we perform FD and feature extraction, and compare its feature to each face class stored in the database. In Figure 1.5, we show an example of how these three steps work on an input image.

There have been many researches and algorithms proposed to deal with FR problem, but before going into details of techniques and algorithms of FR, we'd like to make a digression here to talk about pattern recognition. The discipline, pattern recognition, includes all cases of recognition tasks such as speech recognition, object recognition, data analysis, and FR, etc.

The general structure of pattern recognition is shown in Figure 1.6 In order to generate a system for recognition; we always need data sets for building categories and compare similarities between the test data and each category.

A test data is usually called a "query" in image retrieval literatures. From Figure 1.6, we can easily notice the symmetric structure.

- <u>Starting from the data sets side:</u>

Step 1: Perform dimension reduction (i.e. feature extraction) on the stored raw data.

Step 2: After a dimension reduction, each raw data in the data sets is transformed into a set of features, and the classifier is mainly trained on these feature representations.

- <u>When a query comes in:</u>

Step 3: Perform the same dimension reduction procedure on it and enter its features into the trained classifier. The output of the classifier will be the optimal class (sometimes with the classification accuracy) label or a rejection note (return to manual classification).

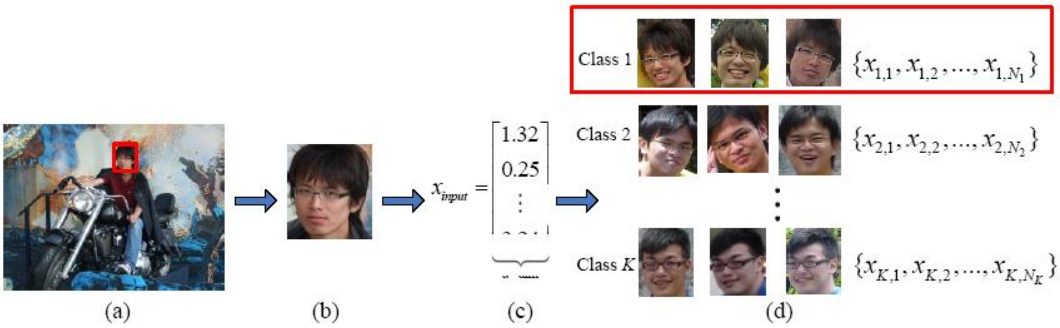


Figure 1.5: An example of how the three steps work on an input image. (a) The input image and the result of FD (the red rectangle) (b) The extracted face patch (c) The feature vector after feature extraction (d) Comparing the input vector with the stored vectors in the database by classification techniques and determine the most probable class (the red rectangle). Here we express each face patch as a *d*-dimensional vector, the vector $x_{m,n}$ as the n_{th} vector in the m_{th} class, and N_k as the number of faces stored in the k_{th} class [41].

1.7 Research Overview

In this section, we present detailed information about this thesis. First, we start by identifying the importance of this field in natural life including the motivations behind this study, objectives to be accomplished, methodology that has been followed, and our contributions throughout this work and finally, we show the content of this research.

1.7.1 Motivation

FR is one of the most successful applications of image analysis and understanding. This field is expected to improve many fields especially in security systems that can be applied in many places such as agencies and governments, etc. and provide the user ability to judge if the detected face person image is the same one stored in database. The main motivation for FR filed is because it is considered a passive, no intrusive system to verify and identify people [11].

Many recent events, such as terrorist attacks, exposed serious weakness in most sophisticated security systems such as (credit card verification and criminal identification). Various government agencies are now more motivated to improve security data systems based on body or behavioral characteristics, often called biometrics [6]. FR is an important issue in such security systems. For example, the ability to model a particular face and distinguish it from a large number of stored face models would make it possible to vastly improve criminal identification. However, different environment conditions such as detecting faces with different poses, different lightning conditions, complex backgrounds and detecting faces with glasses, tend to significantly affect system FR performances and leads to an inability to recognize faces correctly.

Understanding these vulnerabilities can lead to developing new techniques for improving recognition performance. Many techniques have been developed to solve these conditions [33, 62], but, they still have some defects that may affect the FR system performance. In this research a new technique will be used for FD and FR which will improve the recognition performance rates.

1.7.2 Objectives

This research explores the capabilities, strengths, and weaknesses of a new proposed technique based in feature-based approach for detecting face which will pave the way for using fast statistical approaches for FR.

The fundamental goal for this research is twofold. First, it explores a new technique for detecting face correctly from still images taking into account the different vulnerable conditions that can be presented in the image [1]. Second, it processes the detected face image as a preceding step for helping in recognizing the face using robust recognition techniques.

1.7.3 Methodology

This research composed of two main parts which are relevant to each other and in some cases, we can't separate them. Firstly, FD part which prepare for the second part which is FR.

It is worth mentioning that any FR technique can be broadly divided into three categories [2]:

i) Those that operate on intensity images,
ii) Those that deal with video sequences, and
iii) Those that require other sensory data such as 3D information or infra-red imagery.

In this research; we deal with intensity images, and this study is conducted in the following phases:

- Study the basic principal of colored-based images.
- Review of the existing skin segmentation approaches that can pave the way for implementing a robust and efficient FD technique.
- Review of the existing FD approaches.
- Determine the major vulnerable conditions in the still images that affect the FD process.
- Determine the types of the still images and which one can be selected in our research.
- Review of the existing FR approaches.
- Identify the main problems that affect the FR system.
- Design a new robust technique to FD under vulnerable conditions which can pave the way for implementing an efficient FR technique.
- Evaluate the proposed new techniques in different conditions.

1.7.4 Contributions

This thesis explores the obstacles that face any FD system. These obstacles varies from image to other but, they will affect with no doubt any FR system. Many researches done in this field from past years but this field is still need more and more work to solve all obstacles. Hence, it is all the more necessary for developers to incorporate methods of detecting face in the presence of different obstacles. The main contributions of this research are highlighted hereunder:

- This research, provide algorithms that can overstep a wide range of variations challenges that appears in FR systems to provide efficient results. These algorithms start from skin segmentation (filtering) till FD.
- We present a new FD algorithm which is deal with still images and works with different viewing conditions (profile and near profile faces) under different surrounded conditions such as complex background, various illumination conditions, different people's distractions (eye glasses) and different people races.
- We use FD the feature-based approach as the base approach in searching for important features located in the face. So, the face features are the main

components in our proposed algorithm and this leads to work with high-level rather than low-level stages.

- This work is different from another researches because, it uses feature-based approach for detecting different face features which are together form mixture of features that lead to detect the face correctly and propose a robust face algorithm deals with different environment conditions.
- This research uses existing techniques such as Template Matching to strength the proposed algorithm in dealing with different face poses.

1.7.5 Organization

We commence this thesis by providing a background in the FR filed and we mention to the importance of FD step to get a robust FR system. After that, we proceeds with introducing the research objectives, methodology, contributions and provide an overview of the remainder of this document. Chapter 2, provides a brief review about the FD and FR approaches related works done in this field previously. In Chapter 3, we present a brief survey of known techniques in FD and FR used in this field. Afterwards Chapter 4, contains comprehensive and formal description of this thesis work. We propose new algorithms for FD that deals with different viewing conditions (profile faces and near profile faces) and under different surrounded conditions. Finally, in the end of this chapter we mention to the FD technique used for FR in our thesis. Chapter 5 explains the evaluation of this research effort via testing the proposed algorithms in FD in relevant face databases. Also, we provide testing results for applying our used FR algorithm to the detected face after applying some processing steps on it. This chapter includes the discussion on these results and how they can give good results compared to other already implemented algorithms in this field. Finally, Chapter 6 concludes the thesis with a summary of the research findings, including important concepts; techniques behind this research effort, significance of research results and the future work can be done in this field.

Chapter 2

RELATED WORK

2.1 Overview

Various methods to FR have been proposed to resolve the variations occur with FR process [10,11,39, 40]. PCA[42], NNs[38], machine learning, information theory, geometrical modeling, (deformable) Template Matching, Hough transform, motion extraction, and color analysis are examples of these methods.

FR algorithms can be classified into two main categories according to feature extraction schemes: *feature-based* methods and *holistic appearance-based* methods [1,52]. In the first type, properties and geometric relations such as the areas, distances, and angles between the facial features like eyes, nose, and mouth are used as descriptors for FR. The use of geometric features for FR is popular in the earlier literature [75]. The advantages of this representation include: it drastically reduces the number of input dimensions and it is less sensitive to variation in illumination. Also, it can work real-time with minimal training. However, these methods are usually criticized for two reasons. First, geometric features are hard to be extracted in some complicated case. For instance, some eye localization techniques assume some geometric and textural models and do not work if the eye is closed. Besides that the lightening sources and skin colors also affect the extraction of the features. Second, geometric features alone are not enough to fully represent a face because rich information contained in the facial texture or appearance is discarded. These two reasons will lead to the important downfall of this category which is its inaccuracy.

2.2 Face Detection Approaches

2.2.1 Feature-Based Methods

- **Geometrical Features**

Many researchers have explored geometrical feature based methods for FR. Kanade [74] presented an automatic feature extraction method based on ratios of distances (between feature points such as the location of the eyes, nose, etc.) and reported a recognition rate of between 45-75% with a database of 20 people. Brunelli and Poggio [75] compute a set of geometrical features such as nose width and length, mouth position, and chin shape. They report a 90% recognition rate on a database of

47 people. However, they show that a simple Template Matching scheme provides 100% recognition for the same database. Cox, Ghosn and Yianilos [76] have recently introduced a *mixture-distance* technique which achieves a recognition rate of 95% using 95 test images and 685 training images (one image per person in each case). Each face is represented by 30 *manually* extracted distances.

Shih et.al [83], presents a novel approach for the extraction of human head, face and facial features. They try to locate a face in a normal lighting condition and using plain background and. However, it is difficult to detect a face in a complex background and under various lighting conditions.

Hsu et.al.[62] present a new approach for FD in color image in the presence of varying lightening conditions as well as complex background which gives a good results. Wu et.al [84] proposed an efficient face candidates selector algorithm for FD tasks in still gray level images. They use eye-analog segment information. But, this algorithm fail when detecting a person wears glasses.

Geometrical facial templates and the Hough transform were incorporated to detect grayscale frontal faces in real time applications [39]. Face detectors based on Markov random fields [48] and Markov chains [49], make use of the spatial arrangement of pixel gray values.

- **Template Matching**

Template Matching is another method that can be used in FR that fall into feature-based category. Template Matching such as [75] operates by performing direct correlation of image segments (e.g. by computing the Euclidean distance). Template Matching is only effective when the query images have the same scale, orientation, and illumination as the training images [76].

Bruneli et.al [79] automatically selected a set of four features templates, i.e., the eyes, nose, mouth, and the whole face, for all of the available faces. They compared the performance of their geometrical matching algorithm and Template Matching algorithm on the same database of faces which contains 188 images of 47 individuals. The Template Matching was superior in recognition (100 percent recognition rate) to geometrical matching (90 percent recognition rate) and was also simpler. The disadvantage of this method is that it is very slow when it is used alone in critical applications.

In this thesis, we try to make a better system using this feature-based approach which has all the advantages of feature-based machine and also is very accurate.

2.2.2 Appearance-Based Methods

On the contrary, holistic appearance-based methods emphasize preserving the original images as much as possible. In these methods, usually the global properties of the face image intensity pattern are used to extract information about texture and shape that are useful for distinguishing faces. In some direct comparisons of FR using feature-based and appearance-based representations, the appearance-based approaches outperformed the feature-based systems [75,82]. The weakness of this category is the large number of training that has to be performed on the system for faces and non faces, so that the system can detect faces in different background complexities. One of the most well-known appearance-based projection method is the Eigenfaces technique [17,18], which is based on PCA. In the late 1980s, Sirovich and Kirby [27] were the first to utilize PCA [28] to economically represent face images. They demonstrated that any particular face can be efficiently represented along the eigenpictures coordinate space, and that any face can be approximately reconstructed by using just a small collection of eigenpictures and the corresponding projections ('coefficients') along each eigenpicture.

The PCA method has been proved to discard noise and outlier data from the training set, while they may also ignore some key discriminative factors which may not have large variation but dominate our perception.

2.3 Face Recognition Approaches

The method for acquiring face images depends upon the underlying application. For instance, surveillance applications may best be served by capturing face images by means of a video camera while image database investigations may require static intensity images taken by a standard camera. Some other applications, such as access to top security domains, may even necessitate the forgoing of the nonintrusive quality of FR by requiring the user to stand in front of a 3D scanner or an infra-red sensor.

Therefore, depending on the face data acquisition methodology, FR techniques can be broadly divided into three categories: methods that operate on intensity images, those that deal with video sequences, and those that require other sensory data such as 3D information or infra-red imagery.

The following sections discussion will mention the methods in each category and attempts to give an idea of some of the benefits and drawbacks of the schemes mentioned there in general.

2.3.1 Face Recognition From Intensity Images

FR methods for intensity images fall into two main categories which are feature-based category and holistic category [55, 56].

i) Feature-Based Category

Feature-based approaches first process the input image to identify and extract (and measure) distinctive facial features such as the eyes, mouth, nose, etc., as well as other fiducial marks, and then compute the geometric relationships among those facial points, thus reducing the input facial image to a vector of geometric features. Standard statistical pattern recognition techniques are then employed to match faces using these measurements.

More sophisticated feature extraction techniques involve deformable templates [57], Hough transform methods [58] and morphological operations [65]. However, all of these techniques rely heavily on heuristics such as restricting the search subspace with geometrical constraints [54]. Furthermore, a certain tolerance must be given to the models since they can never perfectly fit the structures in the image. However, the use of a large tolerance value tends to destroy the precision required to recognize individuals on the basis of the model's final best-fit parameters and makes these techniques insensitive to the minute variations needed for recognition.

The main advantage offered by the featured-based techniques is that since the extraction of the feature points precedes the analysis done for matching the image to that of a known individual, such methods are relatively robust to position variations in the input image [53]. In principle, feature-based schemes can be made invariant to size, orientation and or lighting. Other benefits of these schemes include the compactness of representation of the face images and high speed matching.

The major disadvantage of these approaches is the difficulty of automatic feature detection and the fact that the implementer of any of these techniques has to make arbitrary decisions about which features are important [29].

ii) Holistic Category(Appearance-Based)

Holistic approaches are also called appearance-based methods attempt to identify faces using global representations, i.e., descriptions based on the entire image rather than on local features of the face. To be more clearly distinguished from feature-based methods, we can say that feature-based methods directly extract information from some detected fiducial points (such as eyes, noses, and lips, etc). During the past twenty years, holistic-based methods attract the most attention against other methods, so we focus more on this category which we use for FR.

2.4 Dimension Reduction Approaches

Dimension reduction is one of the most important steps in pattern recognition and machine learning. It is difficult to directly use the raw data (ex. face patches) for pattern recognition, not only because significant parts of the data haven't been extracted but also because the extremely high dimensionality of the raw data. Significant parts (for recognition purposes or the parts with more interest) usually occupy just a small portion of the raw data and cannot directly be extracted by simple methods such as cropping and sampling. The goal of dimension reduction is to extract useful information and reduce the dimensionality of input data into classifiers in order to decrease the cost of computation and solve the curse of dimensionality problem [36].

There are two main categories of dimension reduction techniques [41]:

1) **Domain-knowledge approaches**: Perform dimension reduction based on knowledge of the specific pattern recognition case. For example, in image processing and audio signal processing, the discrete Fourier transform (DFT) discrete cosine transform (DCT) and discrete wavelet transform are frequently used because of the nature that human visual and auditory perception have higher response at low frequencies than high frequencies.
2) **Data-driven approaches**: Extract useful features from the training data by some kinds of machine learning techniques. For example, the eigenface are one of major techniques that fall into this approach which determines the most important projection bases based on the Principal Component Analysis (PCA), which are dependent on the training data set, not the fixed basis like the DFT

or DCT. This approach is the one we use it in this thesis for dimension reduction in order to build a robust FR method.

