A

Dissertation

On

# An Efficient Biogeography Based Face Recognition Algorithm

Submitted in partial Fulfillment of the requirement

For the award of the Degree of

**Master of Technology**

**In**

**Computer Science & Engineering**

Submitted By

**Abhishek**

**University Roll No. 2K11/CSE/01**

Under the esteemed guidance of

**Dr. Daya Gupta**

**HOD, Computer Engineering Department, DTU, Delhi**

**DELHI TECHNOLOGICAL UNIVERSITY**

**2011-2013**

**DELHI TECHNOLOGICAL UNIVERSITY**

**DELHI - 110042**

# CERTIFICATE

This is to certify that the dissertation titled "**An Efficient Biogeography Based Face Recognition Algorithm**" is a bonafide record of work done at **Delhi Technological University** by **Abhishek, Roll No. 2K11/CSE/01** for partial fulfilment of the requirements for degree of Master of Technology in Computer Science & Engineering. This project was carried out under my supervision and has not been submitted elsewhere, either in part or full, for the award of any other degree or diploma to the best of our knowledge and belief.

Date: _____________

**(Dr. Daya Gupta)**

**HOD & Project Guide**

**Department of Computer Engineering**

**Delhi Technological University**

# ACKNOWLEDGEMENT

I would like to express our deepest gratitude to all the people who have supported and encouraged me during the course of this project without which, this work could not have been accomplished.

First of all, I am very grateful to my project supervisor Dr. Daya Gupta for providing the opportunity of carrying out this project under her guidance. I am deeply indebted to her for the support, advice and encouragement she provided without which the project could not have proceeded smoothly.

I am highly thankful to Ms. Lavika Goel, Research scholar in Department of Computer Engineering. who enlightened me at every step of this project by giving helpful directions and guidance. I am grateful to all my friends and family for their continued support and encouragement throughout the research work.

**Abhishek**

**University Roll no: 2K11/CSE/01**

**M.Tech (Computer Science & Engineering)**

**Department of Computer Engineering**

**Delhi Technological University**

**Delhi - 110042**

# Abstract

Extracting the optimal features from images is always required in face recognition algorithms to achieve high accuracy. In this thesis we have presented an efficient face recognition algorithm based on Biogeography Based Optimization (BBO).

We use Principal Component Analysis (PCA) as the face recognition technique to extract the most desirable features from the image. This is because an image doesn't require all the data present in it for its representation. These extracted features are minimal required features to represent an image.

First we apply Gabor filters to the correct the images to be given as input to PCA. Gabor filters help in proper alignment of images. Then we extract the features present in the images with the help of PCA. Finally we apply BBO to select the most desirable features out of the features already selected by PCA. The proposed biogeography based face recognition algorithm is applied to search the most desirable features based on well defined fitness function.

Performance analysis is performed using Olivetti research Laboratory (ORL) face database. Performance results show that biogeography based face recognition algorithm generates better results than original PCA technique with gabor filters.

# TABLE OF CONTENTS

# List Of Figures

# List Of Tables

# Chapter 1
# Introduction

There is a lot of research avenues in the field of face recognition due to challenges present in the field. The goal of face recognition is to match a given image against a large database of images to check its presence. The face recognition has been applied to two most important applications i.e verification (one to one matching) and identification (one to many matching) [9]. Face Recognition has applications in security systems all around the world. Converting the input face images into a collection of features is known as feature extraction. Feature extraction from the given images is one of the prime tasks of face recognition. Researchers have presented a lot of techniques for face recognition [1]. In our work we have used Principal Component Analysis (PCA) as the face recognition technique.

PCA is a standard technique for extracting features from the images. One of the biggest challenges in PCA is that we cannot use it on raw images directly. They need to be properly aligned and uniformly illuminated. This challenge can be solved using gabor filters which takes a raw image, generates gabor filter response and convert the raw image into properly aligned and constantly illuminated image. We have explored the research that showed that Gabor PCA based method for face recognition outperforms PCA based Eigenface method [3]. To improve further this hybrid technique we applied Biogeography Based Optimization (BBO) to further optimize the desirable features. First we extract the features using the principal Component Analysis (PCA) after applying gabor filters and then we apply BBO to get the most desirable features. The proposed biogeography based face recognition algorithm is applied to search the most desirable features.

## 1.1 Motivation

Face Recognition has been one of the most challenging research area in the past decades. There is a growing need of security system that can perform automatic face recognition and can identify or verify an individual. During the past decades face recognition technology has grown significantly [17, 18]. So far, researchers have explored many techniques for face recognition [1]. PCA is a standard technique which is used to extract the features from a given image. Although there are many techniques present for feature extraction but PCA is one of the common techniques.

The basic idea of PCA is to convert the large number of features present in an image into lower dimensional feature vectors. PCA has some of its own challenges and we have tried to solve those challenges by using Gabor filters.

Biogeography Based Optimization (BBO) is a new optimization technique inspired from biogeography. Biogeography is the study of geographical distribution of biological organisms. It aims to learn how species migrate from one island to another. BBO is quite similar to the genetic algorithm and particle swarm optimization (PSO) with few differences. BBO was first presented by D. Simon in 2008 [1]. BBO has been applied to problems of different domains [13, 14, 15, 16]. BBO has also been applied to Discrete Cosine Transform (DCT) in the field of face recognition [10].

In BBO we share features of a good solution with the bad ones with the help of migration. These transferred features tend to improve the bad solution while at the same time retaining goodness in the good solution. This is the essence of BBO. By the motivation of these approaches we try to develop an efficient face recognition algorithm in which we can use PCA as the basic face recognition technique along with gabor filters and use BBO to further optimize PCA by selecting the most desirable features from the face

## 1.2 Related Work

Face Recognition is typically used in security systems. There is a growing need of automatic face recognition system. Extracting the minimal set of features from an image for its representation is one of the fundamental tasks in face recognition. Researchers have been in search for the best face recognition techniques.

The problem of face recognition has been solved by using the traditional classical approaches like Discrete Cosine Transform [62], Linear Discriminant Analysis [63], Principal Component Analysis (PCA) [2, 19, 40, 41]. Some researchers also used neural network to propose a hybrid or semi-supervised method [64]. They combined unsupervised methods for extracting features and supervised methods for finding features able to reduce classification error. However these techniques alone can't be used to achieve high accuracy and some of them have their own limitations. For example PCA need images with proper alignment and uniform illumination [12].

Some researchers tried clubbing techniques to make hybrid algorithms for face recognition. Some researchers combined neural networks with Gabor filters to achieve high accuracy [65]. Some researchers used Gabor filters with Principal Component Analysis (PCA) and achieved high accuracy [12].

Recently some nature inspired techniques such as Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO) and hybrid ACO/PSO have been used in this domain and shown promising results [14, 15, 16]. Very recently a new nature inspired technique that is Biogeography Based Optimization (BBO), proposed by Dan Simon in 2008, has been applied on Discrete Cosine Transform (DCT) [11]. This motivated us to use BBO for face recognition in combination with PCA and gabor filter. Our algorithm first, correct the image input given to the PCA with the help of gabor filter to overcome the shortcomings of PCA. Finally, we extract the most required features from the image using PCA and apply BBO to further minimize the extracted features.

## 1.3 Problem Statement

The goal of face recognition is to match a given image against a large database of images to check its presence. Researchers have presented a lot of techniques for face recognition [1]. These techniques can be categorized as holistic matching method for e.g Principal Component Analysis (PCA) and local feature matching method [12].

In this work we will focus on PCA. PCA extracts the most desirable features that are required to represent a face. One of the biggest challenges in PCA is that we cannot use it on raw images directly. They need to be properly aligned and uniformly illuminated. This challenge can be solved by using gabor filters which takes a raw image, generates gabor filter response and convert the raw image into properly aligned and constantly illuminated image.

We have explored the research that showed that Gabor PCA based method for face recognition outperforms PCA based Eigenface method [3]. To improve further this hybrid Gabor PCA based technique we apply BBO to further minimize the desirable features from the image and go on to illustrate that in our result. The performance analysis supports our theory when compared to Gabor PCA based face recognition technique.

Our problem statement can be proposed as follows:

"**To develop a biogeography based algorithm for face recognition that can optimize the feature extraction from the face images**"

## 1.4 Scope of the Work

In this work we design a new face recognition algorithm based on biogeography that can work effectively and accurately to extract the most desirable features from the images. Our algorithm is capable of providing efficient face recognition. Our algorithm has been applied on the standard ORL database acquired at the Olivetti Research Laboratory, Cambridge, U.K.

In this work we have adapted BBO for the domain of image processing. First raw image is aligned using gabor filter, then PCA is used for extracting features that represent the face. BBO then selects the most desirable features from the output of PCA to provide us the minimal set of features. This makes selection as features get reduced. Thus our algorithm is hybrid of Gabor, PCA and BBO. We have designed two algorithms using two different approaches. First algorithm uses migration from ideal matrix and fitness function as mean whereas the second algorithm uses migration from test matrix and fitness function as standard mean. Both the algorithms have provided better result than the original gabor and PCA technique.

The scope of this work can be summarized as:

- To adapt BBO in our approaches to tackle the face recognition problem and to develop taxonomy of algorithms for the approaches we used to build an efficient biogeography based face recognition algorithm
- To test our hybrid algorithm on the standard ORL face database and acquire the result to support our algorithm. We have used the standard ORL database acquired at the Olivetti Research Laboratory, Cambridge, U.K to test our algorithm and to get results.

- To compare the efficiency of our algorithms with other technique. We have obtained the result in the form of ROC curves. We have plotted curves for the proposed approaches and the original PCA and gabor technique. The proposed approaches curve shows the improvement over the existing technique.

## 1.5 Organization of the Dissertation

Rest of work is organized as follows:

Chapter 2: Face Recognition

This section provides literature review for face recognition. It deals with the concept of Principal Component Analysis. It also provides the basic knowledge of various challenges associated with and how we can overcome those challenges with the help of gabor filters. It provides the evolution of face recognition techniques. It also provides the classification and details of various techniques used so far for face recognition.

Chapter 3: Optimization Techniques

This chapter talks about the optimization techniques like ant colony optimization (ACO), Particle swarm Optimization (PSO), Hybrid ACO/PSO and Biogeography based optimization (BBO).

Chapter 4: Proposed Approach

It explains our model for face recognition problem, approaches we have used to solve face recognition problem, parameters setting for the approaches and their algorithms.

Chapter 5: Experiments and Results

This section talks about the experimental setup we used to tackle the face recognition problem and the results we have obtained from various approaches, we developed for the problem.

Chapter 6: Conclusion and Future Scope

In this section the conclusion of the thesis work and the future scope of the work are presented.

Chapter 7: Publication from thesis.

This chapter gives the details of publications from the thesis and the conference details.

References: This section gives the reference details of the thesis.

Appendix A Abbreviations

Appendix B An introduction to MATLAB

# Chapter 2
# Face Recognition

In this chapter we understand the fundamentals of principal component analysis (PCA) and gabor filters. We discuss the development and evolution of face recognition techniques. We also discuss various sub processes like face detection, feature extraction etc used in the process of face recognition and their challenges and classification. We also discuss the different approaches for classification of face recognition techniques.

## 2.1 Face recognition- state of the art

Face recognition is one of the most challenging area in the field of image analysis. It's a challenge to build a system which can match human ability to recognize faces. Although humans can pretty easily identify faces known to them, we are unable to do it easily when we have to deal with a large number of unknown faces. The computers, possessing a huge amount of memory and computational speed, can easily outperform humans in this business.

Face recognition is still an unsolved problem which requires a technology as described in table 2.1 below. There are many different industry areas which are interested in what face recognition could offer. Some of the examples are human-machine interaction, video surveillance, photo cameras, virtual reality or law enforcement. This is a multidisciplinary interest pushes the research and attracts researchers from various disciplines. Therefore, face recognition can't be considered as a problem limited to area of computer vision. Face recognition is an important subject in neural networks, pattern recognition, image processing, computer graphics and psychology [31]. In fact, psychology is the area in which the earliest works on this subject was done [32]. They also gave attention to other issues like perception of gestures, face expression or interpretation of emotion.

| Areas | Applications |
| --- | --- |
| Information Security | Access security (OS, data bases)<br>Data privacy (e.g. medical records)<br>User authentication (trading, on line banking) |
| Access management | Secure access authentication (restricted facilities)<br>Permission based systems<br>Access log or audit trails |
| Biometrics | Person identification (national IDs, Passports, voter registrations, driver licenses)<br>Automated identity verification (border controls) |
| Law Enforcement | Video surveillance<br>Suspect identification<br>Suspect tracking (investigation)<br>Simulated aging<br>Forensic Reconstruction of faces from remains |
| Personal security | Home video surveillance systems<br>Expression interpretation (driver monitoring system) |
| Entertainment - Leisure | Home video game systems<br>Photo camera applications |

Table2.1 Face Recognition Application

Interest in face recognition as an engineering field was started in the 1960's. One of the early researches on this subject was done by Woodrow W. Bledsoe. Bledsoe, along other researches, started Panoramic Research, Inc., in Palo Alto, California. The most of the work undertaken by this company comprises of Artificial Intelligence concerned contracts from the U.S. Department of Defense and various intelligence agencies [33].

Bledsoe, along with Charles Bisson and Helen Chan, worked on how to use computers to perform face recognition [34, 35, 36]. At Stanford Research Institute, he continued his research [36]. Bledsoe implemented a semi-automatic system in which a human operator was used to select some face recognition, and then this information is used by computers for recognition. Most of the problems from which face recognition today suffer like variations in illumination, head rotation, facial expression and aging were described by him about 50 years back.Many approaches were used back on the 1970's. Some researchers first used a set of geometric parameters to describe a face and then based on these parameters they perform some pattern recognition. But in 1973, kenade developed the first fully automated face recognition system was Kenade [37]. He implemented a face recognition program and ran in a computer system specially designed for this purpose. The algorithm extracted sixteen facial parameters automatically. In his work, Kenade compares this automated extraction of facial parameters with the manual extraction, resulting in only a small difference. He got a correct identification rate of 45-75%. He showed that better results can be achieved in the absence of irrelevant features.

In the 1980's researchers followed a variety of approaches, majority of them continuing with previous methods. Some of the research work which tried to improve the techniques by measuring subjective features. Some researchers used artificial neural networks to build face recognition algorithms [38]. The first practical use of eigenfaces in image processing, a technique that would prove to be a dominant approach in following years, was made by L. Sirovich and M. Kirby [39]. The methods used in their approach were based on the Principal Component Analysis (PCA). Their aim was of describing an image in a reduced dimension and not losing important information, and later uses it to get the image again [40]. Their research provided the foundation for many new face recognition algorithms.

In 1990's the mentioned eigenface approach was broadly recognized as the foundation for the first industrial applications and state of the art techniques. Alex Pentland and Mathew Turk of the MIT presented their research work for face recognition using eigenfaces [2]. Their algorithm was capable of tracking, locating and classifying a face. In 1990's, with a considerable increase in the number of publications face recognition area attracted a lot of attention. Many approaches were developed which has lead to different algorithms. Some of the most important algorithms in this field are Principal Component Analysis (PCA), Independent Component Analysis (ICA), Linear Discriminant Analysis (LDA) and their derivatives. The technologies which use face recognition as a technique have also undergone evolution through the years.

## 2.2 Face Recognition System: Structure

Face Recognition is a big task that involves several sub problems. In face recognition system, input is always given in the form of a video stream or an image. The output of face recognition system is either verification or identification of the subject that was given as input in the video or image. Some approaches define face recognition as a process that comprises of three step - see Figure 2.1 [31].


Figure 2.1. A simple face recognition system

Face detection can be defined as the process in which of faces gets extracted from scenes. So, for this to happen the system must positively detect a certain image region as a face. This procedure has many applications like pose estimation, face tracking or compression. The next step, which is feature extraction, involves obtaining important facial features from the face image. These features could be certain variations, face regions, angles or measures, which can be human relevant (e.g. eyes spacing) or not. This phase has other applications like emotion recognition or facial feature tracking. Finally, the system recognizes the face. In an identification task, an identity is reported by the system from a database which is used to conduct search. **This phase involves a comparison method, a classification algorithm and an accuracy measure**. In this phase some methods common to many other areas are used which also do some classification process like sound engineering and data mining. These phases can be merged, or new ones could be added. Therefore, we could find many different engineering approaches to a face recognition problem. Face recognition and detection could be performed as one behind the other, or we can first analyze an expression and then can normalize a face [41].

### 2.2.1 Detecting a Face

There are certain applications of Face Recognition which don't require face detection. In few cases, face images in the facial databases are stored in normalized form only. Then the image is input in a predefined standard format, such that there is no detection need to be performed. Criminal data base could be an example in this context. There are many law enforcement agencies in the world which stores only faces of criminals with a criminal report.