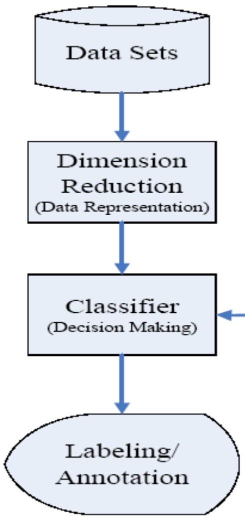


Figure 2.1: The general structure of a pattern recognition system [41].

Automatic FR (see Figure 2.1), can be seen as pattern recognition problem which is hard to be solved due to its nonlinearity. We can think of it as a Template Matching problem, where recognition has to be performing in a high dimensionality space. Since higher the dimension of space is, more the computations are needed to find the match. A dimensionality reduction technique is used to project the problem in a lower dimensionality space. These approaches can be categorized according to the projection methods used for feature extraction (dimensionality reduction) into linear and non-linear methods. PCA , LDA and Template Matching considered as linear methods. While, there are another non-linear methods used in FR field such as Artificial Neural Network (ANN) approach.

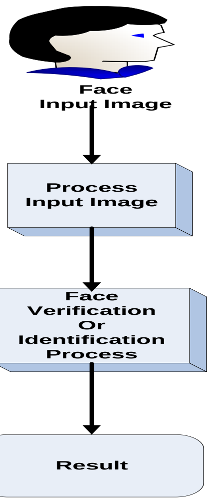


Figure 2.2: The major part of FR algorithm.

2.4.1 Linear projection methods

- **PCA and Eignfaces**

Eigen faces [63] can be considered as one of the first approaches in this sense. One important date for FR was beginning of the 90’s when Turk et.al [17,18] implemented the eigenfaces approach based on PCA algorithm, which is surely the most popular FR method [23] and the resulting eigenfaces are classified by comparison with known individuals.

Turk and Pentland [63] present results on a database of 16-subjects with various head orientation, scaling, and lighting. Their images appear identical otherwise with little variation in facial expression, facial details, pose, etc. For lighting, orientation, and scale variation their system achieves 96%, 85% and 64% correct classification respectively. Scale is re-normalized to the eigenface size based on an estimate of the head size. The middle of the faces is accentuated, reducing any negative effect of changing hairstyle and backgrounds.

Recently, in [79] experiments with ear and FR, using the standard PCA approach, showed that the recognition performance is essentially identical using ear images or face images and combining the two for multimodal recognition results in a statistically significant performance improvement. For example, the difference in the rank-one recognition rate for the day variation experiment using the 197-image training sets is 90.9% for the multimodal biometric versus 71.6% for the ear and 70.5% for the face.

In summary, it appears that eigenfaces is a fast, simple, and practical algorithm. However, it may be limited because optimal performance requires a high degree of correlation between the pixel intensities of the training and test images.

- **LDA and Fisher Faces**

The LDA (Linear Discriminant Analysis) [64] has been proposed as a better alternative to the PCA. It searches for those vectors in the underlying space that best discriminate among classes (rather than those that best describe the data). Indeed the LDA provides better classification performances only when a wide training set is available, and some results discussed by Martinez et.al [60], confirm this thesis.

Fisher's Linear Discriminant (FLD) is a widely used method for feature extraction and dimensionality reduction in pattern recognition. Peter et.al [33] develop a FR algorithm which is insensitive to large variation in lighting direction and facial expression. They use a projection method that based on Fisher's Linear Discriminant and produces well separated classes in a low-dimensional subspace, even under severe variation in lighting and facial expressions.

2.4.2 Non-Linear projection methods

- **Neural Network Approaches**

ANN, often just called NNs, is a mathematical model or computational model based on biological NNs. The attractiveness of using NNs could be due to its non-linearity in the network. It consists of an interconnected group of artificial neurons and processes information using a connectionist approach to computation. In most cases an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network during the learning phase [69].

One of the first NN techniques used for FR is a single layer adaptive network called WISARD which contains a separate network for each stored individual [80]. The way in constructing a NN structure is crucial for successful recognition. It is very much dependent on the intended application. For FD, multilayer perceptron [81] and convolutional NN [46] have been applied. For face verification, [61] is a multi-resolution pyramid structure. Lawrence et.al [83] proposed a hybrid NN which combines local image sampling, a self-organizing map (SOM) NN, and a convolutional NN. The authors reported 96.2% correct recognition on ORL database of 400 images of 40 individuals. The classification time is less than 0.5 second, but the training time is as long as 4 hours.

Much of the present literature on FR with NNs presents results with only a small number of classes (often below 20). For example, in [77] the first 50 principal components of images are extracted and reduced to 5 dimensions using an auto-associative NN. The resulting representation is classified using a standard multilayer perceptron. Good results are reported but the database is quite simple: the pictures are manually aligned and there is no lighting variation, rotation, or tilting. There are 20 people in the database.

Lin et.al [51] used probabilistic decision-based neural network (PDBNN) which inherited the modular structure from its predecessor, a decision based neural network (DBNN) [50].

The importance of such approach appears from building a system without defining "features" of a face manually and such approach can potentially be extended to detect other spatially well-defined objects since the system learns from the appearance of the training examples without any prior knowledge. However, the disadvantage of this approach is that it is difficult to collect enough training examples for such tasks. Moreover, there is no theory about the number of examples needed to train the system to achieve a certain detect rate [23].

These methods give a suitable results in some applications but, it fail in other practical applications; for example, the NN-based [43, 44, 68] and view-based [45] approaches require a large number of face and non-face training examples, and are designed primarily to locate frontal faces in grayscale images. Schneiderman and Kanade [47] extend their learning-based approach for the detection of frontal faces to profile views.

***What's more, there are some issues in building robust FR algorithm that have never been issued before, like*:**

- All the previously [63,82,83,84] success researches have been presented using eignfaces, neural networks, Template Matching and other researches in FR field failed to locate faces with some kinds of poses (such as a near-profile).

- Detect a face in a complex background and various lighting conditions or when a person wears glasses because these obstacles let the detection process is very complex.

- Using a hybrid of Template Matching method with feature-based method is a new idea. So by using this idea we can avoid the slowness of using Template Matching alone.

In summary, no existing technique is free from limitations. Further efforts are required to improve the performances of FR techniques. In this thesis, we suggest a new approach for developing new algorithms that address the above problems for obtaining robust FR algorithms. So, our proposed method detects a face with/without eyeglasses with various poses and under different environments conditions.

Chapter 3

BACKGROUND

3.1 Overview

FR is one of the few biometric methods that possess the advantages of both high accuracy and low intrusiveness. It has the accuracy of a physiological approach without being intrusive. Within the last several years, numerous algorithms have been proposed for FR [3, 32, 33, 63]. Two of the commonly discussed FR methods are Eigenface and Fisherface.

Any FR system faces many environment obstacles that affect the system output, the main one is variation in lightning conditions. Note that lighting variability includes not only intensity, but also direction and number of light sources. As showing in Figure 3.1, the same person, with the same facial expression which seen from the same viewpoint, can appear dramatically different when light sources illuminate the face from different directions.

Figure 3.1: The same person seen under different lightning conditions can appear dramatically different: In the left image, the dominant light source is nearly head-on; in the right image, the dominant light source is from above and to the right [33].

In this work, we search for an approach which takes into account the above considerations. This is done by finding a linear projection of the faces from the high-dimensional *image space* to a significantly lower dimensional *feature space* which is insensitive to variation in lighting direction. So, the main task in any FR system and which we do is to reduce image space dimension [67].

3.2 Face Recognition Approaches

In the following sections some techniques which can be used in reducing the high-dimensionality space are presented.

3.2.1 Eigenface and Principal Component Analysis

1) Principal Component Analysis (PCA)

One of the important techniques used for dimensionality reduction is PCA [64, 66], also known as Karhunen-Loeve method, which maximizes the scatter of all projected samples. PCA projection basis is computed from the dataset we have. Algorithm 3.1 presents how does PCA work.

Algorithm 3.1: PCA Algorithm procedure.

Algorithm 3.1: PCA Algorithm procedure

Purpose: Calculate the eignface representation of the face image.

Input: Given *N* *D*-dimensional vectors (In FR task, usually N < D).

Output: Get at least min (*N*-1, *D*-1) projection basis with one mean vector.

Procedure:

(1) Compute the mean vector Ψ (D-by-1 vector).

(2) Subtract each by Ψ and get ϕ_i

(3) Calculate the covariance matrix Σ of all the ϕ_{is} (D-by-D matrix)

(4) Calculate the set of Σ (*D*-by-(N-1) matrix, where each eigenvector is aligned as a column vector)

(5) Preserve the *M* largest eigenvectors based on their eigenvalues (*D*-by-*M* matrix *U*) .

(6) $U^T\phi_i$ is the eigenface representation (*M*-dimensional vector) of the *i*th face .

2) Eigenface Approach

The eigenfaces main advantage is dimension reduction; it seeks to answer the question of which features in feature space are important for classification, and which are not. This is done by using PCA of the images of the faces that reduces the dimensionality of the training set since it applies a linear projection of the image space to a low dimensional feature space, leaving only those features that are critical for FR [17, 18]. The PCA yields projection directions that maximize the total scatter across all classes, i.e., across all images of all faces and this will retain unwanted variations due to lighting.

The eigenface algorithm gave significant influences on the algorithm design for FR in the past twenty years, so it is a great starting point for readers to try building a FR system.

☒ **<u>Eignface Mathematical Overview:</u>**

More formally, let us consider a training set of N-sample images {x_1,x_2,…, X_N }, each of size m x n, and assume that each image belongs to one of c-classes {X_1, X_2 ,…, X_c }. Let us also consider a linear transformation mapping the original *n-dimensional image space* into an *m-dimensional feature space*, where m < n .

The new feature vectors $Y_k \in \Re^m$ are defined by the following linear transformation:

$$Y_K = W^T x_K \qquad K=1, 2 ,…, N \quad ………… \quad (3.1)$$

where,

$W \in \Re^{n\times m}$ is a matrix with orthonormal columns.

Assume the *total scatter matrix* S_T is defined as:

$$S_T = \sum_{K=1}^{N} (x_k - \mu)(x_k - \mu)^T \quad ……………..……. \quad (3.2)$$

where,

N is the number of sample images.

$\boldsymbol{\mu} \in \Re^n$ is the mean image of all samples.

Then, after applying the linear transformation W^T, the *scatter of the transformed feature vectors* { y_1, y_2, …., y_N } is :

$$W^T S_T W \quad …………………………..(3.3)$$

In PCA, the projection W_{opt} is chosen to *maximize* the determinant of the *total scatter matrix* of the projected samples, i.e.,

$$W_{opt} = \arg\max_{w} \left| W^T S_T W \right| = [\ w_1 \ \ w_2 \ \ …\ w_m\ ] \quad ……… (3.4)$$

where, $\{w_i \mid i = 1,2,....,m\}$ is the set of n-dimensional *eigenvectors* of S_T corresponding to the *m*-largest *eigenvalues*. These eignvectors can be thought of as a set of features that together characterize the variation between face images. Each image location contributes more or less to each eignvector. The dimension of the feature space is thus reduced to *m*.

Since these eigenvectors have the same dimension as the original images, they are referred to as *Eigenpictures* in [27] and *Eigenfaces* in [17, 18]. Figure 3.2 shows the result of applying PCA and Eignface approaches to dataset of images.

Each Individual face can be represented exactly in terms of a linear combination of eignfaces. Each face can also be approximated using only the "*best eignfaces*" those that have the *largest eignvalues*, and witch therefore account for the most variance within of the face set images. The best M-eignfaces span an M-dimensional subspace “face space” of all possible images.

☒ Face Recognition Using Eignfaces

We use eignface approach from holistic-based to FR in our work. The following steps describe how eignface work as shown previously in mathematical overview:

✓ **Initialization Process:**

1) Acquire an initial set of face images (the training set).
2) Calculate the eigenfaces from the training set, keeping only the M-images that correspond to the *highest eigenvalues*. These M- images define the face space. As new faces are experienced, the eigenfaces can be updated or recalculated.
3) Calculate the corresponding distribution in M-dimensional weight space for each known individual, by *projecting their face images* onto the *“face space”*.

✓ **Face Recognition Process:**

After initializing the system, the following steps are then used to recognize new input face images:

1) Calculate a set of weights based on the new input image and the M-eigenfaces by projecting the new input image onto **each** of the eigenfaces.

2) Determine if the image is a face at all (whether known or unknown) by checking to see if the image is sufficiently close to “face space" and this can be checked by calculating the Euclidean distances. The test face is recognized as the face of training set with the closest distance, if such distance is below a certain distance.

3) If it is a face, classify the weight pattern as either a known person or as unknown.

a) A database with only 10-faces and each face patch is of size 100-by-100.

b) Mean face through the computation of PCA basis.

c) Nine-eigenface through the computation of PCA basis (the order of eigenfaces from highest eigenvalues is listed from left to right, and from top to bottom).

Figure 3.2: Summary of applying PCA and Eignface approaches to image dataset[41].

3.2.2 Fisherface and Linear Discriminant Analysis

1) <u>Linear Discriminant Analysis (LDA)</u>

The objective of applying the LDA is to look for dimension reduction based on discrimination purpose as well as to find bases for projection that minimize the intra-class variation but preserve the inter-class variation.

LDA [28, 35, 59] searches for those vectors in the underlying space that best discriminate among classes (rather than those that best describe the data). More formally, given a number of independent features relative to which the data is described, LDA creates a linear combination of these which yields the largest mean differences between the desired classes [60].

2) <u>Fisherface Approach</u>

The Fisherface [33] algorithm is derived from the Fisher Linear Discriminant (FLD), which uses class specific information. By defining different classes with different statistics, the images in the learning set are divided into the corresponding classes. Then, techniques similar to those used in Eigenface algorithm are applied. Fisherface algorithm results in a higher accuracy rate in independent component axes. Each axis is a direction found by PCA.

The eigenfaces have the advantage of dimension reduction as well as saving the most energy and the largest variation after projection. Eigenface method *maximizes the scatter within the whole training set* so, it is expected to work effectively under idealized conditions but, it may suffer under variation in lighting direction. This happen because the points corresponding to the same class may not be well clustered in the projected space, or may be smeared with each other.

FLD maximizes the between-class variance as well as minimizes the within-class variance. A graphic example of FLD is shown in Figure 3.3 .

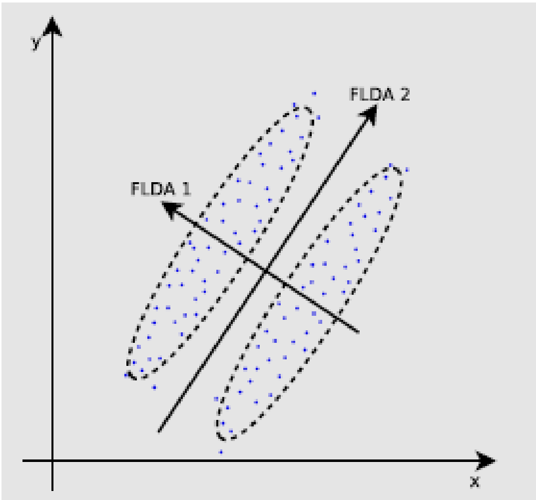


Figure 3.3: An example of FLD in two dimensions, showing the FLD axis that maximizes the separation between the classes and minimizes the variation inside the classes.

The LDA has been proposed as a better alternative to the PCA. It expressly provides discrimination among the classes, while the PCA deals with the input data in their entirety, without paying any attention for the underlying structure. Indeed the main aim of the LDA consists in finding a base of vectors providing the best discrimination among the classes, trying to maximize the between-class differences, minimizing the within-class ones.

FLD algorithm [34, 37, 53] uses LDA in its work to look for dimension reduction based on discrimination purpose as well as to find bases for projection that minimizes the intra-class variation but preserve the inter-class variation. FLD tries to “shape” the scatter in order to make it more reliable for classification. Because it uses a class specific approach in its work. It must be known that in Fisher algorithm the PCA is considered as a preliminary step in order to reduce the dimensionality of the input space, and then the LDA to the resulting space, in order to perform the real classification. Figure 3.4 shows the flow chart of LDA work in a FR system.

In our work, we use fisher algorithm which gives an efficient result in FR. In the following sub section we show how this algorithm can be used in FR system.

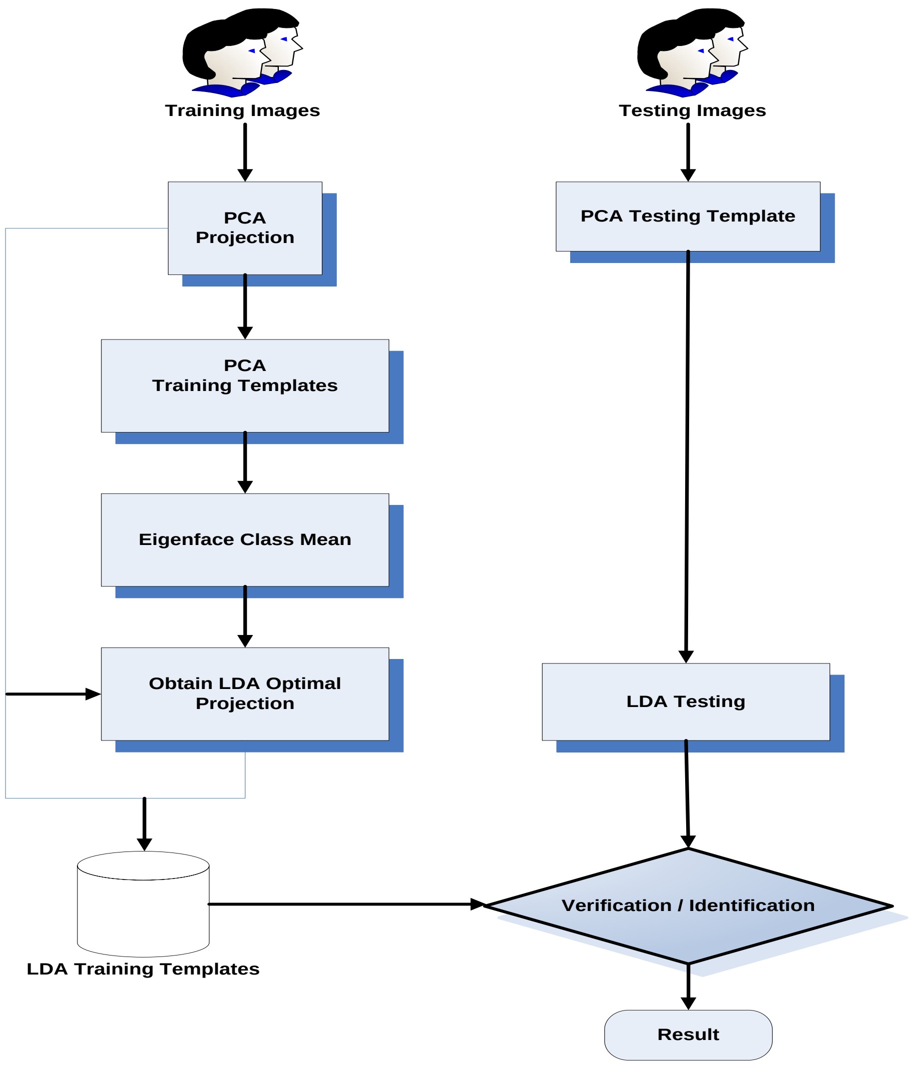


Figure 3.4: Simple flow chart of LDA algorithm.

☒ Fisher Method Mathematical Overview

This method selects W in eq. (3.1) in such a way that the ratio of the *between-class* scatter and the *within-class* scatter is *maximized*. One way to do this is to maximize the ratio $\frac{\|Sb\|}{\|Sw\|}$. Mathematically speaking, for all the samples of all classes, we define two measures:

1) Let the between-class scatter matrix be defined as:

$$S_B = \sum_{i=1}^{c} N_i (\mu_i - \mu)(\mu_i - \mu)^T \quad \text{.........................} \quad (3.5)$$

2) Let the within-class scatter matrix be defined as:

$$S_w = \sum_{i=1}^{c} \sum_{x_k \in X_i} (x_k - \mu_i)(x_k - \mu_i)^T \quad \text{..........................} \quad (3.6)$$

where,

$\boldsymbol{\mu_i}$ is the mean image of class X_i

N_i is the number of samples in class X_i .

If S_w is nonsingular, the optimal projection W_{opt} is chosen as the matrix with orthonormal columns which maximizes the *ratio* of the determinant of the between-class scatter matrix of the projected samples to the determinant of the within-class scatter matrix of the projected samples, i.e.,

$$W_{opt} = \arg\max_{w} \frac{\left|W^T S_B W\right|}{\left|W^T S_W W\right|} \quad \text{......................} \quad (3.7)$$

where, $\{W_i \mid i = 1,2,...,m\}$ is the set of generalized eigenvectors of S_B and S_w corresponding to the *m* largest generalized eigenvalues $\{\lambda_i \mid i = 1,2,...,m\}$, i.e.,

$$S_B W_i = \lambda_i S_W W_i, \qquad \text{i=1,2,…,m} \quad \text{.................} \quad (3.8)$$

Note that there are at most c - 1 nonzero generalized eigenvalues, and so an upper bound on m is c - 1, where c is the number of classes.

In the FR problem, one is confronted with the difficulty that the within-class scatter matrix $S_W \in \Re^{n \times n}$ is always singular. This stems from the fact that the rank of S_W is at most *N* - *c*, and, in general, the number of images in the learning set *N* is much smaller than the number of pixels in each image *n*. This means that it is possible to choose the matrix *W* such that the within-class scatter of the projected samples can be made exactly zero.

In order to overcome the complication of a singular S_W , eq.(3.7) shows an alternative to the criterion. This method, which we call Fisherfaces, avoids this problem by projecting the image set to a lower dimensional space so that the resulting within-class scatter matrix S_W is nonsingular. This is achieved by using PCA to reduce the dimension of the feature space to N - c, and then applying the standard FLD defined by eq. (3.9) to reduce the dimension to c - 1.

More formally, W_{opt} is given by:

$$W_{opt} = W_{fld}^T W_{pca}^T \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (3.9)$$

where,

$$W_{pca} = \arg\max_W \left| W^T S_T W \right| \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (3.10)$$

$$W_{fld} = \arg\max_W \frac{\left| W^T W_{pca}^T S_B W_{pca} W \right|}{W^T W_{pca}^T S_W W_{pca} W} \quad \ldots\ldots\ldots \quad (3.11)$$

Note that the optimization for W_{pca} is performed over $n \times (N - c)$ matrices with orthonormal columns, while the optimization for W_{fld} is performed over $(N - c) \times m$ matrices with orthonormal columns.

In computing W_{pca} , we have only the smallest c - 1 principal components. Note that there are at most c-1 generalized eigenvectors, therefore at most c-1 "fisherfaces". In Chapter 5, the testing results show that using fisherfaces with some image preprocessing will give an efficient result.

3.3 Face Detection Approaches

In the following section we mention to one of the most widely used approach in FD process which is Template Matching. In this thesis we use it with another algorithm for robust FD system that leads to robust FR system. There are many types of Template Matching (Template Matching using Cross-Correlation, Template Matching using deformable templates and others). There are many approaches which can be used for Template Matching such as Cross-Correlation, Two-Dimensional Logarithmic search, Hierarchical search, Sequential search. In this work, we use Cross-Correlation technique in this field because of its simplicity and ability to give an efficient results in this work. The following section shows how does it work to be used in FD process.

3.3.1 Template Matching By Cross-Correlation

The use of cross-correlation for Template Matching is motivated by the distance measure (Squared Euclidean Distance) which has the following equation:

$$d^2_{f,t}(u,v) = \sum_{x,y}[f(x,y) - t(x-u, y-v)]^2 \ldots\ldots\ldots\ldots\ldots \quad (3.12)$$

where, *f* is the image and the sum is over x; y under the window containing the feature positioned at (u , v).

Template Matching is conducted by searching for the location (u , v) for which $d^2_{f,t}(u,v)$ is minimum. In the expansion of d^2 we get:

$$d^2_{f,t}(u,v) = \sum_{x,y}[f^2(x,y) - 2f(x,y)t(x-u, y-v) + t^2(x-u, y-v)] \ldots\ldots\ldots\ldots \quad (3.13)$$

We notice that the term $\sum t^2(x-u, y-v)$ is constant. If the term $\sum f^2(x,y)$ is approximately constant then the remaining cross-correlation term:

$$C(u,v) = \sum_{x,y} f(x,y)t(x-u, y-v) \ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (3.14)$$

is a measure of the *similarity* between the image *f* (*u*, *v*) and the template *t* (*u*,*v*). The minimum of $d^2_{f,t}(u,v)$ is achieved when C(u,v) is *maximum* for all possible locations (u, *v*).

In cases for which the assumption of little gray-level variation is not valid, this measure is very sensitive to gray-level variations within *f* (u,v). In such cases the cross-correlation coefficient, defined as

$$C_N(u,v) = \frac{C(u,v)}{\sqrt{\sum_x \sum_y |f(u,v)|^2 \sum_x \sum_y |t(u,v)|^2}} \quad \ldots\ldots\ldots \quad (3.15)$$

is a more appropriate measure. Here, C_N (m, n) is a normalized version of c(u,v), and variations in *f* (u,v) tend to cancel out. The cross-correlation coefficient value C(u,v), which we choose in this work for classifying a region as a face is defined by experiment. From training and experiments we reach that a good threshold value for classifying a region as a face is if the resulting correlation coefficient value is greater than **0.6**.

Chapter 4

THE PROPOSED FACE DETECTION TECHNIQUE

4.1 Overview

As known, in any FR system; FD is the first step to be done. In this chapter we propose a new algorithm to be used in this field and give good results. In our work, we propose algorithm for detecting the person face region, either the person wear eyeglasses or not. Also, this algorithm focus on the a person head pose so, it deal with the person with different poses. Another point the algorithm takes into its consideration is the illumination condition, it tries to detect the person face region if the face is falling under various environments conditions. As we mention in Chapter one, FD approaches falls into two categories Feature-Based and Image-Based approach. In this thesis, we use feature-based approach for our FD process with some scenario that can give us the intended goal which is " Detecting a person face despite large changes in the visual stimulus such as different illumination conditions, different head poses, different distractions".

Figure 4.1, shows the block diagram of the proposed FR process. The following sections introduce the main steps and parts in our algorithm and how it will work.

4.2 Learning Skin Color Model Parameters

In this section, we avoid the traditional approach that uses skin color thresholds for face segmentation, because this approach doesn't work efficiently in different lighting conditions. We propose a new model for skin color classifying that classify skin pixels based on statistical measures that model face variation within a wide user spectrum.

This proposed model based on using *Gaussian distribution* to represent a skin color cluster. The Gaussian distribution is characterized by its mean (μ) and covariance matrix ($\sum$). Pixel color from an input image can be compared with the skin color model by computing the Mahalanobis distance. This distance measure gives an idea of how close the pixel color resembles the skin color of the model. The advantage of this statistical color model is that color variation of new users can be adapted into the general model by a learning approach.

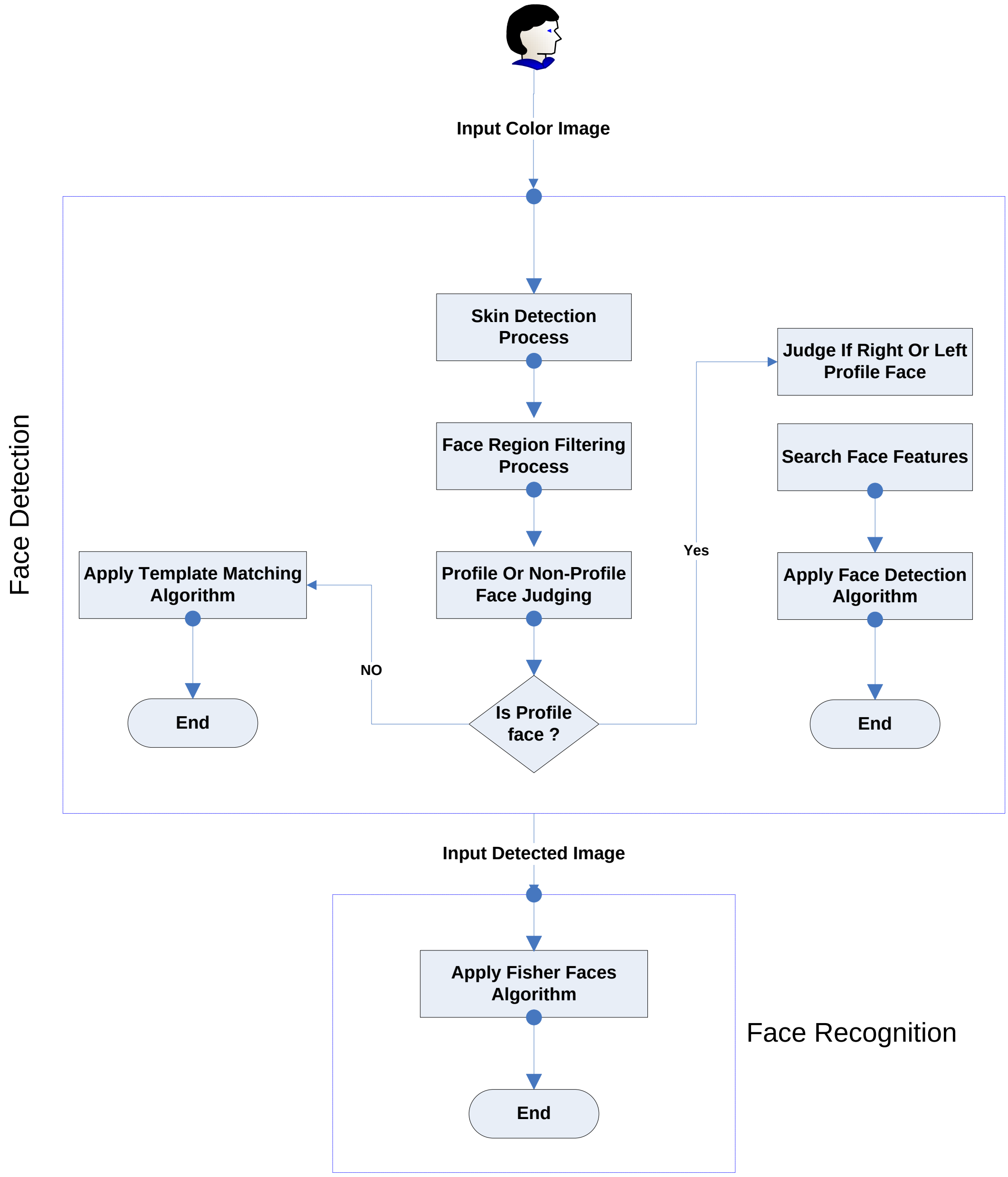


Figure 4.1: Block diagram of the proposed FR scenario.

This proposed model uses a number of skin samples taken from persons of different races (Asian, African, European) to determine the color distribution of human skin in color space. These skin samples are taken from the GTAV [72] and FEI [73] frontal face databases. We segment a total of **9,000 (40*30)** skin samples from these face databases images to be used as a training examples to determine the color distribution of the skin color model. In this work, we take only red and blue chrominance since the human faces dominant colors chrominance are red and blue (cr,cb).

Figure 4.2 shows the color distribution of this skin color model for different selected persons with different races. From Figure 4.2, it can be shown that the proposed skin color distribution from the training example selected previously can be represented by a Gaussian model.

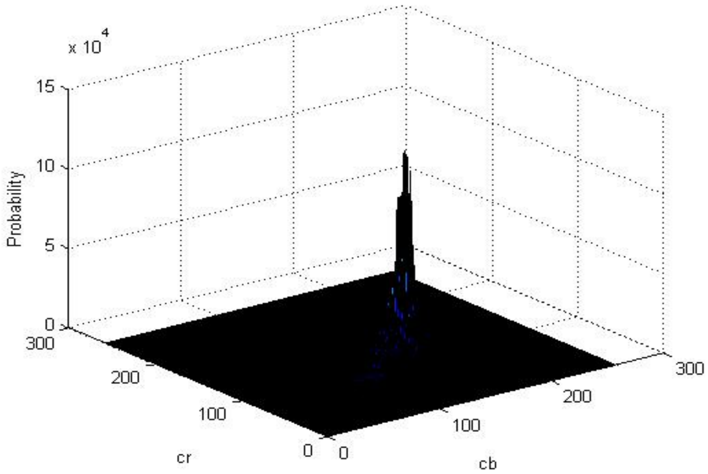


Figure 4.2: Color distribution for skin-color of different people.