Face detection is not necessary in cases like if there is new subject and the cops have already stored his or her image somewhere. However, the traditional images input to the computer vision systems are not that appropriate. They may contain large number of faces. In these cases face detection is strictly required. It can't be avoided even if we have developed an automated face tracking system. As an example, face detection, tracking and recognizing are included in video surveillance systems. So, it's logical to think face detection as part of face recognition problem. Face detection deals with many well known challenges [18, 31]. They are mostly performed in images taken in uncontrolled environments like surveillance video systems. These challenges can be because of some factors:

- **Pose variation** The best condition for face detection is when only frontal images were involved. But, the probability of satisfying this is very low in general unaltered conditions.

Also, the performance of face detection algorithms decreases rapidly when there are large variations are observed in poses. It's an interesting research area. Major reasons for Pose variation could be camera's angle or subject's movements.

- **Feature occlusion** High variability in the face image is introduced because of the presence of glasses, beards or hats. Faces can also get covered by some objects.
- **Facial expression** Variety of facial gestures can bring variations in facial features.
- **Imaging conditions** Different lighting conditions and cameras can also affect the quality of an image, affecting the face appearance.

There exist some problems concerned with face detection besides face classification and extraction of features. For instance, face location is a simplified approach of face detection. Its aim is to find the location of a face present in an image where there's single face only. We can differentiate between face location and face detection, since the former is a simplified problem of the latter. Methods like determining boundaries that represents head were first tested on this scenario and then applied to more complex problems [42]. Detecting features of face is related with locating and detecting some important features, such as nose, eyebrow, lips, ears, etc. Some researchers even tried using human skin color and eye blink detection as a basis for face detection [43, 44, 45].

Some algorithms which are used to extract features are based on detection of facial features. There is much literature on this topic, which is discussed later. Face tracking is a part of face recognition which sometimes results as an aftereffect of face detection. Many system's aim at not only face detection, but also aim at locating this face in no time or in real time. Once again, a good example could be of video camera surveillance.

Face Detection is a big task that can be divided into many small tasks. Some systems are capable of detecting and locating faces simultaneously, others first try to detect the face with the help of a detection routine and then, if positive, then the face is located. This may require use of some tracking algorithms - see Figure 2.2

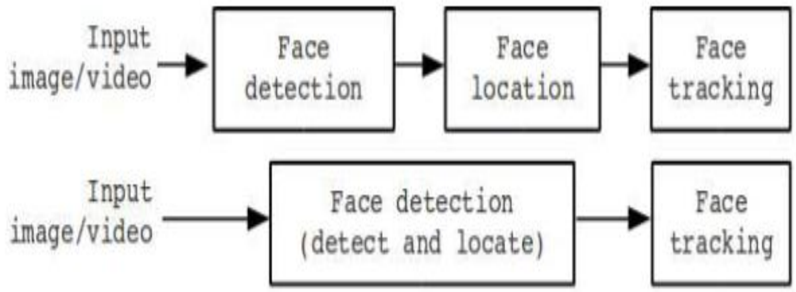



Figure 2.2 Face Detection

Face tracking is essentially a motion estimation problem. Face tracking can be performed using many different methods, e.g., head tracking, feature tracking, image-based tracking, model-based tracking. These are different ways to classify these algorithms [31]:

- **Head tracking/Individual feature tracking** The head can be tracked as a whole entity, or certain features tracked individually.
- **2D/3D** Two dimensional systems track a face and output an image space where the face is located. Three dimensional systems, on the other hand, perform a 3D modeling of the face. This approach allows to estimate pose or orientation variations.

The basic face tracking process seeks to locate a given image in a picture. Then, it has to compute the differences between frames to update the location of the face. There are many issues that must be faced: Partial occlusions, illumination changes, computational speed and facial deformations.

Face detection algorithms mostly have some common steps. Firstly, data dimension reduction is performed to obtain a short response time without losing important information present in data. Sometimes pre-processing can also be performed to make the image a perfect input for the algorithm. Then, some algorithms try to extract certain relevant facial regions and some analyze the image as it is. The next step is of facial features extraction. Then to detect the face these extracted features will weighted, evaluated or compared. Finally, some algorithms are equipped with a learning procedure which helps them to include new data to their models and learn.

Face detection, basically consists of two class problems where we obtain need to answer to a question that if a face is present in a picture or not. This approach of obtaining the answer can be described as a simplification of face recognition problem. A Face recognition also performs classification for a given face, and the number of classes are equal to the number of candidates. Consequently, most of the methods used for face detection are much similar to algorithms used for face recognition. We can also say it like, techniques used to detect faces are often the same used for face recognition

### 2.2.1.1 Division of face detection methods

It's not easy to give taxonomy of face detection methods. There isn't a globally accepted grouping criteria. They usually mix and overlap.  Yan, Kriegman and Ahuja presented a classification that is well accepted [18]. Four categories have been made to categorize the methods. The design of categories is such that they may overlap, due to this algorithm can simultaneously exist in two or more categories. This categorization is as follows:

- **Knowledge based methods**

  These methods are based on certain rules. They form rules by representing the captured knowledge present in faces. Some basic rules from the faces can be easily guessed. For instance, a face comprises of two eyes symmetric to each other, and the eye area which consists of pupil is always darker as compared with cheeks. Distance between the two eyes or the intensity difference of skin between the lower zone and the eyes could be some useful facial features. The major problem associated with these methods is how to build proper set of rules. If the rules become too basic then it could be false positive. On the other hand, if rules become very precise and too much detailed then it could lead to false negatives. A solution to these problems could be forming some knowledge based methods which are of hierarchical nature. But, this strategy alone is very confined. It's unable to find many faces in a complex image. Other researchers have put effort in finding features for face detection which do not varies. This idea helps in removing the limited instinctive knowledge, we posses about the faces. One early algorithm was developed by Han, Liao, Yu and Chen in 1997 [46]. The method is divided in several steps. Firstly, it tries to find pixels that represent eye, so it eliminates unnecessary pixels present in the image. After performing the above mentioned process, they each eye-analogue part is considered as an individual candidate. Then, we try to confirm the potential pair of eyes by executing some rules. Once we confirm the eyes, the face area is calculated in the form of rectangle by the algorithm. Some functions are used to determine four vertexes associated with the face. In this way we normalize potential faces into a fixed orientation and in a fixed size. Then, we verify the face regions using a back propagation neural network. Finally, they apply a cost function to make the final selection. They report a success rate of 94%, even in photographs with many faces. These methods show themselves efficient with simple inputs.

- **Template matching based methods**

Template matching methods are methods which try to represent a face in the form of a function. A basic template which can be used for all the faces is very much required.

We can define independently define different features in different way. For example, a face can be broken into nose, eyes, face contour and mouth. Also a face model can be built by edges. But these methods are limited to faces that are frontal and unoccluded. A face can also be represented as a silhouette. Other templates try to use the similarity between face regions using two parameters i.e darkness and brightness. These observed information is used in comparison to the images input for face detection. This approach can be easily implemented, but it’s inadequate for face detection. It is unable to provide good results if found variations in scale, shape and pose. However, deformable templates have been proposed to deal with these problems.

- **Appearance based methods**

The templates in appearance-based methods have a learning process to learn from the examples in the images. Appearance-based methods usually depend on techniques from the field of machine learning and statistical analysis to obtain the important characteristics present in face images. Some methods which are based on appearances work in a network associated with probability. A feature vector obtained from an image is considered as a random variable with which probability is associated of belonging to a face or not. Another approach could be to explore a function which can easily discriminate between non-face and face classes. Feature extraction in face recognition also makes use of these methods and these will be discussed later. Nevertheless, these are the most relevant methods or tools:

i. **Eigenface based** Sirovich and Kirby developed a method for efficiently representing faces using PCA (Principal Component Analysis) [39, 40]. Their goal of this approach is to represent a face as a coordinate system. The vectors that make up this coordinate system were referred to as eigenpictures. Later, Turk and Pentland used this approach to develop a eigenface-based algorithm for recognition [2].

ii. **Distribution-based** These systems where first proposed for object and pattern detection by Sung [47]. The idea is collect a sufficiently large number of sample views for the pattern class we wish to detect, covering all possible sources of image variation we wish to handle. Then an appropriate feature space is chosen. It must represent the pattern class as a distribution of all its permissible image appearances. The system matches the candidate picture against the distribution-based canonical face model. Finally, there is a trained classifier which correctly identifies instances of the target pattern class from background image patterns, based on a set of distance measurements between the input pattern and the distribution-based class representation in the chosen feature space. Algorithms like PCA or Fisher's Discriminant can be used to define the subspace representing facial patterns.

iii. **Neural Networks** Many pattern recognition problems like object recognition, character recognition, etc. have been faced successfully by neural networks. These systems can be used in face detection in different ways. Some early researches used neural networks to learn the face and non-face patterns. They defined the detection problem as a two-class problem. The real challenge was to represent the "images not containing faces" class. Other approach is to use neural networks to find a discriminant function to classify patterns using distance measures [47]. Some approaches have tried to find an optimal boundary between face and non-face pictures using a constrained generative model [48].

iv. **Support Vector Machines (SVMs)** These are linear classifiers which can increase the margin between the decision hyperplane and the examples in the training set. So, an optimal hyperplane should minimize the classification error of the unseen test patterns. This classifier was first applied to face detection by Osuna [49].

v. **Naive Bayes Classifiers** Schneiderman and Kanade described an object recognition algorithm that modeled and estimated a Bayesian Classifier [50]. They computed the probability of a face to be present in the picture by counting the frequency of occurrence of a series of patterns over the training images. They emphasized on patterns like the intensity around the eyes. The classifier captured the joint statistics of local appearance and position of the face as well as the statistics of local appearance and position in the visual world. Overall, their algorithm showed good results on frontal face detection. Bayes Classifiers have also been used as a complementary part of other detection algorithms.

vi. **Hidden Markov Model** This statistical model has been used for face detection. The challenge is to build a proper HMM, so that the output probability can be trusted. The states of the model would be the facial features, which are often defined as strips of pixels. The probabilistic transition between states is usually the boundaries between these pixel strips. As in the case of Bayesians, HMMs are commonly used along with other methods to build detection algorithms.

### 2.2.2 Extracting Features

Humans are able to recognize faces since they are 5 year old. It seems to be a very well defined automated and dedicated process that took place in our brains. What it’s clear is that we can recognize people we know, even when they are wearing glasses or who have grown a beard. In fact, face recognition's goal is to extract information from images. This feature extraction process can be defined as the process in which important information is extracted from a face image. This information must be valuable to the later step of identifying the subject with an acceptable error rate. The feature extraction process must be efficient in terms of memory usage and computing time. The output should also be optimized for the classification step.

Feature extraction comprises of several steps - dimensionality reduction, feature extraction and feature selection. These steps may overlap, and dimensionality reduction could be seen as a consequence of the feature extraction and selection algorithms. Both algorithms could also be defined as cases of dimensionality reduction.

Dimensionality reduction is an essential task in any pattern recognition system. The performance of a classifier depends on the amount of sample images, number of features and classifier complexity. However, if the number of features present in the images increases then the performance of a classification algorithm starts degrading. This may occur due to the fact that the number of training samples is less as compared to the number the features. This problem is called "curse of dimensionality" or "peaking phenomenon". A generally accepted method of avoiding this phenomenon is to use at least ten times as many training samples per class as the number of features. This requirement should be satisfied when building a classifier. The more complex the classifier, the larger should be the mentioned ratio [51]. This "curse" is one of the reasons why it's important to keep the number of features as small as possible. The other main reason is the speed. The classifier will be faster and will use less memory. Moreover, a large set of features can result in a false positive when these features are redundant. Ultimately, the number of features must be carefully chosen. Too less or redundant features can lead to a loss of accuracy of the recognition system. We can make a distinction between feature extraction and feature selection. Both terms are usually used interchangeably. Nevertheless, it is recommendable to make a distinction. A feature extraction algorithm extracts features from the data. It creates those new features based on transformations or combinations of the original data. In other words, it transforms or combines the data in order to select a proper subspace in the original feature space. On the other hand, a feature selection algorithm selects the best subset of the input feature set. It discards non-relevant features. Feature selection is often performed after feature extraction. So, features are extracted from the face images, then a optimum subset of these features is selected. The dimensionality reduction process can be embedded in some of these steps, or performed before them. This is arguably the most broadly accepted feature extraction process approach - see figure 2.3.

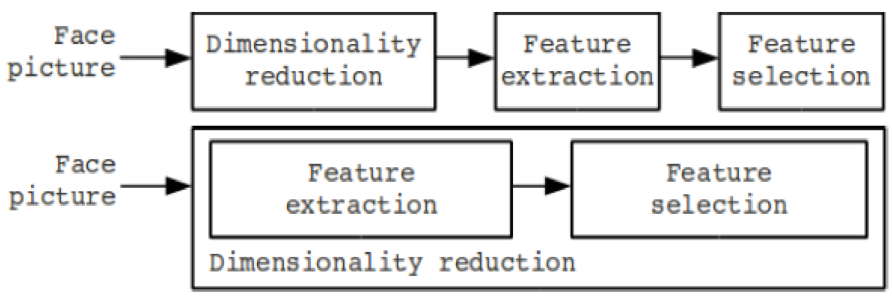



Figure 2.3 Feature Extraction

#### 2.2.2.1 Methods for Extracting Features

There are many feature extraction algorithms. They will be discussed later in this thesis. Most of them are used in other areas than face recognition. Researchers in face recognition have used, modified and adapted many algorithms and methods to their purpose.

For example, PCA was invented by Karl Pearson in 1901, but proposed for pattern recognition 64 years later [52, 53]. Finally, it was applied to face representation and recognition in the early 90's

[2, 39, 40]. Some commonly used feature extraction algorithms are Principal Component Analysis (PCA), Kernel PCA, Weighted PCA, Linear Discriminant Analysis (LDA), Independent Component Analysis (ICA), Kernel LDA, Neural network based methods, Discrete Cosine Transform (DCT) and Gabor wavelet transforms. PCA and its technique has been discussed in detail in chapter 2. Kernel PCA is quite similar to PCA. Kernel PCA is also eigenvector based method but unlike PCA it is based on non linear map and uses kernel methods. Weighted PCA is PCA which uses weighted coefficient in its technique. LDA is quite different from PCA but it is also a eigenvector based method which uses supervised linear map. Kernel LDA is a variant of LDA which uses kernel methods just like kernel PCA. Neural network based methods can also be combined with PCA and Gabor filters [65]. DCT uses fourier transform [10, 62].

### 2.2.3 Face Classification

The next step after features extraction and selection is the image classification. Different types of classification methods are used by appearance-based face recognition methods. It is also possible to use a combination of two or more classifiers to obtain better results. On the other hand, many of the model-based algorithms use model or template to match the samples. Then, a learning method is can be used to improve the algorithm. One way or another, classifiers have a big impact in face recognition. Classification methods are used in many areas like data mining, finance, signal decoding, voice recognition, natural language processing or medicine. Here classifiers will be addressed from a general pattern recognition point of view.

Classification algorithms usually involve some learning - supervised, unsupervised or semi-supervised. Unsupervised learning is the most difficult approach, as there are no tagged examples. However, many face recognition applications include a tagged set of subjects. Consequently, most face recognition systems implement supervised learning methods. There are also cases where the labeled data set is small. Sometimes, the acquisition of new tagged samples can be infeasible. Therefore, semi-supervised learning is required.

#### 2.2.3.1 Classifiers

According to Jain, Duin and Mao, there are three concepts that are key in building a classifier - similarity, probability and decision boundaries [51]. We will present the classifiers from that point of view.

- **Similarity** This approach is simple and intuitive. Similar patterns should belong to the same class. Algorithms implemented later used this approach. The goal is to define a metric that defines similarity and a representation of the samples which belongs to the same class. For example, the metric can be the euclidean distance. The representation of a class can be the mean vector of all the patterns belonging to this class. The 1-NN decision rule can be used with this parameters. Its classification performance is usually good. This approach is similar to a k-means clustering algorithm in unsupervised learning.

  There are other techniques that can be used. For example Vector Quantization, Learning Vector Quantization or Self-Organizing Maps. Researches classify face recognition algorithm based on different criteria. Some publications defined Template Matching as a kind or category of face recognition algorithms [54]. However, we can see template matching just as another classification method, where unlabeled samples are compared to stored patterns.

- **Probability** Some classifiers are build based on a probabilistic approach. Bayes decision rule is often used. The rule can be modified to take into account different factors that could lead to miss-classification. Bayesian decision rules can give an optimal classifier, and the

Bayes error can be the best criterion to evaluate features. Therefore, a posteriori probability functions can be optimal.

- **Decision Boundaries** This approach can become equivalent to a Bayesian classifier. It depends on the chosen metric. The main idea behind this approach is to minimize a criterion (a measurement of error) between the candidate pattern and the testing patterns. One example is the Fisher's Linear Discriminant (often FLD and LDA are used interchangeably). It's closely related to PCA. FLD attempts to model the difference between the classes of data, and can be used to minimize the mean square error or the mean absolute error. Other algorithms use neural networks. Multilayer perceptron is one of them. They allow nonlinear decision boundaries. However, neural networks can be trained in many different ways, so they can lead to diverse classifiers. They can also provide a confidence in classification, which can give an approximation of the posterior probabilities. A decision tree is a special type of classifier. The training in case of decision tree is performed by selecting the most important features of each node present in the tree, iteratively. During classification, just the needed features for classification are evaluated, so feature selection is implicitly built-in. The decision boundary is built iteratively.