So, the distribution of this training dataset is over-fitted to Normal (Gaussian) model, because when choosing images from different races we get a distribution similar to Gaussian. The result of this over-fitting is shown in Figure 4.3 .

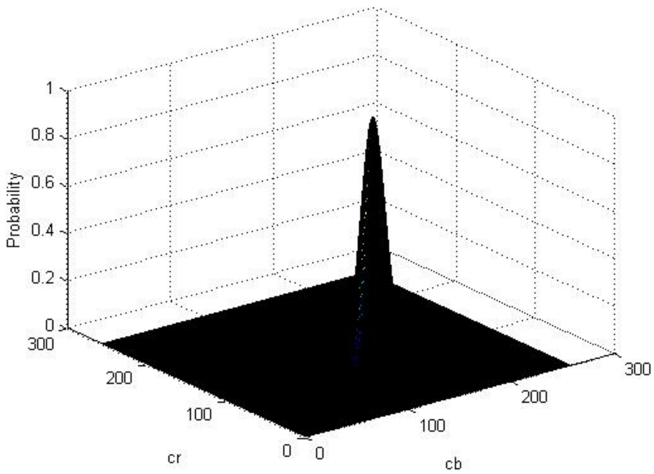


Figure 4.3: Fitting skin color into a Gaussian Distribution.

Then the prosperities of Gaussian are applied to this model and so, calculation of Mean and Covariance can be done simply as follows:

1) **Mean**: $m = \sum(x)$ ………………………………..…….………… (4.1)

where:

$x = (cr, cb)^T$,

cr: Red color component of color space,

cb: Blur color component of color space,

2) Covariance: $C = \sum\{(x-m)(x-m)^T\}$………………………….………… (4.2)

where:

$x = (cr, cb)^T$,

cr: Red color component of color space,

cb: Blur color component of color space,

m: Gaussian model mean property.

4.3 Proposed Skin Segmentation

After obtaining the previous skin color model, we can now obtain the gray likelihood of skin for every pixel (The gray value at each pixel shows the likelihood of the pixel belonging to the skin) of an image using multivariate Gaussian equation obtained using the previous color model.

If a pixel, having transform from RGB color space to chromatic color space, has a chromatic pair value of (r,b), the likelihood of skin for this pixel can then be computed as follows:

$$\textbf{\textit{Likelihood}} = \textbf{\textit{exp}}\ \{(-0.5)(x-m)^T C^{-1}(x-m)\}, \ldots\ldots\ldots\ldots\ (4.3)$$

where,

$x : (r, b)^T$, m : Mean and C^{-1} : Covariance Inverse Matrix

Equation 4.3 results a gray scale image between [0-255] and each pixel represents the likelihood of skin in this pixel. Using *adaptive thresholding* approach, an appropriate threshold value is chosen, so that the gray scale images (Likelihood images) can then be further transformed to a binary image presenting skin regions and non-skin regions, which gives a quite good result. In some cases, the segmentation procedure doesn’t present the face region only but, rather than there are some regions appears in the binary segmented image that doesn't represent a face region. This happen because of not adjusted illumination conditions and due to different people races. In order to obtain only face regions, we do some filtering steps on the binary segmented image to exclude the non-face regions from the image.

4.3.1 Image Filtering

After segmenting the image to **skin** and **non-skin** regions, it may contain some regions that don’t belong to a face but have the same face skin color (such as backgrounds, clothes… , etc), in this step we do some candidate filtering operations to remove these irregular regions. Some of these filtering operations are already exists and others is presented by us to enhance the segmentation operation. The following topics present these filtering operations.

1) <u>Number of Holes Inside A region</u>

An important step in any face segmentation algorithm is its ability to determine the human face regions from the image and eliminating non-face regions. In this step, we use the image topological properties which are useful for global descriptions of regions in the image plane. This operation not a new one, it is used before in many image processing researches. It calculates the number of the connected components in the image and based on this description, an accurate decision will be given if the tested region should be removed or not.

The *Euler number* can be used to remove irregular regions. The Euler number of the region can be computed to define the number of holes inside the region which is defined as:

$$E = C - H \text{ …………..…….. (4.4)}$$

where,

E: is the Euler number.
C: The number of connected components.
H: The number of holes in a region.

We base on a rule in order to take the advantage of Euler formula. Any face has some characteristics that describe it. One of the important characteristics in the face are the eyes appears on it either it is frontal or profile face. The frontal face image has two eyes appears in it while, the profile face image has one eye appears in it. These eyes are converted to holes after segmentation process and from this idea we conclude the following fact. The fact states that: "*any skin region in image should have at least one hole inside that region in order to be a face region*". So, if any region doesn't has at least one hole it will be deleted. This fact is tested under different databases and gives correct filtering usually but, this fact may fail in some cases.

2) Width and Height of a region

Some segmented face images has regions that have irregular shapes and they must be deleted in order to have only face region. To remove these irregular regions we propose a filtering step that depend on determining the width and height of the region bounding rectangle and delete the regions that shape like a vertical thin bar or horizontal thin bar as follows:

- If the region bounding rectangle width or height < Width_Heigh_Th1 ➔ Delete it

3) Geometrical Properties:

In this step, we focus into the geometrical properties that can be used to help us for obtaining a robust face segmentation algorithm. Our work in this step present two geometrical properties that give a good result in our proposed segmentation algorithm.

A) Solidity Property:

This not a new property but we use it because it describe the solidity of the regions in the segmented image so, we can take its advantages in our segmentation algorithm. The solidity of any region in the binary segmented image is an important geometrical feature. Based on this geometrical property, we take the solidity of the face regions into consideration as follows:

- If the region solidity < **=** ***Solidity_Threshold*** ➔ Delete the region.

B) Aspect Ratio Property:

In this filter we take into consideration another point which is that any region shape in segmented image mustn't looks like a horizontal or vertical thin bar. The idea comes from that the face shape has only circle and oval shapes. So, if any region shape look like a horizontal or vertical thin bar this means it doesn't represent a face region and it must be removed. From geometrical prosperities we can know if the region has a horizontal or vertical shape and this based on the aspect ratio of the region as follows.

i) Horizontal Thin Region:

The region is horizontal thin bar if the region width is very large compared to region height. In this work, we suggest a ratio threshold that judge if the region is horizontal thin bar as follows:

- If (**x / y**) <= ***Horizontal_Threshold*** ➔ Delete the region.

ii) Vertical Thin Region:

The region is vertical thin bar if the region high is very large compared to region width. In Our work, we suggest a ratio threshold that judge if the region is vertical thin bar as follows:

- If (**y / x**) >= ***Vertical_Threshold*** ➔ Delete the region.

4) Shoulders Removal:

The aim from these filtering steps is to detect a correct face region only. Depending on this idea, in this step we present a technique for removing shoulders from the face image if exist either the face is profile or non-profile. This filtering step will help in obtaining face region only. The shoulders removal process consists of two stages: Shoulders searching and Shoulders Elimination.

✓ Shoulders Searching

We depend on the fact that "*The shoulders part of people is wider than the head and neck parts*". For testing and determining if the image has shoulders or not; we follow certain steps that lead us to judge if the image contains shoulder parts or not.

1) A *horizontal projection* for the face skin pixels of a face image is taken as shown in Figure 4.4(b).

2) Determine the *middle line* that partitions the horizontal projected image to upper and lower regions, as shown in Figure 4.4(b). The process of determining the middle line will be shown in shoulder elimination topic.

3) Depending on this line, calculate the number of pixels that exist below and above the middle line. These pixels used to judge if the shoulders exists or not as follows:

"In most cases, If *the number of the skin pixels* ***below*** *the middle line part of the whole skin region larger than the number of skin pixels* ***above*** *the middle line* then *the shoulders should be exist* ". This is tested experimentally and it give a good results with several databases.

4) After this calculation; we find the relation between upper part and lower part pixels using (*Lower pixels / Upper pixels*) ratio and depending on this ratio the algorithm judge if the shoulders exists or not as follows:

- If lower/Upper > ***Shoulder_Threshol*** ➔ Shoulders exists.

a) Right Profile image

b) Horizontal projection Histogram

c) The middle line partitions the horizontal projected image in (b).

Figure 4.4 : The horizontal projection of a profile face.

✓ Shoulders Elimination:

After we make sure that the image have shoulders then, the next step is to remove these shoulders. The removing step requires a reference point to do the cutting (removing) thresholds start from it. In our work, we select the neck point to be this reference point and the following steps used to determine the neck point pixel coordinates and then removing the shoulders below this point to hold the face region only without shoulders. Figure 4.5, shows an illustration example for shoulders elimination. The following steps describe the details of this step.

- *Draw* a line passing between the beginning point and maximum point of the previous calculated horizontal projection values calculated in shoulders searching step. Figure 4.5(b) shows the image after applying all the filtering steps and Figure 4.5(c) shows this line.
- Then, *project* the skin pixels segmented after previous filtering steps and which are below this line on this line.
- The pixel with the *maximum distance* from this line will be the neck point, and this step gives a good results in most cases and under different databases.
- *Delete* the face pixels below this neck point row. Figure 4.5.(d) shows the shoulders elimination result.

a) Original image

b) Segmented Image

c) Line passing between the beginning point and maximum point

d) Face after Shoulder removal

Figure 4.5: Successful shoulder removal process.

4.4 Profile or Non-Profile Face Judging

The following steps are used for judging if the candidate face region is profile or not.

1) *A vertical projection* for the face skin pixels of a face image is taken.
2) Determine the region in projected image with maximum area. We choose the maximum area because it will represent the face region only. So if there are any small region appears due to false segmentation they must be deleted as shown in the next step.
3) Remove all regions in the vertical projected image that are less than the maximum area.
4) Draw the vertical line which passes through the center of the region with maximum area.
5) Calculate the area of the two new generated/splited regions which are to the left and right of the vertical line passing through the center of the vertical projected histogram.
6) Determine if the face is profile or not according to the following steps that work correctly in most cases but, this depends on the segmentation process so this procedure fails in other cases.

Step 1: Calculate the ratio between the right region and left region around the vertical line.

Step 2: If the Right ratio is larger than the left ration then the image contains a right profile face.

Step 3: If the Left ratio is larger than the right ratio then the image contains a left profile face.

Step 4: If the right ratio equals the left ratio then, the image contains a non-profile face.

4.5 Face Feature Searching

After segmenting the face and maintaining only the pixels from the still image that represent the face skin region, the main next step is how to detect face features that can lead us to detect the face region exactly from the still image. The work done on this stage contains a lot of alternative choices which can be used for FD process such as (NNs , Eigenface, SVM, …etc.). These choices fall into two major categories feature-based and image-based. In our work, we focus in using the "*Feature-based*" approach with support of "*Template Matching*" strategy for proposing a robust FD technique. Template Matching is a famous technique and can be used separately for FD but, we try to be different so, a hybrid of feature-based approach with Template Matching approach is used for obtaining the face features. Our proposed work depends on detecting high-level features from the image, so we are not going into detecting low level features, instead we are looking to detect high level features and these detected features will lead to a stable judge if the tested image has a face or not regardless of availability of many features that take a huge work in filtering and isolating the important features and rejecting the unnecessary features.

Our proposed technique detects the face in different poses, this is done by using feature-based technique for detecting faces in profile situation, and we use template-matching for detecting faces in non-profile situation. In the following sub-section we show how each technique can be used in this work and what the features that can be generated by each technique.

4.5.1 Feature-Based Approach

Using this approach, we search for high level features which can be used to judge if the still image has a face or not depending on these features. Before extracting face features, we want to give some hint which must be taken into consideration about the still images we work on them. The hint is that the still images are divided into two parts as follows:

1) **Profile** images which are divided into two categories:
- Right Profile image.
- Left Profile image.

2) **Non-Profile** images.

The face has a lot of features to be detected; we work on a profile images and we take one case for discussion which is the *right profile* images. While, the *left profile* images have the same steps and we don't discuss it but, experimentally we test it. In our proposed FD algorithm we are looking for detecting the following features:

i) Nose point.

ii) Chin Point.

iii) Nose features which are (The points above and below the Nose).

i) *Nose above* point.

ii) *Nose Bottom* point.

iv) Neck Point.

v) Head point (i.e. the top start point of the person skin).

vi) The height of the segmented skin region.

vii) The End of the person skin region.

viii) The vertical distance between head point and nose point.

ix) The vertical distance between Nose point and Chin point.

x) The vertical distance between the nose above point and the chin point.

xi) The vertical distance between nose point and nose bottom point.

xii) Angle between the three points (Nose point, Nose above and Nose below).

The following sections show what these features are and how to detect these features in order to detect the face based on them.

i) Nose point

The Nose point is one of the high level features. We use Nose point as a base feature for detecting other features. This point is the one with the furthest right distance of the face skin region segmented previously. This point calculated according to the following steps:

Step 1: Compute the distance from each skin region pixel (x_1,y_1) and the y-axis (x_2,y_2) according to the following distance equation:

$$\Delta \boldsymbol{x} = x_2 - x_1 \quad \text{(4.5)}$$

$$\Delta \boldsymbol{y} = y_2 - y \quad \text{(4.6)}$$

$$\boldsymbol{Distance} = \sqrt{\Delta x^2 + \Delta y^2} \quad \text{(4.7)}$$

Step 2: In most cases, the point with the maximum distance will be the *Nose point*. Figure 4.6 (b) shows the vertical line passing through the Nose point in the segmented image.

a) Original Image b) Vertical line passing through Nose point

Figure 4.6: The vertical line passing through the Nose point of the face.

ii) Chin Point

The chin point is another feature we use in our FD algorithm. This point is calculated based on the Nose point calculated previously. The following steps show the steps for detecting this point:

Step1: After detecting the vertical line passing through the nose point previously, a rotation to this line clockwise around the nose point is done until the length of the intersection segment between the rotated vertical line and the segmented skin region is larger than some *chin threshold*.

The Chin termination criterion is a procedure used to determine the chin point and it is calculated according to the following pseudo code algorithm described in Algorithm 4.1 .

Figure 4.7, shows the rotated vertical line by 25^o and it is clearly that this line rotated around the nose point.

Step 2: Calculate the distance from each pixel (x_1,y_1) of the segmented skin region below the nose point and to the right side of the rotated vertical line (x_2,y_2) resulted in step1, according to the Euclidean distance equations 4.5-7.

Step 3: In most cases, the point with the maximum distance is considered as the chin point. Figure4.7(c) shows the line passing through the chin point.

Algorithm 4.1: Chin termination criterion determination.

Purpose: Determine the Chin termination criterion.

Input:

Row_{count} = The pixels count in each row of the intersection segment.

$Segment_{Sum}$ = The sum of all the number of pixels calculated for all rows in the intersection segment.

$Segment_{Mean}$ = The mean of the $Segment_{Sum}$.

Output:

Procedure:

- Rotate the vertical line by 5^{o} .
- Initialize:

  $Row_{count} = 0$

  $Segment_{Sum} = 0$
- *For* each row of the intersection segment, which is reside between the rotated vertical line and the face segmented skin

  Calculate Row_{count} .

  $Segment_{Sum} = Segment_{Sum} + Row_{count}$.

  End
- Calculate the $Segment_{Mean}$.
- Let the Chin_Threshold = $Segment_{Mean}$.

If ($Segment_{Mean}$ > **0.2** and $Segment_{Mean}$ <**1.2**)

Stop rotation of the vertical line.

Else

Continue rotation.

End

End

a) Profile image.

b) Rotated vertical line by 25^{o} around the "Nose point" which satisfies the "Chin threshold".

c) The Detected Chin Line row.