  Other method widely used is the support vector classifier. These support vectors define the classification function. Support Vector Machines (SVM) are originally two-class classifiers. That's why there must be a method that allows solving multiclass problems. There are two main strategies [55]:

  i. **One vs. all approach** A SVM per class is trained. Each one separates a single class from the others.

ii. **Pairwise approach** Each SVM separates two classes. A bottom-up decision tree can be used, each tree node representing a SVM. The coming face's class will appear on top of the tree.

### 2.2.4 Different approaches to Face Recognition

Face recognition is a challenging research area, changing and improving constantly. Computer vision, optics, pattern recognition, neural networks and machine learning are the few examples of research areas which affect face recognition. The figure below shows different approaches to face recognition.

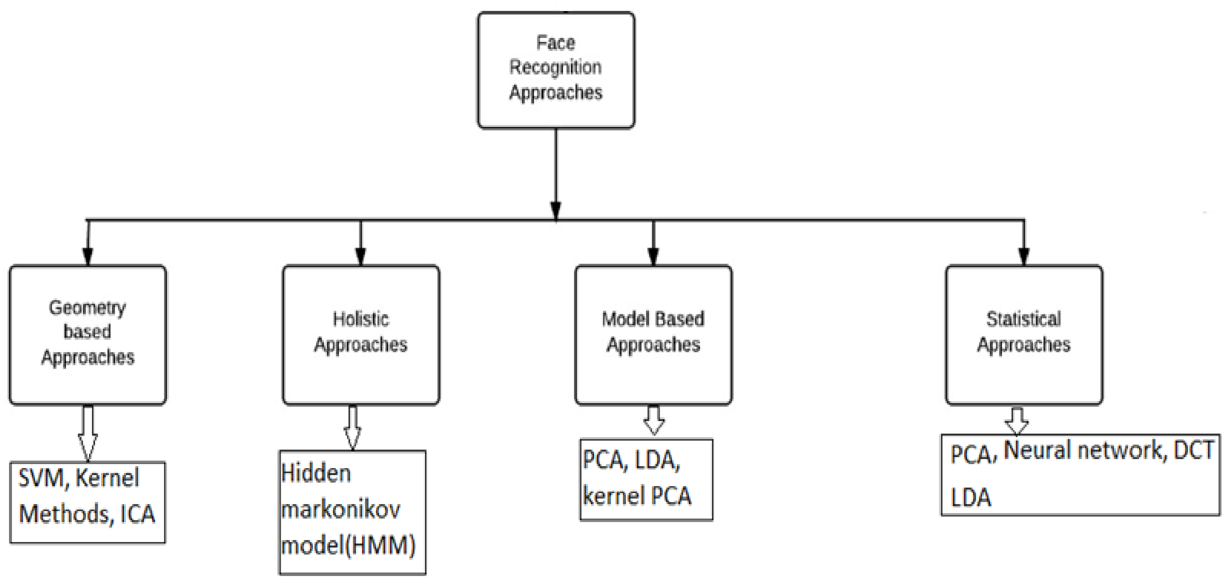



Figure 2.4 Approaches to face recognition

Previous sections explain the different stages of a face recognition process. However, these steps can overlap or change depending on the bibliography consulted. There is not a consensus on that regard. All these factors hinder the development of a unified face recognition algorithm classification scheme. This section explains the most cited criteria.

### 2.2.4.1 Template or Geometry based approaches

Face recognition algorithms can be classified as either geometry based or template based algorithms [56, 57]. The template based methods compare the input image with a set of templates. The set of templates can be constructed using statistical tools like Support Vector Machines (SVM) [55], Principal Component Analysis (PCA) [2, 12, 41], Linear Discriminant Ananlysis (LDA) [20], Independent Component Analysis (ICA) [58] or Kernel Methods [59]. The geometry feature-based methods analyze local facial features and their geometric relationships. This approach is sometimes called feature based approach [54]. This approach is less used nowadays. There are algorithms developed using both approaches as well.

### 2.2.4.2 Holistic approaches

Faces can often be identified from little information. Some algorithms follow this idea, processing facial features independently. In other words, the relation between the features or the relation of a feature with the whole face is not taken into account. Many early researchers followed this approach, trying to deduce the most relevant features. Some approaches tried to use the eyes, a combination of features [54], and so on. Some Hidden Markov Model (HMM) methods also fall in this category. Although feature processing is very important in face recognition, relation between features (configural processing) is also important. In fact, facial features are processed holistically. That’s why nowadays most algorithms follow a holistic approach.

### 2.2.4.3 Appearance based or model based approaches

Face recognition methods can be divided into appearance based or model based algorithms. The differential element of these methods is the representation of the face. Appearance-based methods represent a face in terms of several raw intensity images. An image is considered as a high-dimensional vector. Then statistical techniques are usually used to derive a feature space from the image distribution. The sample image is compared to the training set. On the other hand, the model-based approach tries to model a human face. The new sample is fitted to the model, and the parameters of the fitted model used to recognize the image. Appearance methods can be classified as linear or non-linear, while model-based methods can be 2D or 3D. Linear appearance based methods perform a linear dimension reduction. First, the face vectors are projected to the basis vectors, then the feature representation of each face image is done by projection coefficients. Examples of this approach are LDA or ICA. Non-linear appearance methods are more complicate. In fact, linear subspace analysis is an approximation of a nonlinear manifold. KernelPCA (KPCA) is a method widely used [60]. Model-based approaches can be 2-Dimensional or 3-Dimensional. These algorithms try to build a model of a human face. These models are often morphable. A morphable model allows to classify faces even when pose changes are present. 3D models are more complicate, as they try to capture the three dimensional nature of human faces. Example of this approach is 3D Morphable Models [61].

#### 2.2.4.4 Neural network, Template and Statistical approaches

A similar separation of pattern recognition algorithms into four groups is proposed by Jain [51]. We can group face recognition methods into three main groups. The following approaches are proposed:

- **Template matching** Patterns are represented by samples, models, pixels, curves, textures. The recognition function is usually a correlation or distance measure.
- **Statistical approach** Patterns are represented as features. The recognition function is a discriminant function.

- **Neural networks** The representation may vary. There is a network function in some point.

Note that many algorithms, mostly current complex algorithms, may fall into more than one of these categories. The most relevant face recognition algorithms will be discussed later under this classification.

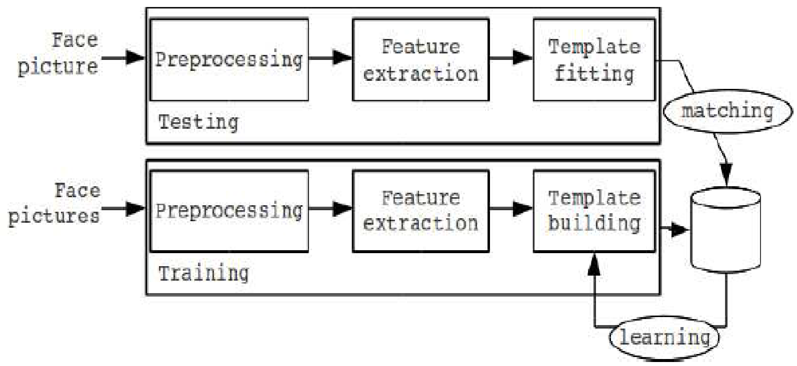



Figure 2.5 Template matching method

## 2.3 Statistical approaches for face recognition

Face images are usually available as large arrays of pixel, often. In statistical approach, each image can be represented in terms of the certain set of features. So, if ‘d’ is the total number of pixels needed to represent the image then the image can be viewed as a point (vector) in a d-dimensional space [12]. The dimensionality which is the number of coordinates required to specify a data point- of this data is very high. So, the aim is to wisely select and use the right statistical tool for extraction and analysis of the features. These tools must be able to represent the face space in the image space and extract the basis functions from the face image. This would permit patterns belonging to different classes to occupy disjoint and compacted regions in the feature space.

Consequently, we would be able to define a line, curve, plane or hyperplane that separates faces belonging to different classes.

Many of these statistical tools are not used alone. They are modified or extended by researchers in order to get better results. Some of them are embedded into bigger systems, or they are just a part of a recognition algorithm.

Different statistical approaches are as follows:

### 2.3.1 Discrete Cosine Transform

The Discrete Cosine Transform, DCT-II standard (often called simply DCT) represents a series of data points as a sum of mathematical cosine functions oscillating with different frequencies [10, 62]. It has strong energy compaction properties. Therefore, it can be used to transform images, compacting the variations, allowing an effective dimensionality reduction. They have been widely used for data compression. The DCT is based on the Fourier discrete transform, but using only real numbers. When a DCT is performed over an image, the energy is compacted in the upper-left corner. An example is shown in figure 2.6. The face image has been taken from the ORL database, and a DCT performed over it.

Figure 2.6 DCT of a Face image

Let C is the DCT of the image$D_{NxM}$.

$$Cpq = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} D_{mn} a_p a_q \cos\frac{\Pi(2m+1)p}{2M}\cos\frac{\Pi(2n+1)q}{2N} \qquad (1)$$

where

$$a_p = \begin{cases} \frac{1}{\sqrt{M}}, p = 0 \\ \sqrt{\frac{2}{M}}, 1 \le p \le M-1 \end{cases}$$

$$a_q = \begin{cases} \frac{1}{\sqrt{N}}, q = 0 \\ \sqrt{\frac{2}{N}}, 1 \le q \le N-1 \end{cases}$$

where M is the row size and N is the column size of A. We can truncate the matrix B, retaining the upper-left area, which has the most information, reducing the dimensionality of the problem. where M is the row size and N is the column size of A. We can truncate the matrix B, retaining the upper-left area, which has the most information, reducing the dimensionality of the problem.

### 2.3.2 Linear discriminant Analysis

LDA is widely used to find linear combinations of features while preserving class separability. Unlike PCA, LDA tries to model the differences between classes. Classic LDA is designed to take into account only two classes. Specifically, it requires data points for different classes to be far from each other, while points from the same class are close. Consequently, LDA obtains differenced projection vectors for each class. Multi-class LDA algorithms which can manage more than two classes are more used.

Suppose we have m samples $x_1,...,x_m$ belonging to c classes; each class has $m_k$ elements. We assume that the mean has been extracted from the samples, as in PCA. The objective function of the LDA can be defined as [63]:

$$a_{opt} = argmax \frac{a^T s_b a}{a^T s_t a} \tag{2}$$

$$s_b = \sum_{k=1}^{c} (m_k u^k (u^k)^T) = X W_{mxm} X^T \tag{3}$$

$$s_i = \sum_{i=1}^{m} (x_i (x_i)^T) = X X^T \tag{4}$$

where $W_{wxm}$ is a diagonal matrix defined as

$$W_{wxm} = \begin{pmatrix} W^1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ & \cdots & \end{pmatrix} \tag{5}$$

Finally we write the eigen problem as

$$S_b a = \lambda a \tag{6}$$

The solution of this eigenproblem provides the eigenvectors; the embedding is done like the PCA algorithm does.

### 2.3.3 Principal Component Analysis

Extraction of features from the images is the foremost step in face recognition. PCA is a standard technique used for feature extraction. PCA was first introduced by Pearson in 1901 and developed independently by Hotteling in 1933 [19]. Feature extraction is the conversion of input images into the set of features. The input images contain large amount of data which is difficult to be processed and also there exists redundancy of data contained in these images. So feature extraction helps us to convert the huge data present in the images into a reduced set of features [10].

The basic idea of PCA is dimensionality reduction. It reduces the dimensionality of input data which contains a large number of interrelated data and at the same time it also tries to preserve the maximum possible variance present in the input data. This dimensionality reduction can be accomplished by transforming the input data into a new set of variables known as principal component (PCs).

These new set of variables, the principal components are uncorrelated and they preserve the maximum possible variations present in the original variables that belongs to the input data. Computation of principal components reduces to the solution of an eigen value-eigen vector problem. This simple technique has a large number of application and as well as a different number of applications [19].

#### 2.3.3.1 Interpreting Principal Components: Anatomical measurements

Anatomical measurement is one of the several applications where PCA can be useful. There are variations present in anatomical measurements of different species. PCA can be useful in finding the source of these variations. On individuals of a species a large number of measurements are performed. The first principal component mostly has positive coefficients for all variables and it also reflects the overall size of the individuals. Later principal components contrast some measurements of an individual with the other individuals. These principal components also define some important aspects of shape. Table 2.2 below represents a small data set of 28 students (15 women and 13 men). The seven measurements which are made on each individual are lengths of hand, forearm, circumferences of chest, waist, wrist and head and overall height.

The PCA is performed on correlation matrix. The nature of the first two PCs are similar for the two sexes with some similarity for the third PC too. It appears that sources of variations as observed by the nature of the three PCs, are similar for each sex. In order to get better estimates of three PCs we can perform PCA on all 28 observations. When we interpret PCs with other data present in tabular form, it is the coefficient pattern that we are interested in and not in the values to several decimal places which can provide a wrong precision.

Table 2.3 simplifies the table 2.2 further. A '+' or '-' indicates a coefficient whose absolute value if we take would be greater than the half of the maximum of absolute value of the corresponding principal component.

| Component number | | 1 | 2 | 3 |
|---|---|---|---|---|
| | | | Women | |
| Hand | Coefficients | 0.33 | 0.56 | 0.03 |
| Wrist | | 0.26 | 0.62 | 0.11 |
| Height | | 0.40 | −0.44 | −0.00 |
| Forearm | | 0.41 | −0.05 | −0.55 |
| Head | | 0.27 | −0.19 | 0.80 |
| Chest | | 0.45 | −0.26 | −0.12 |
| Waist | | 0.47 | 0.03 | −0.03 |
| Eigenvalue | | 3.72 | 1.37 | 0.97 |
| Cumulative percentage of total variation | | 53.2 | 72.7 | 86.5 |
| | | | Men | |
| Hand | Coefficients | 0.23 | 0.62 | 0.64 |
| Wrist | | 0.29 | 0.53 | −0.42 |
| Height | | 0.43 | −0.20 | 0.04 |
| Forearm | | 0.33 | −0.53 | 0.38 |
| Head | | 0.41 | −0.09 | −0.51 |
| Chest | | 0.44 | 0.08 | −0.01 |
| Waist | | 0.46 | −0.07 | 0.09 |
| Eigenvalue | | 4.17 | 1.26 | 0.66 |
| Cumulative percentage of total variation | | 59.6 | 77.6 | 87.0 |

Table 2.2 first 3 PCs: anatomical measurements [19]

In the similar manner a (+) or (-) indicates a coefficient whose absolute value if we take would be between a quarter and half of the maximum of absolute value for the corresponding principal component.

There are many other ways to simplify principal components as shown in table 2.2.

| Component number | 1 | 2 | 3 |
|---|---|---|---|
| | | Women | |
| Hand | + | + | |
| Wrist | + | + | |
| Height | + | − | |
| Forearm | + | | − |
| Head | + | (−) | + |
| Chest | + | (−) | |
| Waist | + | | |
| | | Men | |
| Hand | + | + | + |
| Wrist | + | + | − |
| Height | + | (−) | |
| Forearm | + | − | + |
| Head | + | | − |
| Chest | + | | |
| Waist | + | | |

Table 2.3 Simplified version of the coefficient in table 2.2[19]

For example we can resize the coefficient of each principal component so that maximum value is '+1' and minimum value is '-1'. We write only values of coefficient which are rounded to one decimal place and value is above the cutoff, like 0.5 or 0.7. Coefficient values which are below the cutoff are not written in the table and a blank space is left at the position. It is always advised to round off the coefficient value to one or two decimal places.

As far as interpretation of PCs is concerned with respect to above example, the first PC is used to measure the overall size for men as well as women. It can also be confirmed by the correlations between all the seven variables, which are positive as expected.

First PC can be accounted for 60% (men) and 53% (women) of the total variations. The second PC tells us that the second main source of variation is between individuals having large hand and wrist measurements with respect to their heights. Head and chest measurements in case of women are contributors to the second PC. In case of men forearm measurement which is close to height is the contributor. The second PC is responsible for around 20% of the total variation for men and women. The third PC differ

more between men and women. In case of women it shows a contrast between head and forearm and in case of men, in addition to these wrist and hand measurements have the same signs as head and forearm, respectively. Third PC's contribution to the total variation is 9-14%. Overall, the first three PCs contribute to 86.5% and 87% for both women and men respectively [19].