Figure 4.7: Determining the Nose point.

iii) Nose Features

Till now, we are detecting two critical features for detecting profile images. The face also contains many features that can be detected. Nose is one which still has a lot of features that can be detected such as (Nose length, Nostril, Nose size, Nose shape, Nose above point, Nose below point, etc.). In this work, we focus on detecting only two features we see that they help us in detecting the face region which are (Nose *above* and Nose *Bottom* points). The following next topics will discuss these two features.

i) Nose Bottom Point

This point is the one that resides below the nose point and the nostrils reside above it. The following procedure is used to detect this point.

Step 1:Determine the line (*NCLine*) passing between Nose point and Chin point. Figure 4.8, shows the line passes between nose point and chin point. In Figure 4.8, this line is marked by red arrows.

Figure 4.8: Line passing from Nose point to Chin point

Step 2: Calculate the Euclidean distance from each pixel (x_1, y_1) reside to the left of the *NCLine* and the *NCLine* (x_2, y_2), according to the Euclidean distance equations 4.5-7.

Step 3: In most cases, the point with the maximum distance from the *NCLine* considered as the "*Nose Bottom point*".

Figure 4.9 shows the horizontal line passing through this point and the red arrow points toward this point.

Figure 4.9: Line passing through nose bottom point.

ii) Nose Above Point

This point is the one that resides above the nose point peak and the nostrils reside below it. The following procedure is used to detect this point.

Step 1: Locate the Nose point feature pixel which determined previously.

Step 2: Draw a circle such that the nose point defined as its center and with a radius defined as the distance from the nose bottom point to the circle center.

Step 3: In most cases, the top point of the face skin region touched the circle is considered as the *"Nose above point*". Figure 4.10, illustrates the Nose features that will be used in our FD algorithm in a color image. While, Figure 4.11 shows the line passing through the Nose above point in the segmented face image and the red arrow point toward the Nose above point.

Figure 4.10: Nose above and Nose bottom points.

Figure 4.11: Line passing through nose top line.

iv) Neck Point

Neck point is one of the features used in this work. The following steps summarize the procedure for detecting this point.

Step 1: A *horizontal projection* for the face skin pixels in a face image is taken as shown in Figure 4.12 (b).

Step 2: Draw a line passing between the beginning point and maximum point of the previous calculated horizontal projection values calculated in previous step as shown in Figure 4.12 (c).

Step 3: Project the segmented skin pixels which are below this line on this line.

Step 4: The pixel with the *maximum distance* from this line will be the *neck point* as shown in Figure 4.12 (d). Algorithm 4.2, shows the procedure used to determine this point.

Algorithm 4.2: Procedure used to determine the neck point

Let

Beginning_Maximum_Line = a line passing between the beginning point and maximum point of the horizontal projection histogram.

Begin

- **Initialize**
    - Max_Distance = 0
    - A *horizontal projection* for the face skin pixels in a face image is must be taken as shown in Figure 4.3.
- *For* each skin pixel below Beginning_Maximum_Line

  Distance = Calculate the distance between the pixel and the Beginning_Maximum_Line using Euclidean equation.

  If (Distance > Max_Distance)

  Max_Distance= Distance

  End

End

v) Skin Region Start (Head Top)

This point is the point which resides at the start of the face forehead, or on other words it resides on the top start point of the person skin. The following steps are used to determine this point.

a) Original image

b) Image horizontal projection

c) Line passing between the beginning point and maximum point of the projection histogram

d) Neck point in the projection histogram.

Figure 4.12: Neck point marking process.

Step 1: Locate the Nose point feature pixel row (*NoseRIndex*).

Step 2: apply the following pseudo code in Algorithm 4.3.

Algorithm 4.3: Determine the start of the skin region

Let

$\text{RowPixels}_{\text{Count}}$ = Count of the pixels in a row.

Begin

- **Initialize**

  $\text{RowPixels}_{\text{Count}}$ = sum (*NoseRIndex*)

- *For* each row starting from the *NoseRIndex* to the top of the skin region

  If $\text{RowPixels}_{\text{Coun}}$ equal zero

  Stop.

  Else

  Continue.

  End

End

vi) Skin Region end (Neck Row)

This point used to determine the end of the skin region. Before defining a procedure for detecting this point, we ask our selfs why we don't choose the chin point as the end of the skin region?. The answer is that in some cases the segmentation process doesn't work efficiently and so the chin point doesn't give a good indication to the end of the skin region.

The skin region end point is the one which resides at the end of the face region. In another words it the point that resides on the end of the neck of the person. The following procedure is used to determine this point.

Step 1: Locate the Chin point feature pixel row (ChinRIndex).

Step 2: Apply the pseudo code in Algorithm 4.4.

Algorithm 4.4: Determine the end of the skin region

Let

$RowPixels_{Count}$ = Count of the pixels in a row.

Begin

- Initialize

  $RowPixels_{Count}$ = sum (*ChinRIndex*)

- *For* each row starting from the *ChinRIndex* to the end of the skin region

  If $RowPixels_{Coun}$ equal zero

  Stop.

  Else

  Continue.

  End

End

vii) The Height of The Segmented Skin Region

After detecting the top and end of the segmented skin region, it is easily now to detect the height of this region by taking the absolute value after subtracting the index of the end row coordinate of the skin region from the index of start row coordinate of the skin region as follows:

***Height_Face_Region* = | *Skin End Row* - *Skin Start Row* |**

viii) Distance Between Head Point and Nose Point

Also, the vertical distance between the start of the skin region (head start) and the nose point is a good feature that helps us in our FD algorithm. This distance can be calculated by subtracting the index of the nose point row coordinate from the start of the skin region row coordinate and taking the absolute value of the subtraction as follows:

***Nose_Head_Distance* = | *Nose Point Row Index* - *Skin Start Row* |**

ix) Distance Between Nose and Chin Points

The vertical distance between the Nose point and the Chin point is a base feature which used in our FD algorithm. This distance can be calculated by subtracting the index of the chin point row coordinate from the index of the nose point row coordinate and taking the absolute value of the subtraction as follows:

***Nose_Chin_Distance* = | *Chin Point RowIndex* – *Nose Point Row Index* |**

x) Distance Between Nose Above Point and Chin Point

The vertical distance between the one of the Nose features which is *Nose above* feature point and the *chin* point is one which will be an efficient feature that we use it in our FD algorithm. This distance can be calculated according to the following equation as follows:

***NoseAbove_Chin_Distance* =** **| *Chin point Row Index* - (*Nose Row Index* – | *Nose Row Index*- *Nose Bottom Row Index*|) |**

xi) Distance Between Nose Point and Nose Bottom Point

The vertical distance between the Nose point y-index coordinate and the nose bottom x-index coordinate is an important feature that helps us in our FD algorithm.

This distance can be calculated by subtracting the y- index of the nose point row coordinate from the nose bottom x-index coordinate and taking the absolute value of the subtraction as follows:

Nose_NoseBottom_Distance = | Nose y-Index – Nose Bottom X-Index|

xii) Angle Between (Nose, Nose Above and Nose Bottom) Points

The Nose, Nose above and Nose bottom points together compose a triangle and each vertex represents one of these points. In our work we use Heron's formula which splits a triangle in two triangles with a corner of 90^o degrees and then uses sines function to calculate the needed angles.

The Nose angle is the one that restricted between Nose Top and Nose Bottom points. We calculate this point and call it ("*NoseAbove_Nose_NoseBottom*"). Figure 4.13 shows this point marked with red bubble.

Figure 4.13: The main three angles in my FD algorithm.

4.6 The Proposed Feature-Based FD Algorithm

The proposed algorithm will evaluate some rules and thresholds that judge if the image indicates a face or not and these rules will use the previously calculated features. We do a lot of experiments studying on how we can define these thresholds and rules, and from training and testing we reach that in most cases the following rules are correctly satisfied to judge if these calculated features represent a face or not:

- ✓ **Rule 1:** NoseAbove_Chin_Distance > (Height_Face_Region / **4**) .
- ✓ **Rule 2:** Nose_NoseBottom_Distance < (**0.5** * Nose_Chin_Distance) .
- ✓ **Rule 3:** (0.9 * Nose_Head_Distance) < Nose_Chin_Distance < (1.3 * Nose_Head_Distance).

✓ **<u>Rule 4:</u>** **45** < NoseAbove_Nose_NoseBottom < **150**

If the calculated face rules applied successfully to the image, this means that the image has a profile face and its coordinates will be (*x*, *y*, *Width*, *Height*).

where,

x = (Nose Point y-Index coordinate) – (*Nose_Chin_Distance*) .

y = (Nose Point x-Index coordinate) – (*Nose_Chin_Distance*) .

Width = *Nose_Chin_Distance* .

Height = 2 * (*Nose_Chin_Distance*) .

Figure 4.14(b), shows the detected face region with a red rectangle on the right profile segmented image.

a) Original Image

b) Detected right profile in segmented face

Figure 4.14: FD Algorithm output.

4.6.1 Template Matching Technique

In this work, using Template Matching approach in consistent with feature-based approach, gives us a robust FD algorithm. Feature-based approach gives an efficient result when applying it to profile still images but, it fails in non-profile (Frontal) one. Template Matching approach complete the lack of feature-based approach and it will detect the frontal face image to provide a complete FD algorithm as will be shown after finishing this section. One of the most important characteristics of this approach is that it uses a human face template to take the final decision of determining if a skin region represents a face or not.

Template Matching technique assumes that a set of reference patterns (templates) are available to us, and we have to decide which one of these reference patterns an unknown one (the test pattern) matches best. In our work it will be one template which is a face template. In previous section, we illustrated how we can detect the face from test image if it contains profile face. This section shows how to do the matching between the part of the test image corresponding to the skin region and the template face.

4.6.2 Proposed Template Matching Approach

Our proposed algorithm uses these Template Matching algorithm steps that lead to a robust algorithm as described below.

Step 1: In our work, we choose a template face by averaging 16-frontal view faces of different persons in different sex wearing no glasses and having no facial hair as shown in Figure 4.15.

Figure 4.15: Template face (model) used to verify the existence of faces in skin regions.

Notice that the left and right borders of this template are located at the center of the left and right ears of the averaged faces. The template is also vertically centered at the tip of the nose of the model. At this point, the required parameters to do the matching between the part of the image corresponding to the skin region and the template human face are all presented.

Step 2: As known from image processing principles that registering images is a main step which must be prepared before any comparison operations between images. So, first of all, the template face width and height must be in the same width and height as the face region segmented previously, and this is done by resizing the template face to match the segmented region. Figure 4.16 shows how the template face is resized to be in the same size as the face region.

a) Original image

b) Original template face **c)** Resized template face

Figure 4.16: Resized template face to be in the same size as the face region in (a).

Step 3: In order to compute the characteristics of any region, the center (i.e., centroid) of the area in any region must be calculated. It is known that the center of area in a segmented (binary) image is the same as the center of the mass which can be computed as shown below:

$$\bar{x} = 1/A \sum_{i=1}^{N} \sum_{j=1}^{M} jB[i, j] \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (4.8)$$

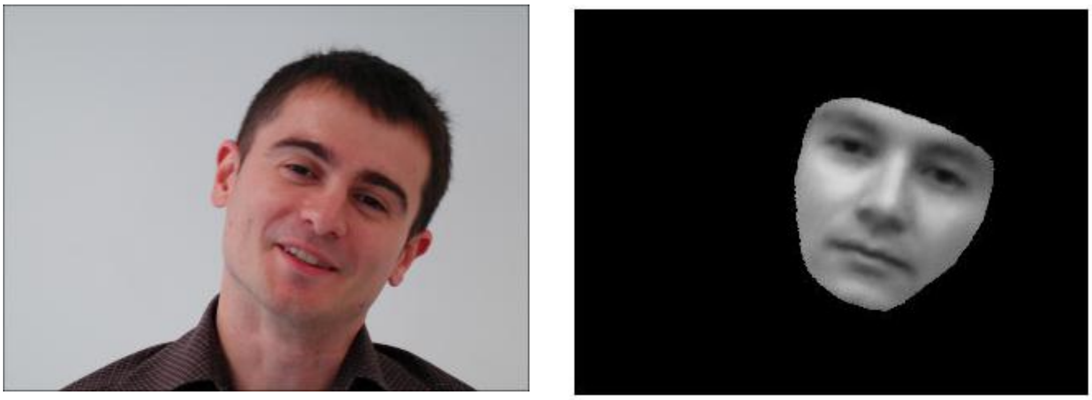
$$\bar{y} = 1/A \sum_{i=1}^{N} \sum_{j=1}^{M} iB[i, j] \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (4.9)$$

where:

B: is the matrix of size [n * m] representation of the region and

A: is the area in pixels of the region.

In our work, we compute the center of the previously rotated template face to use it for registration operation discussed in step 2. Figure 4.17, shows how we calculate the center of the face region and locate the rotated template face in the same center to prepare for the next cross-correlation operation. In step5, we show how the rotation is done.



a) Original image **b)** Locate the center of the rotated template face to the center of image.

Figure 4.17: The center of the Template face is located in the center of the skin region.

Step 4: Another point to be taken into consideration before performing the Template Matching operation is that the template face must be rotated to be in the same coordinate as the segmented image to have a higher matching. One way to determine a unique orientation is by expanding the object. The orientation of the axis of expansion will determine the orientation of the region. In this axis we will find that the inertia should be the minimum.

This axis can be computed by finding the line for which the sum of the squared distances between region points and the line is minimum. In other words, we compute the least-squares of a line to the region points in the image [31]. At the end of the process, the angle of inclination (theta) is given by:

$$\theta = {}^{1}/_{2} \ a \ \tan\frac{b}{a-c} \qquad \ldots\ldots\ldots\ldots\ldots\ldots \qquad (4.10)$$

where:

$$a = \sum_{i=1}^{N}\sum_{j=1}^{M}(x'ij)^2 B[i,j] \ldots\ldots\ldots\ldots\ldots\ldots \qquad (4.11)$$

$$b = 2\sum_{i=1}^{N}\sum_{j=1}^{M}(x'ij)^2 B[i,j] \ldots\ldots\ldots\ldots\ldots\ldots \qquad (4.12)$$

$$c = \sum_{i=1}^{N}\sum_{j=1}^{M}(y'ij)^2 B[i,j] \ldots\ldots\ldots\ldots\ldots\ldots \qquad (4.13)$$

$$x' = x - \bar{x} \ldots\ldots\ldots\ldots\ldots\ldots \qquad (4.14)$$

$$y' = y - \bar{y} \ldots\ldots\ldots\ldots\ldots\ldots \qquad (4.15)$$

$\bar{x}$ and $\bar{y}$ are the center of the mass of the region which calculated in previously in step 3.

Step 5: After determining the angle of rotation from previous step, the template face rotated by (theta) either clock-wise or counter clock-wise depending on the angle of the segmented skin region. Consequently, the template face is aligned in the same direction as the skin region is.

The point to be noticed after the rotation is that, the template face will be bigger than the original template face and this because the rotation process will add black pixels to the image and in our work these pixels will be eliminated by cropping the rotated image to the boundary of the region. Figure 4.18 shows the rotated template of a selected face image.

Step 6: Use neighboring and connected pixels to close all holes in the segmented face region generated from the segmentation and filtering steps to get only binary image which contains white pixels represents the face skin region only with no eyes or other features as shown in Figure 4.19(a). This is done for using this result face region image for masking operation.

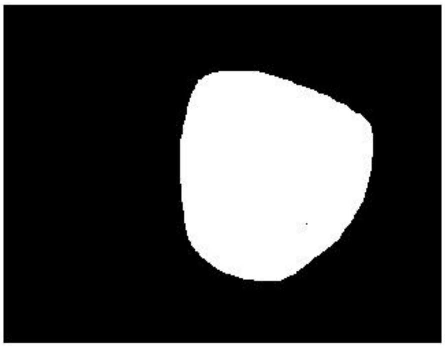
a) Original Image **b)** Rotated template face only

Figure 4.18: Rotated template face example (The black pixels is added to the image), with the center of the template face is located in the center of the skin region.