## 2.3.3.2 PCA Technique

PCA is a standard technique to extract features form images. The image in the form of A x A can be expressed as a point in the space A x A dimensions. The goal of PCA is to find the required vectors that can represent the image information and form another space. PCA steps are as follows:

1. Let the training set of face images be $\Gamma_1, \Gamma_2, \Gamma_3, \dots\dots\dots, \Gamma_p$.
2. We calculate the average face of the set by $\psi = \frac{1}{p}\sum_{r=1}^{p}\Gamma_r$.
3. Now we calculate the difference of each face from the average by vectors $\phi = \Gamma_i - \psi$.
4. We calculate eigen values and eigen vectors from covariance matrix as shown (1) and a new image is transformed into its face components by this operation [2].

$$c = \frac{1}{p}\left[\sum_{R=1}^{p} \emptyset_R {\emptyset_R}^T\right] \quad (7)$$

5. We calculate the weights using (2) and from them we form a vector $\Omega = \omega_1, \omega_2, \dots.\omega_p$ that represents the contribution of each eigenvector in representation of input face image.

$$\omega_k = \mu_K^T(\Gamma - \psi), \text{ k} = 1,2,\dots\dots\text{P} \quad (8)$$

Where $\mu_k$ is eigen vector.

These weights may be used in a face classification algorithm to find which of predefined face classes that describe the face. [12].

### 2.3.3.3 PCA Challenges

One of the biggest challenges in PCA is that we cannot use raw images directly. They need to be properly aligned and uniformly illuminated. In the below figure 2.7 two images are shown of different illumination [20]. The unwanted variations because of illumination and facial expressions are often retained by PCA [20]. The variations between the face image of the same person because of viewing direction and illumination are often larger than the variations we get because of change of the face [21].

Figure 2.7 The same person seen under different lightening conditions can appear dramatically different. In the left image the dominant light source is near head-on; in the right image the light source is from above and to the right [20].

### 2.3.3.4 Gabor Filter

Gabor filters are also known as gabor wavelets or kernels. Gabor filters are widely used as a tool for feature extraction from face image and robust face recognition [4]. The gabor wavelet transform possesses multi-orientation and multi-resolution properties which makes it a popular method for feature extraction [24]. They represent complex band pass filters which are useful in extraction of multiresolutional, spatially local features of a confined frequency band [5].The shape of the gabor filters are similar to the shape of the receptive field of neurons in the primary visual cortex of the mammals [12]. Gabor filters are capable of exploiting many visual properties like orientation selection, spatial frequency characteristics and spatial localization [22, 23]. Among all the work which includes the application of gabor filter, texture representation and face recognition are the most observable applications [24]. The main purpose of using gabor filters is to solve the challenges of PCA mentioned in section 2.3.3.3. The gabor filter takes a raw image, generates gabor filter response and convert the raw image into properly aligned and constantly illuminated image.

In general the family of 2D Gabor filters can be defined in the spatial domain in the following manner. [4, 6, 7, 8, 9]:

$$\psi_{u,v}(a,b) = \frac{f_u^2}{\pi \mathrm{K}\eta} \mathrm{e}^{-\left((f_u^2/k^2)a'^2 + (f_u^2/\eta^2)b'^2\right)} \mathrm{e}^{j2\pi f_u a'} \tag{9}$$

Where $a' = acos\theta_v + bsin\theta_v$, $b' = -asin\theta_v + bcos\theta_v$ , $f_u = f_{max}/2^{(u/2)}$ and $\theta_v = v\pi/8$. Each gabor filter represents a gaussian kernel function modulated by complex plane wave whose center frequency and orientation are given by $fu$ and $\theta v$, respectively. The parameter $\kappa$ and $\eta$ determine the ratio between center frequency and size of Gaussian envelope. Though we can have different values for above mentioned parameters determining characteristics of the filters, the most common parameters used for face recognition are $\kappa = \eta = \sqrt{2}$ and $f$max = 0.25 [6,7,8,9]. When using gabor filters for facial feature extraction, researchers typically construct a filter bank featuring filter of five scale and eight orientation (comprised of 40 filters as shown in Fig. 2.8), that is u = 0,1…...p-1 and v = 0,1…,r-1,where p = 5 and r = 8 [4].

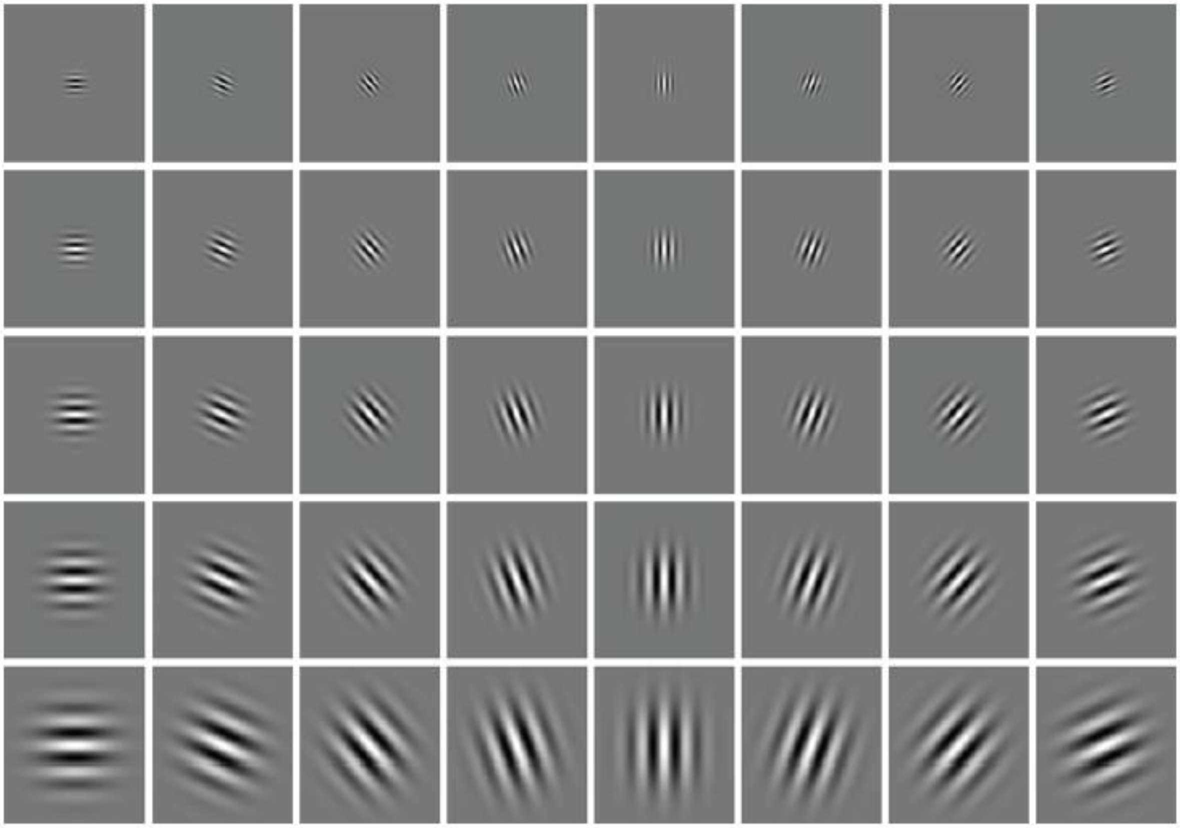


Figure 2.8 The real parts of the gabor filter bank commonly used for feature extraction in the field of face recognition [4].

The below figure i.e figure 2.9 shows the conversion of an original image into gabor filtered image using gabor filter [3].

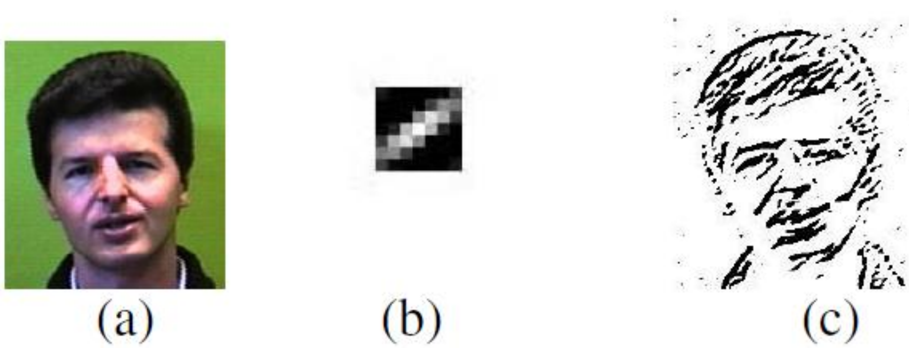


Figure 2.9. (a) Original image. (b) Gabor Filter. (c) Gabor Filtered Image [3].

### 2.3.3.5 Principal Component Analysis of gabor filter

First we need to apply gabor filter to the raw images to generate gabor filter response and then these gabor filter response act as input to the PCA. The whole process till now is described in the figure 2.10 below. Gabor filter provides robustness against varying contrast and brightness. It can also represent characteristics of local face area.

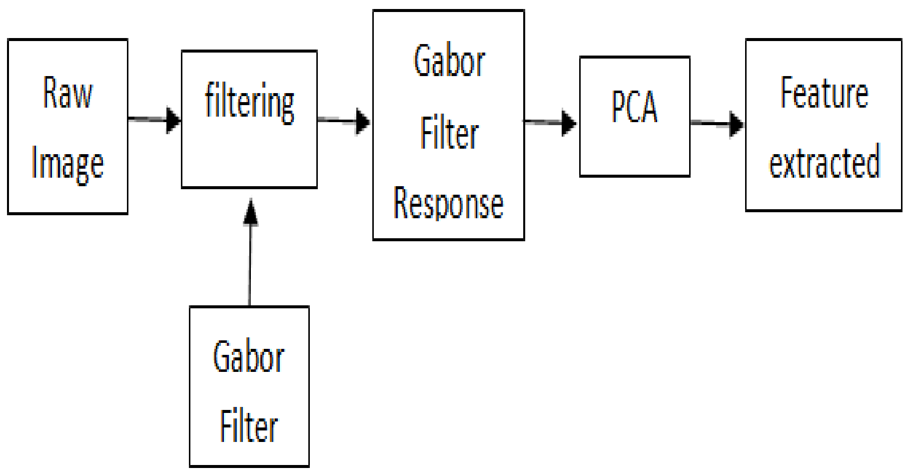



Figure 2.10. Features extraction

Let D(a, b) be a grey scale image of size m x n pixels and let $\psi_{u,v}(a,b)$ denote a gabor filter given by its center frequency fu and orientation θv. The filtering operation of the given face image D(a, b) can be defined as [4, 6, 7, 8, 9]

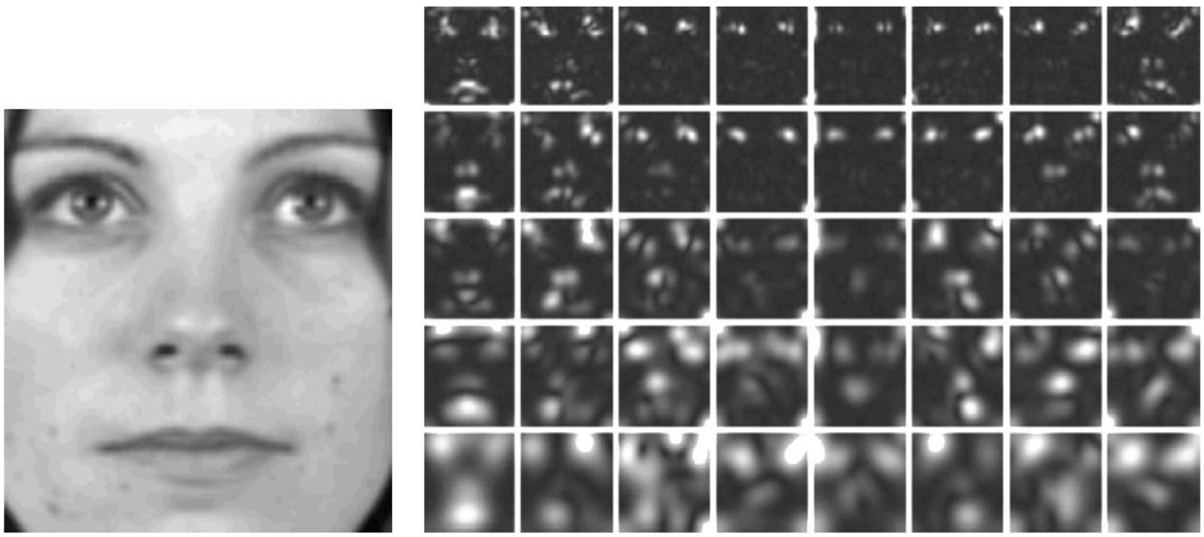
$$M_{u,v}(a,b) = D(a,b) * \Psi_{u,v}(a,b) \quad (10)$$

Where $M_{u,v}(a,b)$ denote the gabor filter response which can be decomposed into real ($E_{u,v}(a,b)$) and imaginary part ($O_{u,v}(a,b)$).

$$E_{u,v}(a,b) = \mathrm{Re}[M_{u,v}(a,b)] \quad (11)$$

$$O_{u,v}(a,b) = \mathrm{Im}[M_{u,v}(a,b)]$$

When deriving the gabor response for a image the first step is to construct the gabor filter bank as we discussed in section 2.3.3.4. Most of the techniques which involves the application of gabor filter adopt a filter bank of eight orientations ( v = 0,1………,7) and five scales (u = 0,1…...4). The figure 2.11 below describes the output obtained after applying gabor filter on the input image [4].



(a) (b)

Figure 2.11 An example of gabor magnitude output: a sample image (a) & the magnitude output of the filtering operation with the entire gabor filter bank of 40 filters (b) [4].

## 2.3.4 Neural Networks

Artificial neural networks are a popular tool in face recognition. They have been used in pattern recognition and classification. Many different methods based on neural network have been proposed since then. Some of these methods use neural networks just for classification. There are methods which perform feature extraction using neural networks.

For example, Intrator proposed a hybrid or semi-supervised method [64]. They combined unsupervised methods for extracting features and supervised methods for finding features able to reduce classification error. They used feed-forward neural networks (FFNN) for classification. They also tested their algorithm using additional bias constraints, obtaining better results.

They also demonstrated that they could decrease the error rate training several neural networks and averaging over their outputs, although it is more time-consuming that the simple method.

### 2.3.5 Neural Networks with Gabor filters

Bhuiyan proposed a neural network method combined with Gabor filter in 2007 [65]. Their algorithm achieves face recognition by implementing a multilayer perceptron with back-propagation algorithm. Firstly, there is a pre-processing step. Every image is normalized in terms of contrast and illumination. Noise is reduce by a "fuzzily skewed" filter. It works by applying fuzzy membership to the neighbor pixels of the target pixel. It uses the median value as the 1 value membership, and reduces the extreme values, taking advantage from median filter and average filter. Then, each image is processed through a Gabor filter. The representation of gabor filter is done using a complex sinusoidal signal modulated by a Gaussian kernel function. The Gabor filter has five orientation parameters and three spatial frequencies, so there are 15 Gabor wavelets.

To each face image, the outputs are 15 Gabor-images which record the variations measured by the Gabor filters. The first layer receives the Gabor features. The number of nodes is equal to the dimension of the feature vector containing the Gabor features. The output of the network is the number of images the system must recognize. The training of the network, the back propagation algorithm, follows this procedure:

1. Initialization of the weights and threshold values.

2. Iterative process until termination condition is fulfilled:

(a) Activate, applying input and desired outputs. Calculate actual outputs of neurons in hidden and output layers, using sigmoid activation function.

(b) Update weights, propagating backwards the errors.

(c) Increase iteration value.

Although the algorithms main purpose is to face illumination variations, it shows a useful neural network application for face recognition. It could be useful with some improvements in order to deal with pose and occlusion problems.

# Chapter 3
# Optimization techniques for face recognition

In this chapter we explain the various nature inspired optimization techniques used in face recognition. This chapter contains discussion on optimizing techniques like Ant Colony Optimization (ACO) and its use in face recognition, Particle Swarm Optimization (PSO) and its use in face recognition, hybrid ACO/PSO and Biogeography Based Optimization (BBO). These techniques help us finding the solution to very complex problems.

## 3.1 Ant Colony Optimization (ACO)

Ant colony optimization (ACO) can be defined as a metaheuristic which is population-based and can be applied on difficult optimization problems to generate approximate solutions.

In ACO, there exists a collection of software agents known as artificial ants. These ants try to obtain optimal solutions for the given optimization problem. In order to use ACO on the problem, the optimization problem should be converted into the problem of finding the best path on a weighted graph. The artificial ants move on the graph and build build solutions incrementally. The process of obtaining solution is stochastic and is biased by a pheromone model, i.e, a set of parameters like edges or nodes and whose values gets modified dynamically by the software agents or ants [66]. In this section we first describe ACO, then its adaptation in face recognition.

### 3.1.1 ACO algorithm

Set parameters, initialize pheromone trails

SCHEDULE_ACTIVITIES

ConstructAntSolutions

DaemonActions {optional}

UpdatePheromones

END_SCHEDULE_ACTIVITIES

The algorithm consists of an initialization step and of three algorithmic components whose activation is regulated by the SCHEDULE_ACTIVITIES construct. This construct is repeated until a termination criterion is met. Typical criteria are a maximum number of iterations or a maximum CPU time.

The SCHEDULE_ACTIVITIES construct does not specify how the three algorithmic components are scheduled and synchronized. In most applications of ACO to NP-hard problems however, the three algorithmic components undergo a loop that consists in (i) the construction of solutions by all ants, (ii) the (optional) improvement of these solution via the use of a local search algorithm, and (iii) the update of the pheromones [66].

### 3.1.2 ACO in face recognition

We have explored the research in which an efficient feature selection algorithm has been developed using ACO [16]. ACO algorithm is a nature inspired algorithm which tries to adopt ant's social behavior. In the proposed algorithm, as heuristics length of selected feature vector and classifier performances are adopted. Such that selection for optimal feature subset can be made without the prior information about features.

In order to use ACO, it is required to describe the problem in terms of a graph. Here features can be represented as nodes, with the edges between any two nodes representing the choice of the next feature. The optimal feature subset can be obtained by a simple ant traversal through on the graph which covers the minimum number of nodes that satisfies the traversal stopping criterion. The overall process can be seen in the below figure.

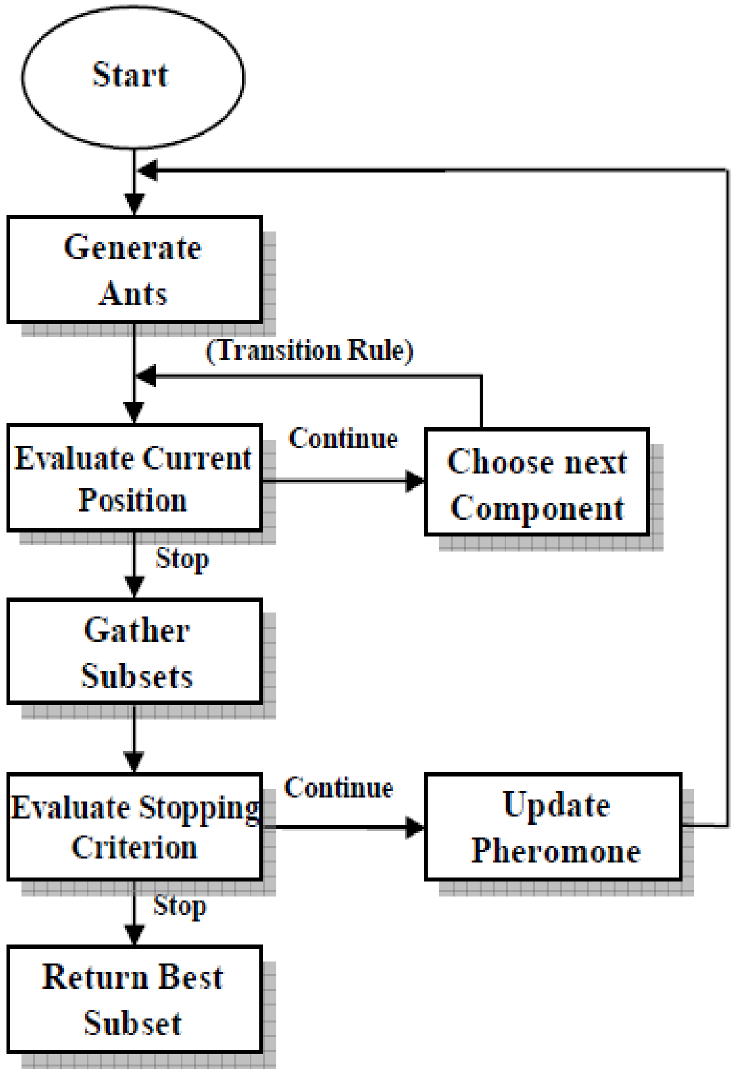



Figure 3.1 ACO-based feature selection overview [16]

## ACO based feature selection algorithm

The main steps involved in the ACO based feature selection algorithm are as follows [16]:

1) Initialization

- Determine the population of ants (p).
- Intensity of pheromone trial belongs to any feature is set.
- Determine the maximum of allowed iterations (k)

2) Ants are generated and each of them is evaluated

- Any ant ( Ai , i=1:p) randomly is assigned to one feature and it should visit all features and build solutions completely. In this step, the evaluation criterion is Mean Square Error (MSE) of the classifier. If any ant could not decrease the MSE of the classifier in first three steps of the algorithm then it gets exit after completing its work.

3) Selected subset of each ant is evaluated

- In this step, classifier performance is considered to evaluate the importance of the selected subset. Then we sort the subsets using their MSE and some of them are selected according to ACO and $AS_{rank}$ algorithms.

4) Check the stop criterion

- Exit in case, the number of iterations is greater than the maximum allowed iteration otherwise continue.

5) Pheromone updating

- Pheromone intensity is updated for features which are selected in the step 3.

6) Go to 2 and continue

## 3.2 Particle Swarm Optimization (PSO)

Particle swarm optimization (PSO) is a population-based stochastic optimization technique modelled on the social behaviors observed in animals or insects, e.g., bird flocking, fish schooling, and animal herding [67]. First, Russell Eberhart and James Kennedy proposed PSO in 1995 [67]. Since its inception, PSO has gained increasing popularity among researchers and practitioners as a robust and efficient technique for solving difficult optimization problems. In PSO, each particle of a swarm is itself a potential solution, which move through search space as described in the problem looking for an optimal solution. Each particle broadcast its information regarding current positions to neighboring particles. Now, each particle’s position is adjusted according to its velocity (i.e., rate of change) and the difference between its current position, respectively the best position obtained by its neighbors, and the best position found by the particle on its own. As the model proceeds with iteration, the swarm emphasizes more and more on the search space area containing high-quality solutions. PSO has close ties to artificial life models. Early works by Reynolds on a flocking model Boids, and Heppner's studies on rules governing large numbers of birds flocking synchronously, indicated that the emergent group dynamics such as the bird flocking behavior are based on local interactions [67].

### 3.2.1 PSO algorithm

In PSO, the velocity of each particle is modified iteratively by its personal best position (i.e., the best position found by the particle so far), and the best position found by particles in its neighborhood. As a result, each particle searches around a region defined by its personal best position and the best position from its neighborhood. Henceforth we use vi to denote the velocity of the ith particle in the swarm, xi to denote its position, pi to denote the personal best position and pg the best position found by particles in its neighborhood. In the original PSO algorithm, vi and xi, for i = 1, . . . , n, are updated according to the following two equations :

$$vi \leftarrow vi + \phi 1 \odot (pi - xi) \odot \phi 2 \, (pg - xi) \tag{12}$$

$$xi \leftarrow xi + vi \tag{13}$$

where $\phi 1$ = c1R1 and $\phi 2$ = c2R2. R1 and R2 are two separate functions, each returning a vector comprising random values uniformly generated in the range [0,1]. c1 and c2 are acceleration coefficients. The symbol $\odot$ denotes pointwise vector multiplication. Equation (13) shows that the velocity term vi of a particle is determined by three parts, the "momentum", the "cognitive", and the "social" part. The "momentum" term vi represents the previous velocity term which is used to carry the particle in the direction it has travelled so far; the “cognitive” part, $\phi 1 \odot (pi - xi)$, represents the tendency of the particle to return to the best position it has visited so far; the “social” part, $\phi 2 \odot (pg - xi)$, represents the tendency of the particle to be attracted towards the position of the best position found by the entire swarm. Position pg in the “social” part is the best position found by particles in the neighborhood of the ith particle.

### 3.2.2 PSO in face recognition

A new efficient face recognition method based on the combination of Neural Networks and Principal Component Analysis (PCA) [15]. This method uses Principal Component Analysis (PCA) to extract principal eigenvectors to obtain the best features that can describe the complete image, hence to feed the minimum number of inputs of neural networks. After this, these reduced image data are input into a feed forward neural network, need to be trained. Then we try to optimize the weights of associated with neural networks using Particle Swarm Optimization (PSO) algorithm. Then we use sample images from face database to test this trained network.

BP Neural Network is a multi-layered feed forward neural network having good adaptability, identification and classification. A simple three-layered BP network consists of input layer, hidden layer and output layer.

The main procedure of this PSO-BP algorithm can be described as: First, the position vector of the particles is represented by the thresholds and weights of the BP neural network. Then we apply PSO to search for the optimal position in order to minimize the Mean Square Error to its minimum. PSO stops when convergence rules are met. At this point, the thresholds and the initial weights of neural network are represented by each dimension of the optimal position vectors. Finally, BP neural network trains itself using the traditional learning method and the values obtained. The process stops once the convergence rules gets satisfied. Figure below shows the specific steps PSO-BP algorithm.

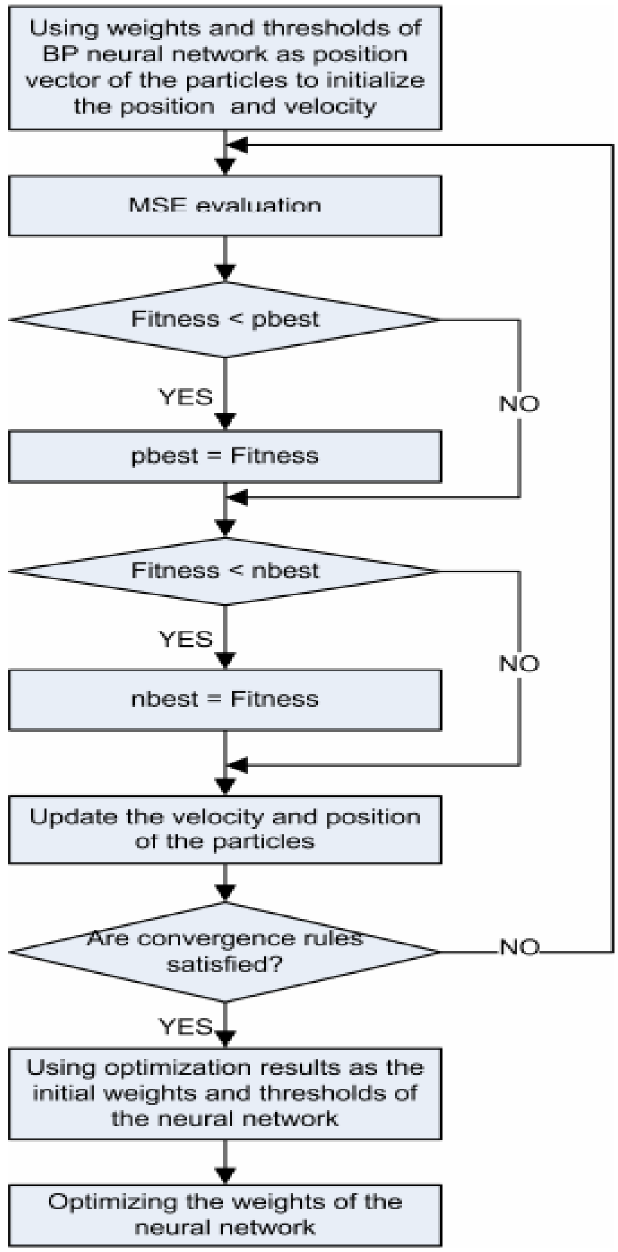



Figure 3.2 PSO-BP algorithm [15]

## 3.3 Hybrid ACO/PSO

Nicholas and Freitas proposed several modifications to the original PSO/ACO algorithm [68]. It involves the changes in the splitting of the rule discovery process into two separate phases. First phase tries to discover a rule using nominal attributes. Second phase tries to extend the discovered rule with continuous attributes. Due to this the ability of the PSO/ACO algorithm to treat nominal and continuous attributes gets increased in many ways. Both the new modified version PSO/ACO2 and the original PSO/ACO algorithm uses a sequential covering approach to discover one classification rule at a time.

### 3.3.1 Hybrid ACO/PSO algorithm

Sequential covering approach used by the hybrid PSO/ACO2 algorithm given by Nicholas and Freitas:

- RuleSet is empty(Φ) initially
- For Each class of cases Trs = {All training cases}
- While (Number of uncovered training cases of class A > Maximum uncovered cases per class)
- Find best nominal rule by applying the PSO/ACO algorithm
- Add continuous terms to Rule by running the standard PSO algorithm, and return the best discovered rule BestRule
- Discovered BestRule is pruned
- RuleSet = RuleSet U BestRule
- Trs = Trs – {training cases correctly covered by discovered rule}
- End of while loop
- End of for lop

- Order these rules in RuleSet by descending Quality

It is necessary to estimate the quality of every candidate rule (decoded particle). A measure must be used in the training phase in an attempt to estimate how well a rule will perform in the testing phase. The quality measure used in this algorithm is given by following equations:

For original Quality measure

$$\text{Sensitivity} \times \text{Specificity} = TP / (TP + FN) \times TN / (TN + FP) \qquad (14)$$

Later equation 15 is modified as:

For Quality measure on Minority class

$$\text{Sensitivity} \times \text{Precision} = TP / (TP + F7) \times TP / (TP + FP) \qquad (15)$$

This is also modified with using Laplace correction as:

New quality measure on Minority Class

$$\text{Precision} = 1 + TP / (1+ k + TP + FP) \qquad (16)$$

PSO/ACO2 takes the best nominal rule built by PSO/ACO2 and then attempts to add continuous attributes using a standard PSO algorithm [68].

### 3.3.2 Hybrid ACO/PSO in facial expression recognition

A novel method for facial expression recognition in three dimension has been devised which is motivated by the advancement of particle swarm optimization ant colony and (PSO and ACO respectively) in the area of data mining [14]. First, we use a generic 3D model of a face to establish anatomical correspondence between faces. In order to match the facial surfaces, we have deformed the 3D face model. We build classification rules using algorithm based on ACO/PSO and then apply them while using surface points as basis for classification. A recognition rate of 92.3% was achieved using this algorithm [14].

We use ACO and PSO to discover classification rules which are then expressed in the form of IF-THEN statements:

IF<term1 AND ... termM> THEN<class>

where the terms in the rule i.e antecedent (IF part) are used to represent triples of the form <attribute, operator, value> and the class in the rule i.e represent consequent (THEN part) is one of the six possible expressions. The way in which these rules are stored is in the form of a sequentially accessed list and the first valid rule is used to predict the class to which expressions are classified. When there is no valid rule then we classify expression to a default class. The hyperspace of attributes are divided into a set of hypercubes and classify each attribute vector is classified to the class with which the attribute vector belongs to.

ACO and PSO involvement comes in the construction of rule antecedent. ACO can be applied to the problem which can be represented in the form of finding shortest path on a graph. The solution to the problem is discovered by "ants" or "agents" who lay a pheromone trail while traversing through the nodes present in the graph. When an ant gets a choice between many nodes, it chooses the node with the largest amount of pheromone. Because shortest edges can be traversed fast, more ants will pass through them leaving behind high collection pheromone.

The limitation of ACO is that it can be applied to discrete attributes which results in loss of order that continuous attributes possess and can be vital for classification. To overcome this shortcoming of ACO we use PSO to handle continuous data.

In PSO architecture, the rule antecedent can be represented by a particle moving in a high dimensional attribute space. To define the movement of a particle we use, its own experience, i.e the best position it has found so far, and by using the best position found by its most successful neighbor or we can say the best local position found so far. During rule construction, the antecedent part is converted to a vector consisting of two dimensions per attribute, one for the lower bound and one for the upper. We allow many particles to move through the search space as per problem and this ends when they converge to an optimum position/rule. The rule gets added to the rule list just after it gets pruned and we remove the samples covered by it. Using this approach we find the rest of needed rules [14].

The database used to evaluate the above explained algorithm is proposed algorithm is BU-3DFEDB database which comprises of 44 male and 56 female subjects. These subjects use four levels of intensity, low, middle, high and highest to show the six universal expressions.

## 3.4 Biogeography Based Optimization (BBO)

Biogeography based optimization (BBO) is a nature inspired algorithm based on biogeography. Biogeography is the study of geographical distribution of biological organisms. The study of biogeography can be linked to the work done by Alfred Wallace [25] and Charles Darwin [26]. Robert MacArthur and Edward Wilson build first mathematical model for biogeography in 1960, which was focused on how species are distributed in neighboring islands [27]. Mathematical models for biogeography are capable of describing the following [11]:

- Migration of species from one island to another.
- Arise of new species
- Extinction of species

An island is same as a habitat for biological species which is geographically separated from other habitats. Habitats that have favorable conditions for the species are expected to have high suitability index (HSI) [28]. Factors on which HSI depends are:

- Rainfall
- Diversity of vegetation
- Diversity of topographic features
- Land area

- Temperature

The variables which are used to characterize habitability are known as suitability index variables (SIVs). One major difference between HSI and SIV is that SIV is the independent variables of the habitat, and HSI is the dependent variable [11].