Step 7: Multiply the generated image from above step with the original (not segmented) gray scale image to obtain only the face region and mask all other parts in the test image as shown in Figure 4.19 (b).

Step 8: Do the Template Matching between the part of the image corresponding to the skin region and the template face resized and centered previously. In our work, we do Template Matching by Cross-Correlation as discussed previously in Chapter 3.

Step 9: Finally get the coordinates of the part of the image that has the template face. With these coordinates, we can draw a rectangle in the original image about the face region if found as shown in Figure 4.19(c). Figure 4.19 shows the summary of the Template Matching process discussed from step. 7 to 9.



a) Closed holes skin face image.

b) Source skin image region is multiplied with invertion of (a).

c) The final detected face

Figure 4.19: Template Matching procedure.

4.7 Used Face Recognition Technique

Assumed that the face of a person is located and segmented from the image, in this section, we talk about the most important techniques that can be used for recognizing this face and we talk about our used FR technique and how we enhance it to match our goal.

As known FR is a famous technique and many approaches are proposed in this field. In this work, we search for an approach which takes into account the obstacles mentioned previously especially the variation in lightning conditions. This is done by finding a linear projection of the faces from the high-dimensional *image space* to a significantly lower dimensional *feature space* which is insensitive to variation in lighting direction and other obstacles. So, the main task in any FR system is to reduce image space dimension.

Dimension reduction (feature extraction) can be categorized into four categories: *holistic-based* method, *feature-based* method, *template-based* method, and *part-based* method. In this work, we use some methods that related into the holistic-based category. There are two famous methods in this category that apply linear projection for dimensionality reduction which are *Eignfaces* and *Fisherfaces* and can be used in FR systems and give satisfied results. We use these two methods in an efficient way so; we can build a FR system that is robust to variation in lighting direction. In Chapter 3, some overview about how these methods are work in order to reduce the effect of high dimensionality.

4.8 Proposed Face Recognition Pre-Processing Steps

In this thesis, we try to propose robust FR algorithm so that, we propose some pre-processing steps that can pave the way for FR algorithm to work perfectly. These steps objectives are to reduce the computation complexity and minimize the effect of illumination conditions that may the detected image have. The following steps are performed to the detected, and image before presenting it to the used FR algorithm.

- **i)** <u>Down Scale Sampling</u>: A step used to down scale the cropped image by some factor which is two in our work before any projection or distance measures operation, which will reduce the computation time for projection when calculating the fisher faces.
- **ii)** <u>Color to gray scale transformation</u>: Convert the colored training set images to gray-scale images which is faster in computation operations
- **iii)** <u>DCT-Bases Normalization</u>: A technique that sets a number of DCT coefficients corresponding to low-frequencies to zero and hence tries to achieve illumination invariance.
- **iv)** <u>Contrast Stretch Normalization</u>: Stretches contrast on the image and normalize image from 0 to 1.

Chapter 5

RESULTS AND DISCUSSION

5.1 Introduction

We have implemented and evaluated our proposed FD algorithm and used FR algorithm using Matlab 7.6, which run under Delll-inspiron laptop with a 2-GHz CPU and has a 3-GB DDR2 Ram.

To test our implemented work, we carry out several tests on three different databases. The *first* is **GTAV** database [72], the *second* is **FEI** database [73], and the *third* one is **Champion** database [71]. The following section describes in details each one of these databases.

5.2 Face Databases

We carry out several tests on these three different databases which include a total of 3167 images. These databases can be worked efficiently in FR field and provide good results as shown in our work.

The *first* used database is **GTAV** database [72] which contains 1775 images of 44 different peoples under three different illuminations conditions (environment or natural light, strong light source from an angle of 45º, and finally an almost frontal mid-strong light source) and the resolutions of the images are 240x320 and they are in BMP format. It contains 260 right profile face images, 255 left profile face images, 150 frontal face images, 560 half profile face images and 550 face images with different expressions.

The *second* used database is **FEI** database [73] which contains 675 images of 100 different peoples with the original size of each image is 640x480 pixels and they are in JPEG format and all images are colorful and taken against a white homogenous background. It contains 100 right profile face images, 100 left profile face images, 125 frontal face images, 250 half profile face images and 100 face images with different expressions.

The *third* used database is **Champions** database [71], this database contains 227 images of Champions database, and each image involves one face. It contains 127 frontal face images and 100 near frontal face images..

These databases also include a lot of images with profile rotation of up to about 180^o degrees with scale might vary about 10%.

The proposed algorithms is tested on these three databases which have more than one face images with different conditions (expression, illumination,...etc.), of each individual. The database images are still images in different views (profile, half profile and frontal), of each individual.

As known that any FR system automatically includes the FD process. In this thesis we show the experimental results for FD, then we use the detected face as an input to the FR stage and also we show the experimental results.

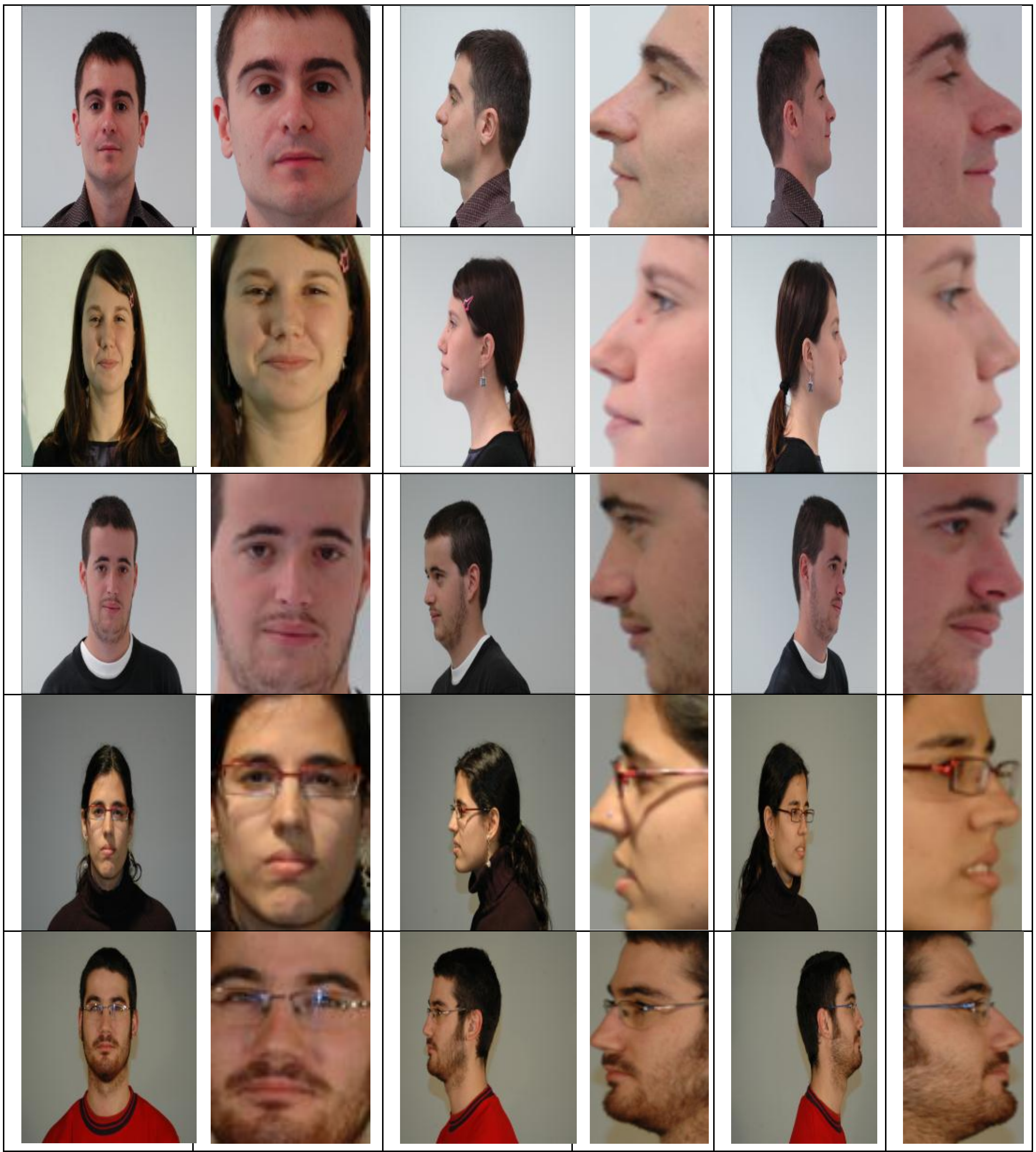
5.3 Face Detection

In this part, we use the previous databases in testing the proposed FD algorithms. So, the experiments are divided according to FD algorithms into two categories.

Firstly, experiments which deal with profile face images (right profile face images and left profile face images), and these experiments are prepared to test our new proposed algorithm that work efficiently in this category. *Secondly*, experiments which deal with non-profile face images (direct and rotated face images); and these experiments are prepared to test our used Template Matching process. The generated results are divided into three cases:

1) **Correct profile FD case**: if the face is profile and the algorithm detects it correctly.
2) **False Profile FD case**: if the face is profile but the algorithm fails to detect it.
3) **Correct non-profile FD case**: if the face is non-profile and the non-profile algorithm detects it correctly.
4) **False non-profile FD case:** if the face is non-profile and the non-profile algorithm fails to detect it correctly.

Figure 5.1, shows the successful results of applying our FD method to GTAV face database. While Figure 5.2, shows the successful results of applying our FD method to samples of FEI face database. Each one of these figures contains the three face positions which are (Right Profile, Left Profile and Frontal face images).



Original Frontal Face	Frontal Face	Original Left Face	Left Profile	Original Right Face	Right Profile

Figure 5.1: Result samples of applying proposed FD algorithm for persons in GTAV database.

5.3.1 Segmentation Process

As known, any FD algorithm contains some pre-processing steps that leads to correct FD. The most important pre-processing step in FD is *face segmentation*. The segmentation algorithm segments the face correctly from the image then this leads to accurate FD algorithm.

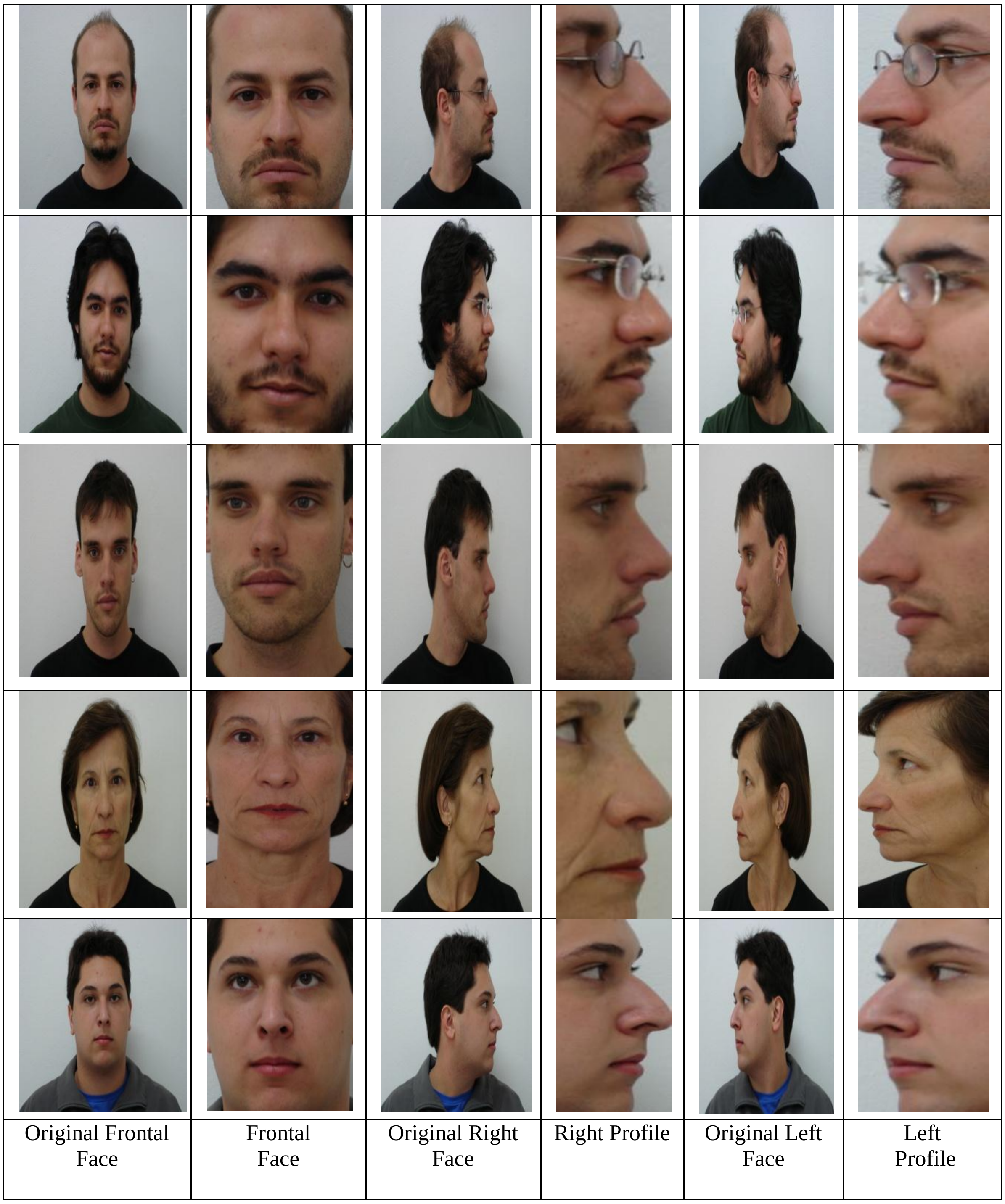


Figure 5.2: Result samples of applying proposed FD algorithm for persons in FEI database.

In this work, as shown in Chapter 4, we build an efficient face segmentation algorithm and this segmentation algorithm passes through different filtering steps. In this section, we present each filtering step and show what threshold value used in testing to retrieve this robust face segmentation algorithm. Table 5.1, presents each filtering name and the threshold value used with it. Note that these values are specified experimentally.

Table 5.1: Filtering Steps thresholds values

Filter Name / Threshold Value	Width and height of the region	Solidity Property	Horizontal thin region	Vertical thin region	Shoulder Removal
Width_Heigh_Th1	1.5				
Solidity_Threshold		0.65			
Horizontal_Threshold			0.5		
Vertical_Threshold				2.5	
Shoulder_Threshold					1.4

Figure 5.3 illustrates the output of our proposed segmentation algorithm applied to samples images from GTAV face database.

Frontal Face	Left Profile	Direct Profile

Figure 5.3: Results of applying proposed segmentation algorithm to GTAV face database.

5.3.2 Proposed Face Detection Algorithm Evaluation Results

In this section, a batch of evaluation should be made to evaluate, among others, the following parameters on different conditions:

- **Number of false Detection:** How may number of times the proposed algorithm fails in FD process.
- **Correct Detection Rate:** The ratio of the correctly detected faces.

In the following tables, detailed performance results for our selected databases when used in our proposed FD algorithms are given. These results partitioned according to the position of the face. In FEI and GTAV face database tests, we use test images with *right*, *left*, *frontal* and *half profile* face images. Also, we test images with various expressions to show the strength of the proposed algorithm. But, in Champions face database we use test images with *frontal* and *near frontal* face images because this database contains only this types of face poses.

Firstly, our proposed FD algorithm is tested using FEI face database and Table 5.2 shows the detail detection results on this database.

Table 5.2: Detail results of testing my proposed FD algorithm on FEI face database

Test Type / Test	Right profile	Left profile	Frontal	Half Profile	Different Expressions	Total
No. of images	100	100	125	250	100	675
No. of false Detection	7	5	4	9	12	37
Correct Detection Rate	93%	95%	96.8%	96.4 %	88%	94.5%

The results extracted from FEI face database shows that the proposed algorithm gives satisfied results especially when it is applied to Frontal face images. Also, the algorithm presents good performance when applying this algorithm to Left and Right profile face images. Besides that, it can be shown that it gives good performance results in different expressions face images.