Habitats with low HSI value has smaller number of species as compared to the habitats with high HSI value which tend to have a large number of species. Habitats having high HSI have species that emigrate to other habitats because of large number of species they possess. We can also state that the habitats with high HSI tend to have a low immigration rate of species because they already possess a large number of species and are nearly saturated. Therefore, habitats having high HSI value are stable than habitats having low HSI value. Also, habitats with high HSI value have high emigration rate but that doesn't mean species completely disappears from it. On the other hand, habitats with low HSI tend to have a high immigration rate because of less population they have. New species immigrate to habitats with low HSI and increase the HSI of the habitat because HSI of a habitat is directly proportional to the variety of species exists on it. However, if the HSI of the habitat is still low then the species that reside there will tend to go extinct, which will further encourage immigration of species. Because of this habitats with low HSI are more dynamic in nature than habitat with high HSI value [11].

BBO can be applied to problems in the field of engineering, medicine, economics, sports, etc, as long as we are able to quantify the suitability of a given solution. A good solution is similar to a habitat with high HSI and a poor solution is similar to the habitat with low HSI. The habitat with high HSI share their features with the habitat having low HSI and this addition of features to the poor HSI tend to increase their HSI value of the habitat. This approach to problem solving is known as biogeography based optimization (BBO) [11, 13].

### 3.4.1 Biogeography

Figure 3.3 below describes a model in which a lot of species are present on a single habitat [27]. The emigration rate and immigration rate are denoted by μ and $\lambda$ respectively.

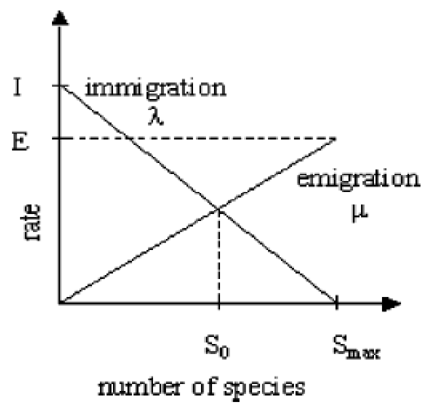



Figure 3.3 species model of a single habitat based on [27]

These rates are the functions of total number of species present in the habitat. According to the immigration curve shown in the figure 3.3 the habitat 'I' witnesses maximum possible immigration rate $\lambda$ when no species is present on the habitat. With the increase in number of species the immigration rate decreases. This is because with the increase in species habitat gets crowded. $S_{max}$ is the maximum number of species that can be supported by the habitat, at which the immigration rate attains value zero. On the other hand, considering the emigration curve, if there are no species present in the habitat then the emigration becomes zero. With the increase in the number of species present in the habitat the emigration rate also increases. This is because more species will be available to leave the habitat and find other habitats to live. The maximum attainable emigration rate is 'E' when the habitat contains the maximum possible number of species. The equilibrium in number of species is reached at $S_0$ at which both emigration rate and immigration rate becomes equal. Figure 3.4 below shows two candidate solutions to some problem. $S_1$ is a relatively poor solution and $S_2$ is relatively good solution [3, 11].

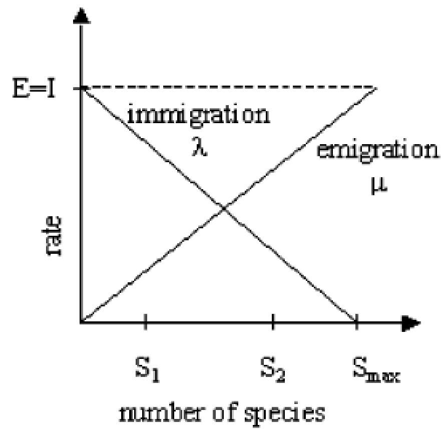



Figure 3.4 Illustration of two candidate solution to same problem

### 3.4.2 Migration

Suppose if some problem is there whose possible solutions can be expressed as vectors of integers. If this is the case then each integer present in the solution can be considered as an SIV. Further let us suppose we have some way of measuring the goodness or fitness of the solution. Good solutions can be assumed as habitats having high HSI, and those that are poor can be assumed as habitats having low HSI. We assume that each solution has same species curve, but the S value represented by the solution is dependent on its HSI. $S_1$ in figure 4.2 indicates a low HSI solution where as $S_2$ on the other hand indicates a solution with high HSI. The immigration rate for $S_1$ is higher than the immigration rate for $S_2$. On the other hand, emigration rate for $S_1$ is lower than the emigration rate for $S_2$. With the probability $P_{mod}$ each solution can be modified based on other solution. If a given solution gets selected for modification, then its immigration rate $\lambda$ for making choice , whether to modify each SIV present in the solution or not. If a given solution gets selected for the modification then we can use the emigration rates of other solution to make our choice which of the features of the solution migrate the randomly selected SIV to the selected solution.

The migration strategy presented in BBO is very much similar to the strategy presented in breeder GA [29] and evolutionary strategies [30] in which a single offspring is resulted from the contributions of many parents. One difference between evolutionary strategies and BBO is that in case of evolutionary strategies new solutions are created using global recombination, while migration strategy used in BBO is applied to change the existing solutions only. BBO migration is used to modify existing habitats to improve their fitness [11]. The migration process is given below [11].

```
Select H_i with probability ∝ λ_i
If H_i is selected
    For j = 1 to n
        Select H_j with probability ∝ μ_i
        If H_j is selected
            Randomly select an SIV σ from H_j
            Replace a random SIV in H_i with σ
        end
    end
end
```

Where H denotes habitat, $\lambda$ is immigration rate and μ is emigration rate.

### 3.4.3 Mutation

A habitat's HSI can be changed because of occurrence of some random events. We can modulate this scenario as SIV mutation. We can use probability for the count the number of species to calculate mutation rates. If we observe the equilibrium point shown in the figure 4.2, we see that low species count and high species count both have relatively low priorities. Each population has some probability attached with it which indicates the possibility of being a solution for the given problem. Solutions having high HSI and solutions having low HSI are both unexpected to be solution. Medium HSI solutions are likely to be the solution as compared to high HSI and low HSI solutions.

If a given solution S has low probability $P_s$ then there is very little chance that it may exist as the required solution because it is likely to mutate into some other solution. On the other hand, if a given solution has a high probability then it is less likely to mutate to some other solution. The mutation rate can be as inversely proportional to the solution probability and given by equation below.

$$m(S) = m_{max} \left( \frac{1 - P_s}{P_{max}} \right) \quad (17)$$

Where $m_{max}$ is a user defined parameter.

This mutation scheme tends to increase diversity among the population. The mutation process is given below [11].

For $j = 1$ to $m$
    Use $\lambda_i$ and $\mu_i$ to compute the probability $P_i$
    Select SIV $H_i(j)$ with probability $\propto P_i$
    If $H_i(j)$ is selected
        Replace $H_i(j)$ with a randomly generated SIV
    end
end

Where H denotes habitat, $\lambda$ is immigration rate and μ is emigration rate.

### 3.4.4 BBO Algorithm

1. Initialize the parameters of the BBO. This means reiterating the problem in terms of SIVs

and habitat which depends on the problem. Also initialize the maximum species count, maximum migration rate and maximum mutation rate.

2. Initialize some habitats to begin with. These habitats denote the possible solutions for the problem.
3. For each initialized habitat a mapping is to be made between HSI and few other parameters like emigration rate (denoted by $\mu$), number of species (denoted by S) and immigration rate (denoted by $\lambda$).
4. Use emigration and immigration probabilistically to modify each of the the habitat and again compute their HSI.
5. Update the probability of species count of each habitat. Now mutate each habitat and compute the HSI associated with it again.
6. Go to step 3. Going to step 3 will set a loop which can be ended after a fixed number of generations or when a solution, pretty much close to the expected solution has been found.

Note that during the entire algorithm we have been modifying each habitat in steps 2, 4 and 5. So, when the habitat gets modified on the said steps then we have to again check the feasibility of the habitat as the problem solution again.

If after performing the above mentioned steps it does not represent a feasible solution then some methods need to be implemented in order to map it to the set of feasible solutions.

### 3.4.5 Dissimilarities between BBO and other Population Based Optimization Algorithms

BBO has some of the distinct features. Though BBO is a population based approach for problem optimization but it does not generate or reproduce children. This feature of BBO differentiates it from other evolutionary strategies and GAs.

BBO is clearly different from Ant Colony Optimization (ACO). In case of ACO, in each iteration a fresh set of solutions are produced. On the other hand, BBO never generates a new set solutions, it only maintains its pre-existing set of solutions while moving on from one iteration to another. BBO make use of migration to adapt these solutions probabilistically.

BBO is very much similar to approaches like Particle Swarm Optimization (PSO) and DE. In these approaches, solutions are preserved while moving from one solution to another. Each solution learns from its neighbors and adapts itself with the flow of algorithm. Each solution in case of PSO is represented as a point in space and change with the time in the solution is represented as a velocity vector. However, in case of PSO solutions do not change directly but their velocities do change and this change effects the solution and the solution gets changed. DE, which is not biologically inspired changes its solution directly. The changes in a particular solution of DE are based on differences between other DE solutions. So unlike PSO and DE BBO solutions tends to change directly with the help of migration from other solutions or habitats. We can say that BBO solution directly share their features with other solutions rather good solutions. This helps in improving the quality of the solutions.

It is because of these dissimilarities between BBO and other population based optimization methods that has proven to be the strength of BBO. But some open research areas are still present in the form of questions like how these dissimilarities differ BBO's performance from other population based approaches.

## 3.5 BBO for feature extraction using DCT

The foremost step in any face recognition system is of extracting features from the given image. A feature extraction algorithm usually builds some model after extracting the features from the images. The model is generally build using some linear or non linear transform of data present in the images. An efficient method of feature extraction using BBO with DCT has been proposed [10]. The BBO based approach has been applied to features extracted by DCT technique. The Discrete Cosine Transform, DCT-II standard (often called simply DCT) represents a series of data points as a sum of mathematical cosine functions oscillating with different frequencies [10, 62]. DCT is a popular transformation method commonly used in digital signal processing because of its ability known as energy compaction. Therefore, it can be used to transform images, compacting the variations, allowing an effective dimensionality reduction. When a DCT is performed over an image, the energy is compacted in the upper-left corner.

Let C is the DCT of the image$D_{NxM}$.

$$Cpq = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} D_{mn} a_p a_q \cos\frac{\Pi(2m+1)p}{2M} \cos\frac{\Pi(2n+1)q}{2N} \tag{1}$$

where

$$a_p = \begin{cases} \frac{1}{\sqrt{M}}, p = 0 \\ \sqrt{\frac{2}{M}}, 1 \le p \le M-1 \end{cases}$$

$$a_q = \begin{cases} \frac{1}{\sqrt{N}}, q = 0 \\ \sqrt{\frac{2}{N}}, 1 \le q \le N-1 \end{cases}$$

where M is the row size and N is the column size of A. We can truncate the matrix B, retaining the upper-left area, which has the most information, reducing the dimensionality of the problem. where M is the row size and N is the column size of A. We can truncate the matrix B, retaining the upper-left area, which has the most information, reducing the dimensionality of the problem.

After extracting the features using DCT we still need minimal set of features from which a face image can be recognized. For this reason, a biogeography based feature selection algorithm has been proposed which is described below [10].

1. **Extracting Features**: Apply Discrete Cosine Transformation to image and get the DCT Array.
2. From upper left corner of DCT Array obtained before take the most desirable features of size mxm.
3. **Selecting Features**: Use the BBO algorithm to obtain the feature subset of the extracted features.
4. Pick up the habitat H with highest HSI value. The best feature subset of the features defined in step 2 is represented by SIVs of this habitat.
5. **Classification**: calculate the difference between the feature subset (obtained in step 4) of each image of facial gallery and the test image with the help of Euclidean Distance defined as classifier. The required index would be the index of the image which has the smallest distance with the image under test.

The above proposed algorithm shows the application of BBO for feature extraction in face recognition. We have also used the BBO in our approach but instead of using DCT our work is PCA focused. Our work also highlights the use of gabor filter, which we have used to overcome the shortcoming of PCA by providing properly aligned and uniformly illuminated images as input to PCA.

# Chapter 4
# Proposed Approach

In this chapter, we are going to explain the model, we considered for the face recognition problem. We will explain the model and the approaches we followed to solve the face recognition problem in an efficient manner. We will also explain the parameters used in our algorithm and how we adapted the Biogeography Based Algorithm (BBO) to solve the face recognition problem.

## 4.1 Model for Face Recognition

In our model shown below, we have used a combined approach to solve the face recognition problem. Our Combined approach consists of Principal Component Analysis (PCA), Gabor Filters and Biogeography Based Optimization (BBO).

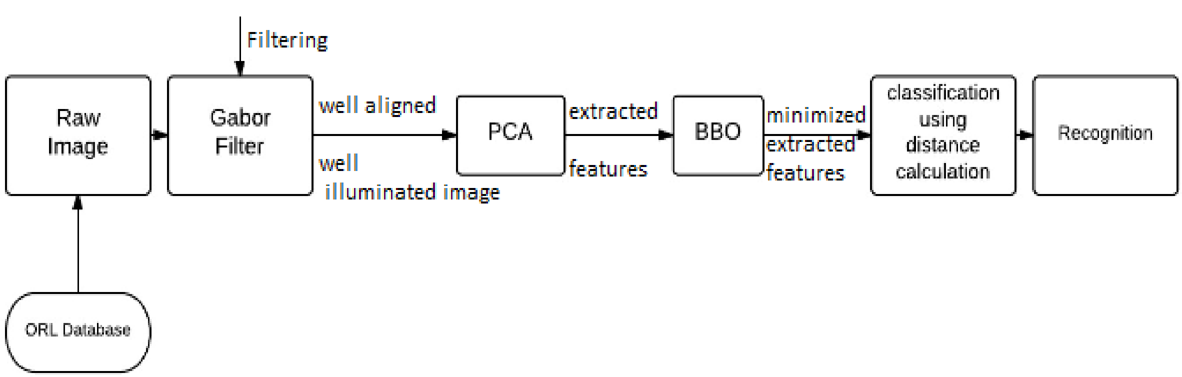



Figure 4.1 Our Model for Face Recognition

Our process for face recognition is shown by the above model. The main steps of this model are as follows:

1. We fetch the raw images from the ORL database, stored in ".pgm" format.
2. We apply gabor filter to the fetched images in order to make them properly aligned and constantly illuminated. This step overcomes the limitation of PCA to get always properly aligned and constantly illuminated images as input.
3. These properly aligned and constantly images are given as input to the PCA. PCA extracts the desirable features from the images.
4. Now we apply BBO to the features extracted by PCA using the two approaches, discussed later in this chapter. BBO further minimizes the extracted features obtained from PCA.
5. Finally we classify the images and perform recognition from the test data.

### 4.1.1 Principal Component Analysis (PCA)

Extraction of features from the images is the foremost step in face recognition. PCA is a standard technique used for feature extraction. The image in the form of A x A matrix can be expressed as a point in the space A x A dimensions. The goal of PCA is to find the required vectors that can represent the image information and form another space. PCA steps are as follows:

1. Let the training set of face images be $\Gamma_1, \Gamma_2, \Gamma_3, \dots \dots \dots, \Gamma_p$.
2. We calculate the average face of the set by $\psi = \frac{1}{p}\sum_{r=1}^{p}\Gamma_r$.
3. We calculate the difference of each face from the average by vectors $\phi = \Gamma_i - \psi$.
4. We calculate eigenvalues and eigen vectors from covariance matrix as shown (1) and a new face image is transformed into its face components by this operation [2].

$$c = \frac{1}{p}\left[\sum_{R=1}^{p} \emptyset_R {\emptyset_R}^T\right] \qquad (7)$$

5. We calculate weights using (2) and from them we form a vector $\Omega = \omega_1, \omega_2, \ldots.\omega_p$ that represents the contribution of each eigenvector in representation of input face image.

$$\omega_k = \mu_K^T(\Gamma - \psi), \text{ k} = 1,2,\ldots\ldots\text{P} \qquad (8)$$

where $\mu_k$ is eigen vector.

These weights may be used in a face classification algorithm to find which of predefined face classes that describe the face [12].

### 4.1.2 Gabor Filter

Gabor filters generally called as Gabor wavelets or kernels are complex band pass filters. They have shape similar to the shape of the cells of the visual cortex of mammalian brains. These are used in many applications such as extraction of multiresolutional, spatially local features of a confined frequency band [5]. Another important feature of the Gabor filters is that they act as an efficient tool for facial feature extraction and robust face recognition [3]. In general the family of 2D Gabor filters can be defined in the spatial domain in the following manner. [4, 6, 7, 8, 9]:

$$\psi_{u,v}(a,b) = \frac{f_u^2}{\pi K \eta} e^{-\left((f_u^2/k^2)a'^2 + (f_u^2/\eta^2)b'^2\right)} e^{j2\pi f_u a'} \qquad (9)$$

Where $a' = acos\theta_v + bsin\theta_v$, $b' = -asin\theta_v + bcos\theta_v$ , $f_u = f_{max}/2^{(u/2)}$ and $\theta_v = v\pi/8$.

The center frequency and orientation of the wave equation described above are given by $fu$ and $\theta v$, respectively. The parameter $\kappa$ and $\eta$ determine the ratio between center frequency and size of Gaussian envelope. Though we can have different values for above mentioned parameters determining characteristics of the filters, the most common parameters used for face recognition are $\kappa = \eta = \sqrt{2}$ and $f$max = 0.25 [6,7,8,9]. When using gabor filters for facial feature extraction, researchers typically construct a filter bank featuring filter of five scale and eight orientation (comprised of 40 filters),that is u = 0,1…...p-1 and v = 0,1…,r-1,where p = 5 and r = 8[4].

### 4.1.3 Principal Component Analysis of Gabor filter responses

First we need to apply gabor filter to the raw images to generate gabor filter response and then these gabor filter response act as input to the PCA. The whole process till now is shown in the Fig. 4.1. Gabor filter provides robustness against varying contrast and brightness. It can also represent characteristic of local face area.

Let D(a, b) be a grey scale image of size m x n pixels and let let $\psi_{u,v}(a,b)$ denote a gabor filter given by its center frequency fu and orientation θv. The filtering operation of the given face image D(a, b) can be defined as [4, 6, 7, 8, 9]

$$M_{u,v}(a,b) = D(a,b) * \Psi_{u,v}(a,b) \quad (10)$$

Where $M_{u,v}(a,b)$ denote the gabor filter response which can be decomposed into real ($E_{u,v}(a,b)$) and imaginary part ($O_{u,v}(a,b)$).

$$E_{u,v}(a,b) = \mathrm{Re}[M_{u,v}(a,b)] \quad (11)$$

$$O_{u,v}(a,b) = \mathrm{Im}[M_{u,v}(a,b)]$$

### 4.1.4 Selecting the optimal features using BBO

We have extracted the features using the principal component analysis of gabor filter. But for efficient face recognition, we still need to find the most optimal set of features using BBO. For this we have adapted the original BBO, proposed by Dan Simon to suit our approach [14, 10]. We will describe the algorithms in next sections.

## 4.2 Distance Calculation

After the development of train model from the training set images, we calculate the similarity matrix using the matrix which contains the features extracted from test set of images. The distance used to compute the similarity matrix is 'Euclidean'.

Euclidean distance is the straight line distance between two points present in a N-dimensional space. For example, the Euclidean distance between any two points $x_i$ and $y_i$ present in N-dimensional space is given by [10]:

$$D = \sqrt{\sum_{i=1}^{N}(x_i - y_i)^2} \tag{18}$$

In the following section we will discuss in detail about the two approaches we used for the face recognition problem. We will also discuss the parameters and the algorithms developed for the two approaches

## 4.3 Adapted BBO- first approach using migration from ideal matrix and fitness function as mean

In our approach we have adapted the BBO, originally proposed by Dan Simon in 2008 [11]. After features are extracted using gabor filter and PCA. The extracted features are further minimized using BBO. The final feature subset consists of the most optimal features. Here we have used standard mean as fitness function and migration is performed using the ideal matrix obtained from the training data.

### 4.3.1 Parameters

We have also modified certain parameters to adapt the BBO in our approach as per our problem requirements. The parameters modified are number of iterations of BBO, Suitability

Index Variable (SIV) and Habitat Suitability Index (HSI). The following table shows the modified parameters.

| | |
|---|---|
| Number of iterations of BBO algorithm | 120 |
| SIV value | Real |
| Number of SIVs in a Habitat | 120 |
| Fitness Function | Standard Mean |

Table 4.1 Parameters used in Adapted BBO first approach using migration from ideal matrix and fitness function as mean

Here SIVs take real values. We have used standard mean as a fitness function and number of SIVs is 120. We are performing iteration for each SIV.

### 4.3.2 Suitability Index Variable (SIV)

In the proposed approach the habitat represents a possible solution (optimal features). The habitat comprises of several Suitability Index Variable (SIVs). Features extracted from every single image represent a SIV.

### 4.3.3 Habitat Suitability Index and Fitness Function

Each habitat is evaluated on the basis of habitat suitability index (HSI). Habitat Suitability Index can be considered as a value return by the fitness function. A habitat is considered rich in features if its HSI value is closer to the ideal habitat's HSI value and if its value is far from the ideal HSI value then the habitat is considered as a poor habitat containing fewer features. Based on the HSI value we can always distinguish between good solution and the bad ones. In order to improve the bad solutions we can migrate certain set of selected features from the ideal solution to bad ones. We have used standard mean as the fitness function for the approach. After developing the training model we use Euclidean distance to calculate the matching score between the features of test image and that of training model.

### 4.3.4 Algorithm – I (Adapted BBO using migration from ideal matrix and fitness function as mean)

Extract the features from the training set and test set in matrices as shown in Fig. 4.1 using Gabor filter and PCA.Lets say ‘train_data’ is a matrix contains feature extracted from training set and ‘test_data’ contains features extracted from test set.

1. Calculate the HSI value for the whole train data using fitness function.It will be treated as ideal HSI.
2. Apply BBO for 120 (number of SIVs) iterations and in each iteration do the following :
   i. Calculate HSI value for the SIV (or column) of the image present in test_data matrix.
   ii. Compare the calcultaed HSI value with ideal HSI.
   iii. If calculated HSI value is close to ideal HSI do nothing.Go to next iteration.
   iv. If calculated HSI is not close to ideal HSI then perform migration to improve test_data.
3. Calculate similarity matrix from train_data and test_data using euclidean distance and plot the required curve.

The algorithm is also described in the figure below.

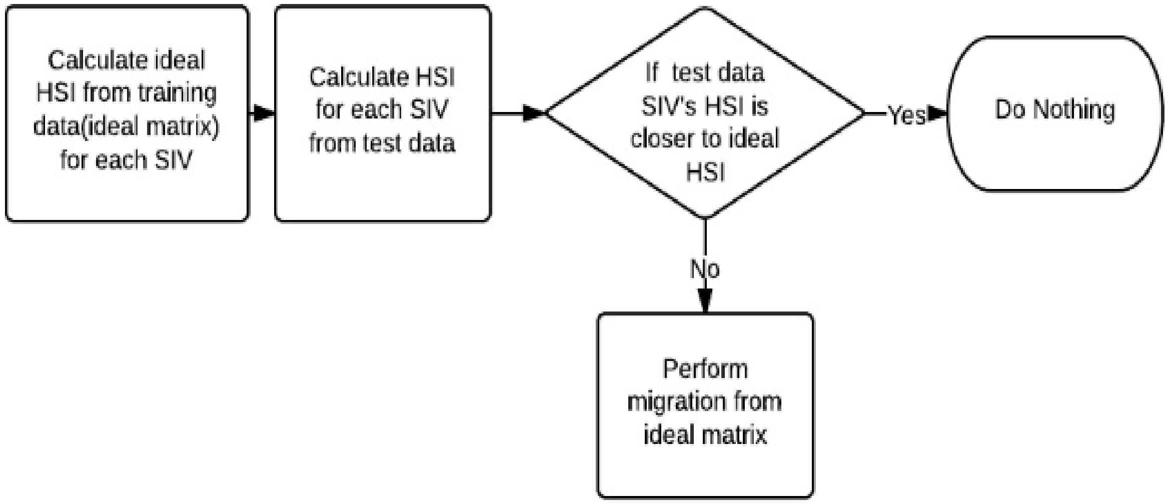



Figure 4.2 Algorithm - I

## 4.4 Adapted BBO- second approach using migration from test matrix and fitness function as standard deviation

In this approach we have we have adapted the BBO just like the first approach but here we have varied the fitness function and the matrix we use for migration. We use the test matrix for the migration. We have also modified the algorithm used in first approach.

### 4.4.1 Parameters

In this approach we have varied few parameters as shown in table 4.2 below.

| | |
|---|---|
| Number of iterations of BBO algorithm | 120 |
| SIV value | Real |
| Number of SIVs in a Habitat | 120 |
| Fitness Function | Standard Deviation |

Table 4.2 Parameters used in Adapted BBO second approach using migration from test matrix and fitness function as standard deviation.

### 4.4.2 Suitability Index Variable (SIV)

In this approach the number of SIVs and the type of value they take would remain the same. The habitat represents a possible solution (optimal features). The habitat comprises of several Suitability Index Variable (SIVs). Features extracted from every single image represent a SIV.

### 4.4.3 Habitat Suitability Index and Fitness Function

In this approach we have changed the fitness function. Instead of using simple standard mean as an approach for basis of our fitness function, we design a new fitness function based on standard deviation approach. Therefore, HSI calculations will be based on standard deviation.

### 4.4.4 Algorithm – II (Adapted BBO using migration from the test matrix and fitness function as standard mean)

Extract the features from the training set and test set in matrices as shown in Fig. 4.1 using Gabor filter and PCA. Lets say 'train_data' is a matrix contains feature extracted from training set and 'test_data' contains features extracted from test set.

1. Use the fitness function and calculate the HSI value for each SIV from the training data and do the same for the test data.
2. Calculate the absolute value of differences between the HSI value of the first SIV of test data and HSI value of first SIV of training data and store this value in an array 'lambda'. Repeat this for all remaining 119 SIVs.
3. Apply BBO for 120 (number of SIVs) iterations and in each iteration do the following :
    i. Select the SIV (out of the 3 SIV which belongs to the same person) with the highest HSI value for the migration.
    ii. If the selected SIV is the same SIV for which we are performing migration then do nothing and go to the next iteration else go to the step iii .

iii. Now probabilistically select the features from the SIV get selected above and migrate those features to the required SIV.

4. Calculate similarity matrix from train_data and test_data using euclidean distance and plot the required curve.

The main steps of the algorithm described above are shown in the figure below.

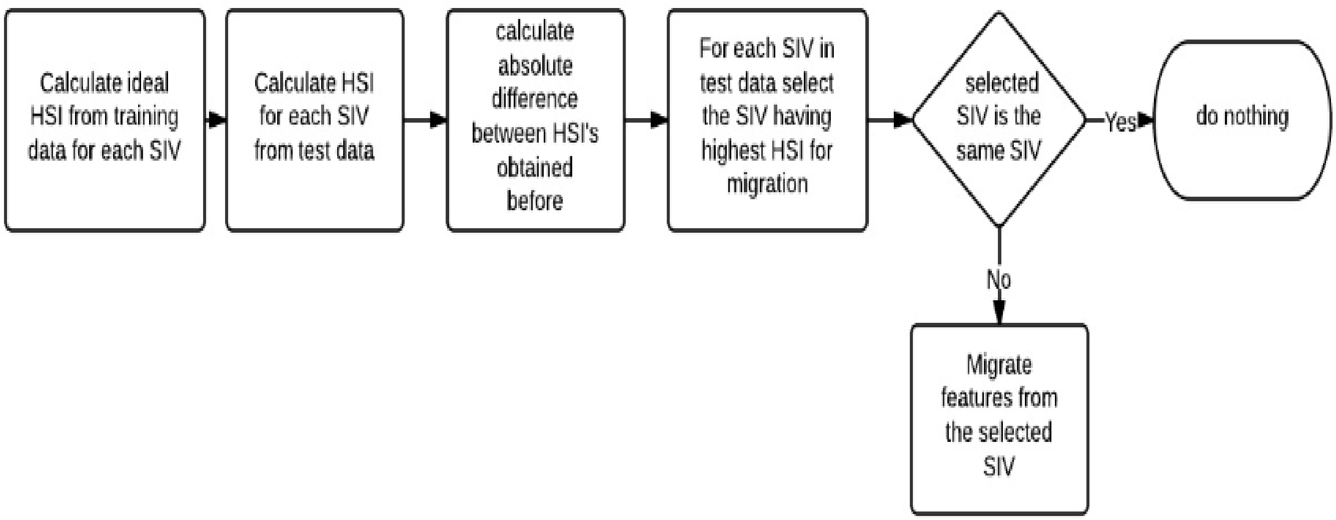



Figure 4.3 Algorithm - II

So far, we discussed the model we framed to solve the face recognition problem. We also discussed the two different approaches, we used for the face recognition problem and how we adapted BBO in these approaches differently. We changed certain parameters while adapting BBO, as per our problem requirement. We also went through the algorithms used in the two approaches. In the next chapter we will talk about the experimental setup we used and the result we obtained from the two approaches.

# Chapter 5
# Experiments and Results

This chapter explains the experimental setup used to find results for our face recognition problem. We explain the facial database we used for the face recognition problem and how we have divided the face database into training set and test set.

This chapter also contains the results we have obtained after applying our algorithm on the face database. The results are shown in the form of Receiver Operating Characteristic (ROC) curve. The results show a clear comparison between our proposed approach and the current standard approach.

## 5.1 Experimental Setup

For our biogeography based face recognition algorithm we use standard ORL database acquired at the Olivetti Research Laboratory in Cambridge, U.K. The software used to test our algorithm is MATLAB 7.9.0 (R2009b). The database consists of 400 distinct images that correspond to 40 distinct subjects. Therefore, each subject has 10 facial images each image has got different illumination, pose and facial expression. The size of each image is 92 x 112 pixels and has 8-bit grey levels [9]. Some of the images from the ORL database are shown below in Fig. 5.1.

Figure 5.1 Representative set of ORL face database

For our experiment we have used 360 images that is six images per subject. We have divided the images of every subject into two basic sets. First set (training set) contains the first three images of every subject. Second set (test set) contains the next three images of every subject. So, both training set and test set consists of 120 images each. We have made a third set (test set) which also contains 120 images. Training set trains the system and build the train model. Test set is used for assessment and calculation of certain parameters. We develop the train model from the training set by applying gabor filter and PCA and use the same techniques to extract features from test set. Both training set and features extracted from test set images are matrices of size 119 x 120. Each column represents an image or a SIV of length equal to 119. Now, we apply BBO to the features extracted from the test set and use train model as ideal habitat.

## 5.2 Performance measures

For performance analysis of our biogeography based face recognition algorithm we calculate the similarity matrices, in which each of the image vector is compared with each image vector present in the train model and client and imposter matching scores are calculated.

A client is an entity who is making a genuine identity claim whereas imposter is the one making a false identity claim. We calculate performance metrics from the similarity matrix like false acceptance rate (FAR), false rejection rate (FRR), half total error rate (HTER), verification rate at 1%, .1% and 1, etc [4].

$$FRR = (a/b) \times 100\% \quad (19)$$

Where a = number of rejected genuine identity claims and b = total number of genuine claims made.

$$FAR = (c/d) \times 100\% \quad (20)$$

Where c = number of accepted false identity claims and d = number of false identity claims made.

And HTER is given by

$$HTER = 0.5(FAR + FRR) \quad (21)$$

And verification rate is equal to 1-FRR.

## 5.3 Results

To show the efficiency of the proposed biogeography based algorithm we have plotted the receiver operating characteristic (ROC) curve having verification rate on the Y- axis and False Acceptance rate at X-axis.

We have plotted two ROC curves for the two approaches we followed in each figure below. In each figure, the lower dashed line shows the ROC curve for simple gabor filter and PCA technique without applying BBO and the upper solid line shows the ROC curve for the proposed BBO based algorithm.

The below two figure show the result, we have obtained using the two approaches, we discussed in the previous chapter. Figure 5.1 shows the result from the first approach using migration from the ideal matrix and fitness function as standard mean. Figure 5.2 shows the result from the second approach using migration from the test matrix and fitness function as standard deviation. Result obtained from first approach shows high verification rate than the original PCA and gabor algorithm. Result from the second approach also shows better performance than the original PCA and gabor filter except at few places where the curve from the BBO approach is slightly lower than the PCA and gabor filter curve. But most of the time it is higher or in line with the PCA and gabor filter curve. Efficiency can be easily seen at the starting point in both the results, where both the approaches outperform the PCA and gabor filter curve.