Secondly, it is tested using GTAV face database and Table 5.3 shows the detail detection results on this database. The results extracted from GTAV face database shows that the proposed algorithm gives more satisfied results especially when it is applied to Frontal face images. Also, the algorithm presents good results for applying this algorithm to Left and Right profile face images. Besides that, it can be shown also that it gives good performance in different expressions face images.

Table 5.3: Detail results of testing my proposed FD algorithm on GTAV face database

Test Type / Test	Right profile	Left profile	Frontal	Half Profile	Different Expressions	Total
No. of images	260	255	150	560	550	1775
No. of false Detection	13	12	10	25	28	88
Correct Detection Rate	95%	95.2%	99.3%	95.5%	94.9%	95.04%

Thirdly, it is tested using Champions face database and Table 5.4 shows the detail detection results on this database. In this test the database images are divided into frontal and near-frontal face images, and these results indicate that the algorithm works efficiently.

Table 5.4: Detail results of testing my proposed FD algorithm on Champions face database

Test Type / Test	Frontal	Near-Frontal	Total
No. of images	127	100	227
No. of false Detection	6	9	15
Correct Detection Rate	95.2 %	91%	93.3%

☒ Discussion and Comparison

Our proposed algorithm gives efficient results compared to other approaches. To see how efficient our proposed algorithm work, our results are compared with other techniques work in this field. We choose the work proposed by Hsu et.al [62] to compare with. There are many other researches which we present some of them in related work chapter that can be used for comparison in this field but, we search for research that is close to the idea of our thesis idea. Hsu et.al. propose an algorithm for FD that is similar to our algorithm because it detect the face in complex background and with varying lightening conditions. So, we select Hsu et.al. research for our comparison. In this comparison, we evaluate the average time consuming and the correct FD percentage parameters. The obtained results are presented in Table 5.5.

Table 5.5: Results of comparing my proposed algorithm with other algorithm

Algorithms / Test	**Proposed Algorithm**	**Hsu. Algorithm**
Correct Percentage %	94.5%	91.63 %
Average Time Execution	5 sec.	-

It is shown from Table 5.5 that our algorithm gives better performance than Hsu et.al [62] nearly by 2.87 %, which is an efficient result in this filed.

5.3.3 Case Of Failure

However the proposed FD algorithm gives a satisfied results in most cases but, there are still some error factors can appears due to different reasons. Tables 5.2-4, shows some of these cases and we present some conditions which lead to failure in FD process.

- **Case 1: Segmentation Error**

Inspite of the robustness of our proposed face segmentation algorithm but, in some cases this algorithm faces some obstacles that lead to its failure. Figure 5.4 shows example of failure happened when the person wearing clothes which are very relevant to skin color hence, the proposed algorithm mark these clothes as a skin pixels and this will lead to fail in next FD step.

In other cases, the algorithm overstep this error and the algorithm can continue working and giving results in FD step but, it detects the clothes as part from face region as shown in Figure 5.5.

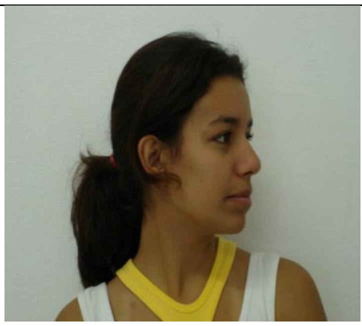
		**Fail !**
a) Original Image	b) Segmented Image	c) Detected Face Image

Figure 5.4: Clothes with similar skin color lead to face segmentation error case.

a) Original Image | b) Segmented Image | c) Detected Face Image

Figure 5.5: Detected face region with false face segmentation error case.

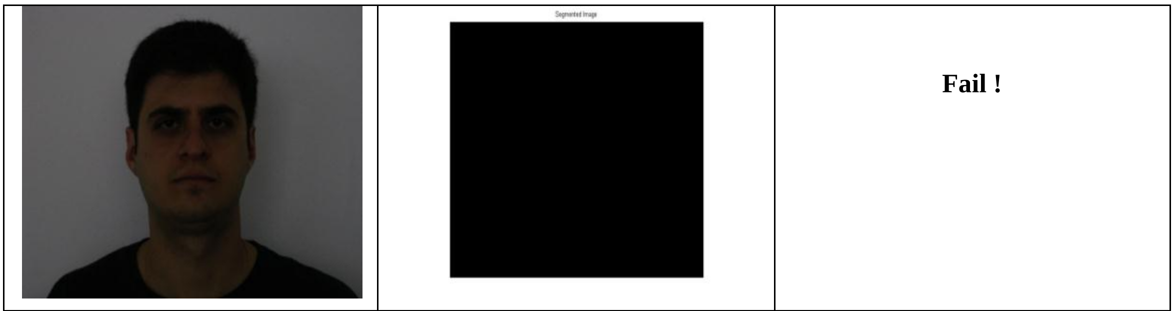
In some case, the long hair with the similar skin color leads to fail in FD process. Figure 5.6, shows an example of this failure.

a) Original Image | b) Segmented Image | c) Detected Face Image

Figure 5.6: Long hair with similar skin color lead to face segmentation error case.

- **Case 2: Noisy Image**

The noise in the image is one of the obstacles that faces the FD algorithm. Noise may occur in digital images because of many reasons such as camera lens, environment conditions, lightening sources,…etc. Figure 5.7, shows an image with some noise which leads to fail in segmentation process.



a) Original Image | b) Segmented Image | c) Detected Face Image

Figure 5.7: Noise in the image.

- **Case 3: Irregular Illumination Condition**

The illumination conditions are important factors that affect the FD process. As shown in Figure 5.8, the lightening source is focused in a direction with some shadow that leads to miss-classification in our proposed algorithm.

		**Fail !**
a) Original Image	b) Segmented Image	c) Detected Face Image

Figure 5.8: Irregular illumination conditions.

5.4 Face Recognition

Now, we are going to see some results obtained from testing and evaluating FR algorithm on two different face databases which are GTAV and FEI for showing the algorithm efficiency.

We divide our experiments according to the obtained images from FD algorithms and to the images into databases into two categories as follows:

- *Firstly*, experiments which deal with profile face images (Right profile face and Left profile face).
- *Secondly*, experiments which deal with non-profile face images (Frontal and rotated face).

As in any FR system, we use two datasets in testing process:

1) **Train dataset.**
2) **Test dataset.**

5.4.1 Face Conditions

In this work, we use different selected images from two selected databases which are GTAV and FEI as training sets. These sets contain images that obeyed to different type of conditions which can affect the FR, and these conditions are as follows:

1) The training datasets contain persons from different races and this leads to images with different colors from one person to another.
2) The training datasets contain images with different direction of lighting, which could vary for the images of the same person.
3) The images for the same person include different variation of facial expressions and angle.

4) In addition to that, the images backgrounds also vary, introducing differences between single images of the same person and introducing a more scatter within the same class (person).

5.4.2 Training

In our work, we used two different training sets for two different face databases which are:

1) *GTAV Face Database.*
2) *FEI Face Database.*

In the following topics, we show how we use these training sets for testing the FR algorithm.

i. GTAV Face Database (Training Set)

We build a training set from previously cropped GTAV face database as an input to FR algorithm. This training set consists of 140 images, containing 28 persons (Classes), and five images per class and the images in each class describe the person with different poses under different lightening conditions. Figure 5.9, shows samples from this training set.

These class images are processed before training it to reduce the computation complexity and the results after applying these pre-processing steps discussed previously in Chapter 4 to the GTAV training set are shown in Figure 5.10.

Figure 5.9: GTAV training set for selected class.

Figure 5.10: GTAV pre-processed training set for selected trained class in Figure 5.9.

ii. FEI Face Database (Training Set)

We build another training set from previously cropped FEI face database as input to FR algorithm. This training set consists of 115 images, containing 23 persons (Classes), and 5 images per class. Figure 5.11, shows samples from this training set. These class images are processed before training to reduce the computation complexity and the results after applying these pre-processing steps discussed previously in Chapter 4 to the FEI training set are shown in Figure 5.12.

Figure 5.11: FEI training set for selected class.

Figure 5.12: FEI pre-processed training set for selected trained class in Figure 5.11.

5.4.3 Testing

Now, after building a training set from two databases, we select a test set images from the these two face databases to form a test. Again these test images taken from the following databases which are:

1) GTAV Face Database (Test set).
2) FEI face Database (Test set).

These two test sets used in testing the FR algorithm. We divide the experiments according to our FD algorithms output into two classes as follows:

- *Firstly*, Test profile faces images under the different conditions.

A) Right profile face.

B) Left profile face.

- *Secondly*, Test non-profile face images under different conditions.

A) Frontal Faces.

B) Half Profile Faces.

C) Different Expression faces.

☒ Discussion

As we stated previously, we used Fisher algorithm for FR and the work strategy in our work is as follows:

i) Any test image from any test set is projected onto fisher faces where fisher faces count is:

Fisher faces count = Number of classes (Persons) in training set – 1

ii) The results weights after projection are then stored as a features. The distances between the test image weight and the training image weights are calculated and the closest training image is chosen.

iii) The close measure must supported with some threshold so that:

- If the closest distance is above the threshold then, the test face is considered unrecognized.
- If the closest distance is below, then match the test face with the identity of the closest face.

We run the FR algorithm in two test sets and the returned results can be divided into three cases:

- ✓ **False rejection (FRej)**: if the face is associated with the correct face, but the algorithm reject it.
- ✓ **False acceptance (Facc.):** if the face is associated with the wrong face, but the algorithm accept it.
- ✓ **Correct rejection (Crej.):** if the face is associated with the wrong face, and the algorithm reject it.
- ✓ **Correct recognition(Crec.):** if the face is associated with the correct face, and the algorithm correctly recognize it.

The results of the two test sets used in our testing is shown in the following tables. In Table 5.6, we show the result of the testing set taken from GTAV Face Database. It is shown that the correctly detected faces ratio equal 95%, while the algorithm gives error rate ratio equal 4.8 %. The results show that the false acceptance is very low which is equal 0.2 %. This value strength our algorithm because the false acceptance is an important value in testing any system. This appears in practical application since it is preferred that the algorithm inform that it can't detect the image rather than it accept it falsely.

Table 5.6: Detail results of testing FR algorithm on GTAV face database.

Number of test faces	Frontal faces = 150 Half profile = 560 Left faces = 255 Right Faces = 260 Different poses with different expressions = 550, **Total = 1775**				
Fisherface Algorithm					
	Frej.	**Facc.**	**Crej.**	**Crec.**	**Total Correct**
	82	5	69	1620	1689
Error Ratio %	4.6%	0.2 %	-	-	4.8 %
Correct Ratio %					95 %

While, in Table 5.7 we show the results of our test set which taken from FEI Face Database. It is shown that the ration of correctly detected faces ratio is nearly 95.2%, with error rate ratio equal 4.6%. The results show that the false acceptance is low which is equal 0.5 %. This value strength our algorithm because the false acceptance is an important value in testing any system. This appears in practical application since it is preferred that the algorithm inform that it can't detect the image rather than it accept it falsely.

Table 5.7: Detail results of testing FR algorithm on FEI face database.

Number of test faces	Frontal faces = 125 Half profile = 250 Left faces = 100 Right Faces = 100 Different poses with different expressions = 100, **Total = 675**				
Fisherface Algorithm					
	FRej.	**FAcc.**	**CRej.**	**CRec.**	**Total Correct**
	28	4	43	600	643
Error Ratio %	4.1%	0.5%	-	-	4.6 %
Correct Ratio %					95.2 %

☒ Face Recognition Computation Time

This part shows the run time of FR operation when testing an input image according to the steps stated previously. In this test we partition the run time into *training* run time and *testing* run time. Table 5.8 and Table 5.9 present the detailed results when applying the FR algorithm to GTAV and FEI face databases.

i) <u>Training Phase:</u>

Table 5.8, shows the training time for two different selected databases that contains different images with different poses, illumination conditions and facial expressions.

Table 5.8: Training run time for FR algorithm on two different selected face databases.

Database Name	Time (sec.)
GTAV	7.5
FEI	7.2

ii) <u>Testing Phase:</u>

Table 5.9, shows the detailed run time when a new query is presented. In this table, we show each FR algorithm step in details and present the time it needs to complete its assigned job.

Table 5.9: Run time of FR algorithm when testing an input image.

Algorithm Step	Time (sec.)
Pre-Processing	0.05
Projection operation	0.38
Distance computation & providing Recognition result	0.52
Total	**0.95**

Chapter 6

CONCLUSION

6.1 Summary and Conclusion Remarks

This thesis has shown, through implementation and testing, the importance of FR approach. FR is both challenging and important recognition technique. Among all the biometric techniques, FR approach possesses one great advantage, which is its user-friendliness (or non-intrusiveness).

The main step that precedes any FR system is a FD, this because an accurate FD system will pave the way for robust FR system. The goal of FD step is to determine whether or not there are any faces in the image and, if present, locate the image face. Many obstacles face the FD process such as face orientation, face size, different facial expressions, different facial feature, occlusion, different lightning conditions,…etc. This research presents a robust working in still images that have different viewing conditions (profile faces and near profile faces) and different surrounded conditions such as complex background, various illumination conditions, different people's distractions (eye glasses) and different people races to reach the robust FR result.

In any FD problem use feature-based approach for locating face firstly, it will start with low level feature analysis which dealt with segmentation of visual features. Features generated from this are likely to be ambiguous and other high level feature analysis should be done for converting these features to knowledge. Throughout this thesis, we have applied a new feature-based technique. We use the low level feature analysis and introduce a new high level technique in a different way that not found in other high level techniques such as Template Matching.

Because different viewing conditions are one of the major obstacles that face any FD system, so that we partition the work to deal with different types of still images (profile, near-profile and frontal). We propose a new algorithm to detect and locate face in right and left profile images while, we use Template Matching approach to locate face in frontal images.

Proposed FD algorithms are tested on three famous face databases that contain peoples from different races under different illumination conditions which are GTAV, FEI and Champions databases to show its efficiency in detecting and locating the faces. The proposed algorithms are implemented using Matlab version 7.6 software and they are evaluated using Delll-inspiron laptop run with a 2-GHz CPU and has a 3-GB DDR2 RAM. The performance of proposed algorithm ranges from 93.3% in Champions face database to 95.04% in GTAV face database. Also, the results are compared with other researchers [62] proposed techniques, and we obtain satisfactory on these results that reach very good degree.

After the FD step, human-face patches are extracted from images. Then FR step now can be applied. Different holistic approaches are used in this field, we used fisherface technique in this work which performed by the LDA. Fisherface algorithm derived from FLD that maximizes the between-class variance as well as minimizes the within-class variance. We propose some pre-processing steps that are applied before recognition using fisherface algorithm and these steps will help for normalizing the variance in the illumination in the images and will reduce the computation complexity when applying fisherface algorithms.

Face Regognition using fisherfaces is tested on two famous face databases used in this field which are GTAV, FEI to present the efficiency of using this algorithm on recognizing the detected faces from the FD step.

In summary, the comparison that has been done among our work and other methods shows the enhancements achieved over the other systems. With this research we have attained a robust system that does not require any sort of training and yet can detect faces in images very accurately and efficiently.

6.2 Recommendations and Future Work

The main goal of this thesis is to introduce the field of FR and implement a FD algorithm that will lead to robust FR system that works efficiently with faces in different poses that fall under various illumination conditions. An overview of the various approaches to the problem is presented. From survey we conclude that all FR algorithms can be categorized into two main approaches which are feature-based approach and image-based approach. Because of the weakness of Image-based approach because of needing a large number of training that has to be performed on the system for faces and non faces. For this reason we tried to make a better system using feature-based approach that has all the advantages of feature-based machine and also is very accurate.

The goal of any future work falls into improving three factors which are:

1) FD rate.
2) Minimizing the number of false positives.
3) The speed of the FD process.

A good place to start would be to aim to minimize the number of false positives. Improving the set of non-face examples would be extremely beneficial, as this is a major weakness in the current system.

FD and FR are still open research and we recommend for solving the following unsolved problems in these fields:

1) Noise in the face.
2) FR in public area (camera in airport, street.. etc.)
3) Small image (like FR in mobile image).
4) Develop robust techniques to be used in unconstrained real-world applications.

Finally it is recommended to take into consideration the previous three factors because they are related to each other and improving one of them will affect the others.

REFERENCES

[1] W. Zhao, R. Chellappa, P. J. Phillips and A. Rosenfeld, "Face Recognition: A Literature Survey," ACM Computing Surveys, Vol. 35, No. 4, pp. 399–458 , December 2003.

[2] R. Jafri and H.R.Arabnia, "A Survey of Face Recognition Techniques," J. of Information Processing Systems, Vol. 5, No.2, June 2009.

[3] R. Chellappa, C.L. Wilson, and S. Sirohey, "Human and machine recognition of faces: A survey," in Proc. IEEE, vol. 83, pp. 705–740, 1995.

[4] H. Wechsler, P. Phillips, V. Bruce, F. Soulie, and T. Huang, "Face Recognition: From Theory to Applications," eds. Springer-Verlag, Berlin, Germany, pp. 578-588, 1996.

[5] C. Zhang and Z. Zhang, "A Survey of Recent Advances in Face Detection," Microsoft Corporation, Redmond, WA USA , Technical Rep. MSR-TR-2010-66, June 2010.

[6] F. Perronnin and J.-L. Dugelay, "An introduction to biometrics and face recognition," In: Proc. Of IMAGE 2003: Learning, Understanding, Information Retrieval, Medical, Cagliari, Italy, pp 1-20, June 2003.

[7] M.-H. Yang, D. J. Kriegman, and N. Ahuja, "Detecting faces in images: A survey," IEEE Trans. Pattern Analysis and machine intelligence, Vol. 24, No. 1, pp. 34–58, January 2002.

[8] T. Sakai, M. Nagao, and T. Kanade, "Computer analysis and classification of photographs of human faces," presented at Proc. of 1st USA-Japan Computer Conf., Kyoto, Japan, pp. 55-62, 1972.

[9] E. Hjelmås and B.K. Low, "Face Detection: A Survey", IEEE Trans. Computer Vision and Image Understanding, vol. 83, no. 3, pp.236-274, Sept. 2001.

[10] C. Aiping, P. Lian, T. Yaobin, and N. Ning, "Face Detection Technology Based on Skin Color Segmentation and Template Matching," In Education Technology and Computer Science (ETCS) , 2nd International Workshop, Wuhan, 2010, vol. 2, pp.708-711 .

[11] Y. Gao and M.K.H. Leung, "Face recognition using line edge map," in IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 24, No. 6 , pp. 764 -779, June 2002.

[12] H. Martin Hunke, "Locating and tracking of human faces with neural network," M.S. thesis, University of Karlsruhe, 1994.

[13] Henry A. Rowley, Shumeet Baluja, and Takeo Kanade, "Neural network based face detection," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol.20, No.1, pp.23-38, 1998.

[14] S. Kr. Singh1, D. S. Chauhan, M. Vatsa, R. Singh, "A Robust Skin Color Based Face Detection Algorithm," in Tamkang J. of Science and Engineering, Vol. 6, No. 4, pp. 227-234, 2003.

[15] J. Yang and A. Waibel, "A real-time face tracker," in IEEE Proc. of the 3rd Workshop on Applications of Computer Vision, Pittsburgh, PA,USA, 1996.

[16] V. Govindaraju, D. B. Sher, R. K. Srihari, and S. N. Srihari, "Locating human faces in newspaper photographs," presented at Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, 1989, p.p 549-554 .

[17] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," presented at IEEE Proc. of Computer Vision and Pattern Recognition, pp. 586-591, June 1991.

[18] M. Turk and A. Pentland, "Eigenfaces for recognition," J. of Cognitive Neuroscience, Vol. 3, No.1, pp. 71–86, 1991.

[19] H. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol.20, No.1, pp.23-38, Jan. ,1998.

[20] E. Osuna, R. Freund, and F. Girosi, "Training support vector machines: An application to face detection," Presented at Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, 1997, pp. 130-136.

[21] H. Schneiderman and T. Kanade, "Probabilistic modeling of local appearance and spatial relationships for object recognition," presented at Proc. of IEEE Conference on Computer Vision and Pattern Recognition, 1998, pp. 45-51.

[22] A. Rajagopalan, K. Kumar, J. Karlekar, R. Manivasakan, M. Patil, U. Desai, P. Poonacha, and S. Chaudhuri, "Finding faces in photographs," presented Proc. of the Sixth IEEE International Conf. on Computer Vision, 1998, pp. 640-645.

[23] A.F. Abate, M. Nappi, D. Riccio, and G. Sabatino, "2D and 3D face recognition: A survey," presented at Pattern Recognition Letters, 2007, pp.1885-1906.

[24] K. Kim, "Intelligent Immigration Control System by Using Passport Recognition and Face Verification," in International Symp. on Neural Networks, Chongqing, China, 2005, pp.147-156.

[25] H. Moon, "Biometrics Person Authentication Using Projection-Based Face Recognition System in Verification Scenario," in International Conf. on Bioinformatics and its Applications, Hong Kong, China, 2004, pp.207-213.

[26] P. J. Phillips, H. Moon, P. J. Rauss, and S. A. Rizvi, "The FERET Evaluation Methodology for Face Recognition Algorithms," IEEE Trans. Pattern Analysis and Machine Intelligence, Vol.22, pp.1090-1104, 2000.

[27] L. Sirovich and M. Kirby, "Low-dimensional Procedure for the Characterization of Human Faces," J. of the Optical Society of America A: Optics, Image Science, and Vision, Vol.4, No.3, pp. 519-524, 1987.

[28] K. Fukunaga, "Introduction to Statistical Pattern Recognition," 2nd ed. Boston, MA: Academic Press, 1990.

[29] R. Cendrillon and B. C. Lowell, "Real-Time Face Recognition using Eigenfaces," in Proc. of the SPIE International Conference on Visual Communications and Image Processing, Vol.4067, 2000, pp.269-276.

[30] J. Song, Z. Chi, and J. Liu, "A robust eye detection method using combined binary edge and intensity information", presented at IEEE Trans. of Pattern Recognition, Vol. 39, No. 6, pp.1110-1125, 2006 .

[31] R. Ramesh, Kasturi R. and B. G. Schunck, "Machine Vision", pp. 31 - 51, McGraw Hill, New York 1995.

[32] A. Samal and P.A. Iyengar, "Automatic recognition and analysis of human faces and facial expressions: a survey", presented at IEEE Trans. Pattern Recognition, vol. 25, pp. 65-77, 1992.

[33] P. Belhumeur, J. Hespanha, and D. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection," IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 19, No. 7, 1997.

[34] R. Duda and P. Hart, "Pattern Classification and Scene Analysis," New York: Wiley, 1973.

[35] L. Chen, H .Liao, M. Ko, J. Lin, G. Yu, "A new LDA-based face recognition system which can solve the small sample size problem," Pattern Recognition, Vol. 33, pp.1713–1726, Oct. 2000.

[36] I. K. Fodor. "A survey of dimension reduction techniques", Lawrence Livermore National Laboratory, Livermore, Technical Report UCRL-ID-148494, 2002.

[37] R.A. Fisher, “The Use of Multiple Measures in Taxonomic Problems,” Ann. Eugenics, vol. 7, pp. 179-188, 1936.

[38] R. Feraud and O. J. Bernier," A Fast and Accurate Face Detection Based on Neural Network, " IEEE Trans. Pattern Analysis and Machine Intelligence, vol.23, no. 1 , pp. 42-53, Jan. 2001 .

[39] D.Maio and D.Maltoni, "Real-time Face Location on Gray-scale Static Images," IEEE Trans. Pattern Recognition, vol. 33, no. 9 , pp. 1525 - 1539, Sept. 2000 .

[40] M. Pantic and L. J. M. Rothkrantz, "Automatic Analysis of Facial Expressions: TheState of the Art," IEEE Trans. Pattern Analysis and Machine Intelligence, vol.22, no.12, pp. 1424-1445 , Dec. 1999.

[41] W. Chao, "Face Recognition," Under-Graduate research, The Graduate Institute of Communication Engineering (GICE), National Taiwan University,2010 .

[42] I. T. Jolliffe, "Principal component analysis," New York, Springer-Verlag, 1986.

[43] H. A. Rowley, S. Baluja, and T. Kanade," Neural Network-Based Face Detection," IEEE Trans. Pattern Analysis and Machine Intelligence, vol.20 , pp.23-38, Jan. 1998.

[44] H. A. Rowley, S. Balu ja, and T. Kanade," Rotation Invariant Neural Network-Based Face Detection," In Proc. IEEE Conf. Computer Vision and Pattern Recognition, 1998, pp. 38-44.

[45] K. K. Sung and T. Poggio, "Example-Based Learning for View-Based Human Face Detection," IEEE Trans. Pattern Recognition and Machine Intelligence, vol.20, pp. 39-51 ,Jan. 1998.

[46] S. Lawrence, C. Giles, A. Tsoi, and A.D. Back, “Face recognition: A convolutional neural-network approach,” IEEE Trans. Neural Networks, vol. 8, pp. 98-113, 1997.

[47] K Yow and R. Cipolla, "Feature-based Human Face Detection, " IEEE Trans. Image and Vision Computing, vol. 15 , no. 9 , pp. 713-735 ,Sept., 1997 .

[48] M. Lew and N. Huijsmans, "Information Theory and Face Detection," presented at Proc. Int'l Conf. Pattern Recognition,1996, pp.601-60 5 ,.

[49] A. Colmenarezand T. Huang, "Face detection with information based maximum discrimination," presented at IEEE Int'l Conf. Computer Vision and Pattern Recognition, 1997, pp.782-787 .

[50] S.Y. Kung and J.S. Taur, "Decision-Based neural networks with signal/image classification applications," IEEE Trans. Neural Networks, vol. 6, pp. 170-181, 1995.

[51] S. Lin, S. Kung, and L. Lin, "Face recognition/detection by probabilistic decision-based neural network," IEEE Trans. Neural Networks, vol. 8, pp. 114-132, 1997.

[52] N. Oliver, A. Pentland, and F. B´erard, "LAFTER: A real-time face and lips tracker with facial expression recognition," in Elsevier J. of Pattern Recognition, Vol.33, pp.1369–1382, 2000.

[25] T. Jebara, "3D Pose Estimation and Normalization for Face Recognition,", M.S. Thesis, Center for Intelligent Machines, McGill University, Canada, May, 1996.

[54] I. Craw, D. Tock, and A. Bennett, "Finding face features," in Proc. 2nd European Conf. on Computer Vision, 1992, pp.92-96.

[55] M. A. Grudin, "On internal representations in face recognition systems," Elsevier J. of Pattern Recognition, Vol.33, No.7, pp.1161-1177, 2000.

[56] B. Heisele, P. Ho, J. Wu, and T. Poggio, "Face recognition: component-based versus global approaches," in ACM J. of Computer Vision and Image Understanding, Vol.91, No.1, pp.6-21, July, 2003.

[57] A. Yuille, D. Cohen, and P. Hallinan, "Feature extraction from faces using deformable templates," in IEEE Computer Society Conf. on Computer Vision and Templates, San Diego, CA, USA, 1989, pp.104-109.

[58] M. Nixon, "Eye spacing measurement for facial recognition," in Proc. SPIE Applications of Digital Image Processing VIII, England, 1985, pp.279-285.

[59] R.A. Fisher, "The Statistical Utilization of Multiple Measurements", In Annals of Eugenics, vol. 8, pp. 376-386, 1938.

[60] A. Martinez and A. Kak, "PCA versus LDA", IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 23, No. 2, pp. 228-233 ,Feb. , 2001,

[61] J. Weng, J.S. Huang, and N. Ahuja, “Learning recognition and segmentation of 3D objects from 2D images,” presented at Proc. IEEE Int'l Conf. Computer Vision, 1993, pp. 121-128.

[62] Rein-Lien Hsu, Mohamed Abdel-Mottaleb, and Anil K. Jain, "Face Detection In Color Images," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 24, No. 5, May, 2002 .

[63] M. Kirby and L. Sirovich, "Application of the Karhunen-Loeve Procedure for the Characterization of Human Faces," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 12, No. 1, pp. 103-108, 1990.

[64] J. Lu, K.N. Plataniotis and A.N. Venetsanopoulos, "Face Recognition Using LDA-Based Algorithms," IEEE Trans. on Neural Networks, Vol. 14, No. 1, pp. 195-200, Jan. , 2003.

[65] H. P. Graf, T. Chen, E. Petajan, and E. Cosatto, "Locating faces and facial parts," in 1st International Workshop on Automatic Face- and Gesture-Recognition, 1995, pp.41-46.

[66] T. Hastie, R. Tibshirani, and J. Friedman, "The Elements of Statistical Learning," 2nd ed., Springer, 2005.

[67] M. Jones, "Face Recognition: Where we are and where to go from here", In IEEE Trans. on Electronic, Information and Systems, Vol. 129 (2009), No. 5, pp. 770-777, January 2009.

[68] L. Mostafa and S. Abdelazeem , "Face Detection Based on Skin Color Using Neural Networks," In J. Of Graphics, Vision and Image Processing, Cairo, Egypt, pp. 9-15, 2007 .

[69] H. Rowley, S. Baluja and T. Kanade, "Neural network-based face detection", In IEEE Trans on PAMI, vol. 20, no. 1, pp. 23-38, 1998.

[70] C. Iancu, " Face Recognition using Statistical Methods," Ph.D. dissertation, Dept. of Electronic Eng., National University of Ireland, Galway, Ireland, September 2010.

[71] "The Champion Database," (2001,Mar.), Available: http://www.libfind.unl.edu/ .

[72] F. Tarrés, A. Rama, "GTAV Face Database," available: http://gps-tsc.upc.es/GTAV/ResearchAreas/UPCFaceDatabase/GTAVFaceDatabase.htm.

[73] "FEI face Database," (Mar. 2006), available: http://fei.edu.br/~cet/ facedatabase.html.

[74] T. Kanade , "Picture Processing by Computer Complex and Recognition of Human Faces", PhD thesis, Kyoto University, 1973.

[75] R. Brunelli and T. Poggio, "Face recognition: Features versus templates", IEEE Trans. On Pattern Analysis and Machine Intelligence, Vol.15, No.10, pp.1042–1052, 1993 .

[76] J. Cox, J. Ghosn and N. Yianilos , "Feature-based face recognition using mixture-distance, Technical report, NEC Research Institute, Princeton, NJ, 1995.

[77] D. DeMers and G. Cottrell, "Non-linear dimensionality reduction," in S. Hanson, J. Cowan and C. L. Giles, eds, ‘Advances in Neural Information Processing Systems 5’, Morgan Kaufmann Publishers, San Mateo, CA, pp. 580–587, 1993.

[78] F. Samaria and F. Fallside, "Face identification and feature extraction using hidden markov models," Image Processing: Theory and Application, G. Vernazza, ed., Elsevier, 1993.

[79] K. Chang, K. Bowyer and S. Sarkar, "Comarison and combination of ear and face images in appearance-based biometrics," IEEE Trans. On Pattern analysis and machine intelligence, vol. 25, No. 9,pp. 1160-1165, Sep. 2003.

[80] T. Stonham,"Practical face recognition and verification with WISARD," J. of Aspects of Face Processing, pp. 426-441, 1984.

[81] K. Sung and T. Poggio, "Learning human face detection in cluttered scenes," IEEE Trans. On Computer Analysis of Image and patterns, pp. 432-439, 1995.

[82] A. Lanitis, C.J. Taylor, T.F. Cootes, "Automatic interpretation and coding of face images using flexible models," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 19, No.7, pp. 743-756, 1997.

[83] F. Y. Shih and C. F. Chuang, "Automatic extraction of head and face boundaries and facial features," Inf. Sci. Vol. 158, No.1, pp.117–130, Jan. 2004.

[84] J. Wu and Z. Zhou, "Efficient face candidates selector for face detection," IEEE Trans. on Pattern Recognition, Vol.36, No.5, pp.1175–1186, 2003.