### 5.3.1 Adapted BBO- first approach using migration from ideal matrix and fitness function as mean

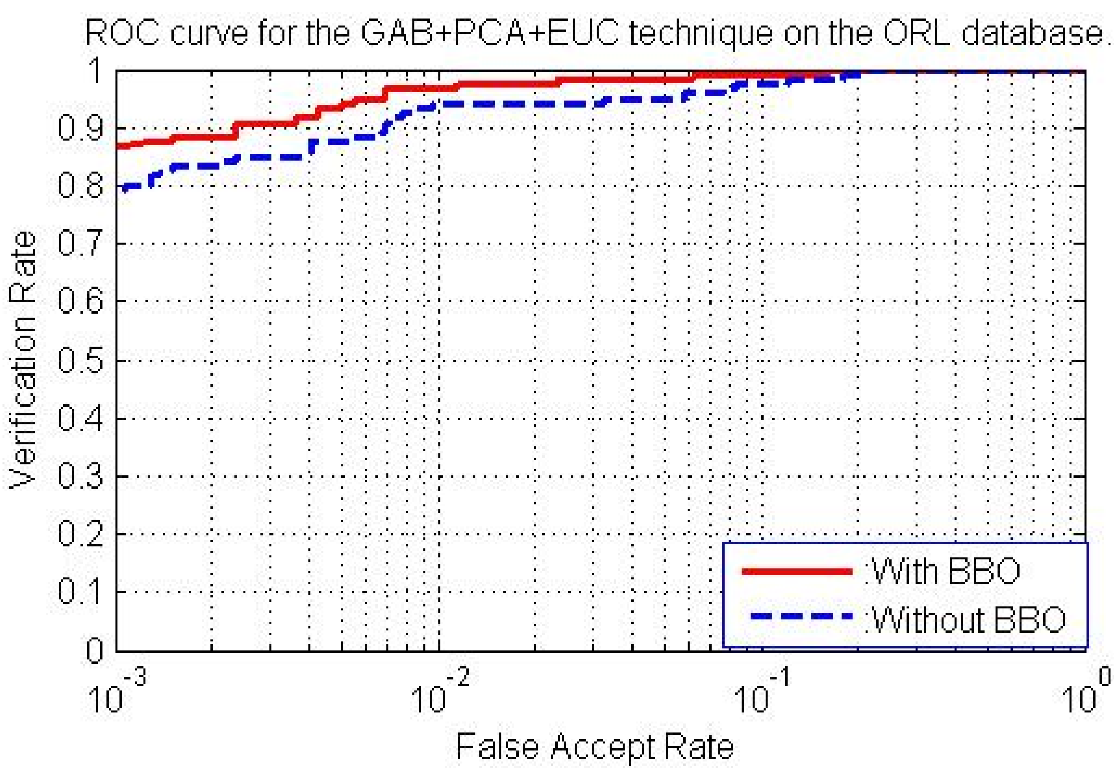



Figure 5.2 ROC curve for Adapted BBO- first approach.

### 5.3.2 Adapted BBO- second approach using migration from test matrix and fitness function as standard deviation

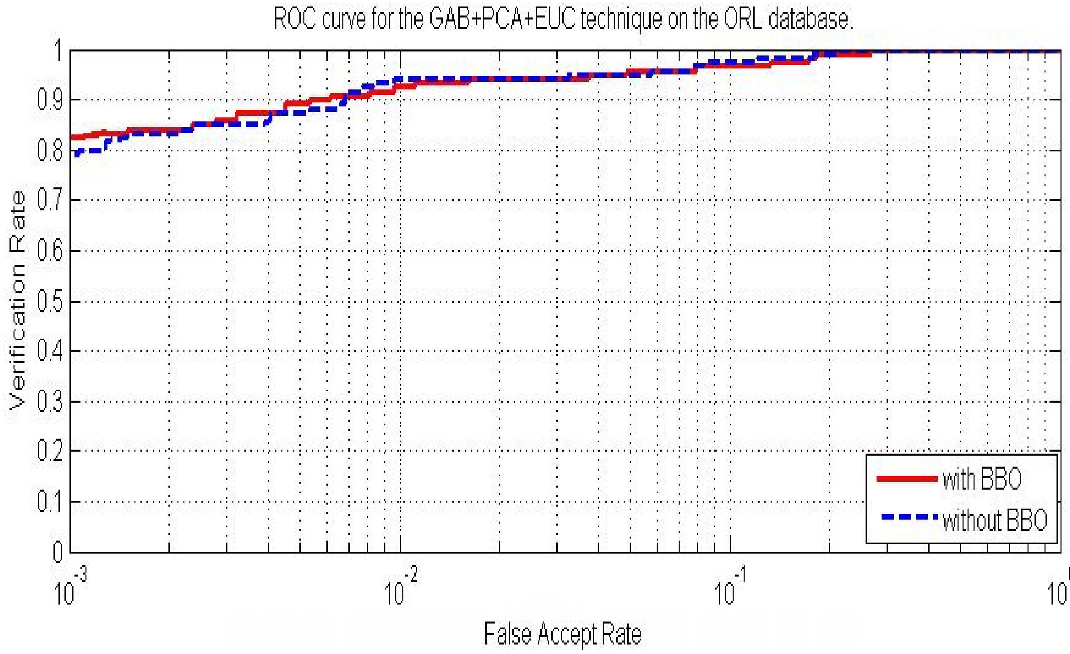



Figure 5.3 ROC curve for Adapted BBO-second approach

# Chapter 6
# Conclusion and future work

In this work we tried to develop a hybrid technique to solve the face recognition problem by using principal component analysis (PCA) and gabor filter. To improve further this hybrid technique we adapted the biogeography based optimization (BBO) technique.

In face recognition, first we need to extract the features from the images that can represent the images. PCA is one of the standard techniques used for feature extraction. The limitation of PCA is that it is not able to perform feature extraction properly, when used with raw images directly. This is because PCA requires properly aligned and uniformly illuminated images as input and this is not the case with raw images. In order to solve this challenge gabor filters can be used. Gabor filters when applied on a raw image, generates gabor filter response and convert the raw image into properly aligned and constantly illuminated image. To improve further this gabor PCA approach we have applied BBO to further optimize the desirable features obtained from the image after applying PCA.

We followed two different approaches to adapt BBO for the face recognition problem. We also modified certain parameters in order to incorporate BBO to build an efficient algorithm for face recognition. The algorithm used in first approach is simple and uses ideal matrix obtained from training set for migration and standard mean as fitness function. On the other hand the algorithm used in second approach is more complicated and uses test matrix for migration and standard deviation as a fitness function.

Result obtained from first approach shows higher verification rate than the original PCA and gabor algorithm. Result from the second approach also shows similar performance except at few places where the curve from the BBO approach is slightly lower than the PCA and gabor filter curve. But most of the time it is higher or in line with the PCA and gabor filter curve. Efficiency can be easily seen at the starting point in both the results, where both the approaches outperform the PCA and gabor filter curve. On the other hand, if we compare the two approaches then the first approach presents better result.

We have modified parameters to adapt BBO and applied BBO on the set of features extracted after applying gabor filter and PCA. It could be possible to find some other spot or multiple spots where BBO can be applied and with different parameter settings. It is also possible to form a hybrid technique by clubbing BBO with other optimization techniques like PSO, ACO etc. and employ it to achieve even better efficiency. We leave it for future work.

.

# Chapter 7
# Publication from Thesis

**Conference Name:** International Conference on Advances in Computer Science and Engineering (CSE-2013)

**Paper Title:** "An Efficient Biogeography Based Face Recognition Algorithm" (accepted)

**Authors:** Daya Gupta, Lavika Goel and Abhishek

**Status:** Paper is accepted at the conference and will be published.

**Location:** Los Angeles, CA, USA.

**Conference date:** 1-2 july, 2013

**Publisher/Proceedings:** Paper will be published by Atlantis Press. All accepted and registered papers will be submitted for indexing in ISI Proceedings (ISTP), Scopus and DBLP, EI Compendex.

# References

[1] Rabia Jafri and Hamid R. Arabnia, “A survey of face recognition techniques,” Journal of Information Processing Systems, Vol.5, No.2, June 2009.

[2] M. Turk and A. Pentland, “Eigenfaces for recognition. Journal of cognitive neuroscience”,3:72-86,1991.

[3] D Murugan, Dr. S Arumugam, K Rajalakshmi and Manish T I “Performance Evaluation of face recognition using Gabor filter, log Gabor filter and discrete wavelet transform”, International jouranal of computer science & information Technology (IJCSIT), Vol.2, No.1, February 2010.

[4] Vitomir s̊truc and Nikola Paves̊ic, “The complete Gabor-Fisher classifier for robust face recognition,” EURASIP journal on Advances in Signal Processing volume 2010.

[5] V. s̊truc, B. Vesnicer, N. Paves̊ic, “The phase-based Gabor Fisher classifier and its application to face recognition under varying illumination conditions,” in Proceedings of the 2nd International Conference on Signal Processing and Communication Systems (ICSPCS ’08), pp, 1-6, Gold Coast, Australia, December 2008.

[6] C. Liu and H. Wechsler, “Gabor feature based classification using the enhanced Fisher linear discriminant model for face recognition”, IEEE Transactions on image Processing, vol. 11, no. 4, pp. 467-476, 2002.

[7] L. Shen and L. Bai, “A review of Gabor wavelets for face recognition”, Pattern Analysis and Applications, vol. 9, no. 2, pp.273-292, 2006.

[8] L. Shen, L.Bai, and M. fairhurst, “Gabor wavelets and general discriminant analysis for face identification and verification”, Image and Vision Computing, vol. 25, no. 5, pp. 553-563, 2007.

[9] Vitomir štruc and Nikola Pavešic, “Gabor-based kernel partial-least squares discriminantion features for face recognition”, Informatica, vol.20, no.1, pp.115-138, 2009.

[10] Er. Navdeep Kaur Johal, Er. Poonam Gupta and Er. Amanpreet kaur, “Face recognition using biogeography based optimization”, International Journal of Computer Science and Information Security, Vol.9, No.5, May2011.

[11] D. Simon, “Biogeography based optimization”, IEEE Transactions on Evolutionary Computation, vol. 12, no. 6, pp. 702-713, 2008.

[12] Ki-Chung Chung, Seok Cheol Kee and Sang Ryong Kim, “Face recognition using principal component analysis of Gabor filter responses”.

[13] Lavika Goel, Daya Gupta and V. K Panchal, “Hybrid bio-inspired techniques for land cover feature extraction: a remote sensing perpective”, Applied soft computing. Elsevier publications, pp. 832-849, vol. 12(2), 2011.

[14] Iordanis Mpiperis, Sotiris Malassiotis, Vassilios Petridis and Michael G. Strintzis, “3D Facial expression recognition using swarm intelligence”.

[15] Lei Du, Zhenhong Jia, Liang Xue, “Human Face Recognition Based on Principal Component Analysis and Particle Swarm Optimization -BP Neural Network”, Third International Conference on Natural Computation (ICNC 2007), 2007.

[16] Hamidreza Rashidy Kanan, Karim Faez and Mehdi Hosseinzadeh, “Face Recognition System Using Ant Colony Optimization-Based Selected Features”, Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Security and Defense Applications (CISDA 2007), 2007.

[17] R.Chellappa, C.L Wilson and S. Sirohey, “Human and machine recognition of faces: a survey”, Proc. Of IEEE, vol. 83(5), pp. 705-741, 1995.

[18] M.-H. Yang, D. Kriegman, and N. Ahuja, “Detecting faces in images: A survey”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(1):34–58, January 2002.

[19] Principal Component Analysis, second edition by I.T Jollife.

[20] Peter N. belhumeur, Joao P. hespaha and David J. kreigman, “eigengaces vs. Fisherfaces : Recognition Using Class Specific Linear Projection”.

[21] Y. Adini, Y. Moses, and S. Ullma, “Face recognition: the problem of compensating for changes in illumination direction”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 19:721–732, 1997.

[22] Proceedings of world academy of science, Engineering and Technology, volume 22, july 2007 ISSN 307-6884P WASET.

[23] J.G Daugman, “Uncertainity Relation for Resolution in space, spatial frequency and orientation optimized by 2-D visual cortical filters”, journal of optical society America, vol. 2, no. 7, 1985, pp. 1160-1169.

[24] Wei-Lun chao, “Gabor wavelet transform and its application”.

[25] A. Wallace, “The geographical Distribution of animals (two volumes)”, boston, M A: Adamant medaia corporation, 2005.

[26] C. Darwin, “the origin of species”, New York, 1995.

[27] R. MacArthur and E. Wilson, “the theory of Biogeography”, Princeton, Princeton univ. press, 1967.

[28] T. Wesche, G. Goertler and W. Hubert, “Modified Habitat Suitability index model for brown trout in southeastern Wyoming”, North Amer. J. Fischeries manage, vol. 7, pp. 232-237, 1987.

[29] H. Muhlenbein and D. Schlierkamp-Voosen, “Predictive models for the breeder genetic algorithm: I. Continuous parameter optimization,” Evol. Comput., vol. 1, pp. 25–49, 1993.

[30] T. Back,“Evolutionary Algorithms in Theory and Practice”. Oxford, U.K.: Oxford Univ. Press, 1996.

[31] W. Zhao, R. Chellappa, A. Rosenfeld, and P. Phillips. “Face recognition: A literature survey”, ACM Computing Surveys, pages 399–458, 2003.

[32] J. S. Bruner and R. Tagiuri, “The percepton of people”, Handbook of Social Psycology, 2(17), 1954.

[33] M. Ballantyne, R. S. Boyer, and L. Hines. Woody bledsoe: His life and legacy. AI Magazine, 17(1):7–20, 1996.

[34] W. W. Bledsoe, “The model method in facial recognition”, technical report pri 15, Panoramic Research, Inc., Palo Alto, California, 1964.

[35] W. W. Bledsoe, “Man-machine facial recognition: Report on a largescale experiment”, Technical report pri 22, Panoramic Research, Inc., Palo Alto, California, 1966.

[36] W. W. Bledsoe, “Semiautomatic facial recognition”, technical report sri project 6693, Stanford Research Institute, Menlo Park, California, 1968.

[37] T. Kenade,“Picture Processing System by Computer Complex and Recognition of Human Faces”, PhD thesis, Kyoto University, November 1973.

[38] T. J. Stonham, "Practical face recognition and verification with wizard", In H. D. Ellis, editor, Aspects of face processing. Kluwer Academic Publishers, 1986.

[39] L. Sirovich and M. Kirby "Low-dimensional procedure for the characterization of human faces", Journal of the Optical Society of America A- Optics, Image Science and Vision, 4(3):519–524, March 1987.

[40] M. Kirby and L. Sirovich, "Application of the karhunen-loeve procedure for the characterization of human faces", IEEE Transactions on Pattern Analysis and Machine Intelligence, 12(1):103–108, 1990.

[41] M. Turk, "A random walk through eigenspace", IEICE Transactions on Information and Systems, E84-D(12):1586–1595, December 2001.

[42] K.-M. Lam and H. Yan, "Fast algorithm for locating head boundaries", Journal of Electronic Imaging, 03(04):351–359, October 1994.

[43] S. K. Singh, D. S. Chauhan, M. Vatsa and R. Singh, "A robust skin color based face detection algorithm", Tamkang Journal of Science and Engineering, 6(4):227–234, 2003.

[44] S. Kawato and N. Tetsutan,"Detection and tracking of eyes for gaze camera control", In Proc. 15th Int. Conf. on Vision Interface, pages 348–353, 2002.

[45] M. Divjak and H. Bischof,"Real-time video-based eye blink analysis for detection of low blink-rate during computer use",In 1st International Workshop on Tracking Humans for the Evaluation of their Motion in Image Secuences (THEMIS), pages 99–107, Leeds, UK, September 2008.

[46] C.-C. Han, H.-Y. M. Liao, K. chung Yu, and L.-H. Chen, Lecture Notes in Computer Science, volume 1311, chapter Fast face detection via morphology-based pre-processing, pages 469–476. Springer Berlin / Heidelberg, 1997.

[47] K.-K. Sung, “Learning and Example Selection for Object and Pattern Detection”, PhD thesis, Massachusetts Institute of Technology, 1996.

[48] H. A. Rowley, S. Baluja, and T. Kanade,”Neural network-based face detection”, IEEE trans. Pattern Analysis and Machine Intelligence, 20(1):23–38, January 1998.

[49] E. Osuna, R. Freund, and F. Girosi, “Training support vector machines: An application to face detection”, Proceedings of the IEEE Conf. Computer Vision and Pattern Recognition, pages 130–136, June 1997.

[50] H. Schneiderman and T. Kenade, “Probabilistic modeling of local appearance and spatial relationships for object recognition”, Proc. IEEE Conf. Computer Vision and Pattern Recognition, pages 45–51, 1998.

[51] A. Jain, R. Duin, and J. Mao,“Statistical pattern recognition: A review”,IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(1):4–37, January 2000.

[52] K. Pearson,“On lines and planes of closest fit to systems of points in space”, Philosophical Magazine, 2(6):559–572, 1901.

[53] S.Watanabe, “Karhunen-loeve expansion and factor analysis theoretical remarks and applications”, In Proc. 4th Prague Conference on Information Theory, 1965.

[54] R. Brunelli and T. Poggio, “Face recognition: Features versus templates”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 15(10):1042–1052, October 1993.

[55] B. Heisele, P. Ho, and T. Poggio, “Face recognition with support vector machines: Global versus component-based approach”, In Proc. Of the Eighth IEEE International Conference on Computer Vision, ICCV 2001, volume 2, pages 688–694, Vancouver, Canada, July 2001.

[56] L. Torres, “Is there any hope for face recognition?”, In Proc. of the 5th International Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS, Lisboa, Portugal, 21-23 April 2004.

[57] R. Gross, J. Shi, and J. Cohn, “Quo vadis face recognition? - the current state of the art in face recognition”, Technical report, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA, June 2001.

[58] M. Bartlett, J. Movellan, and T. Sejnowski, “Face recognition by independent component analysis”, IEEE Trans. on Neural Networks, 13(6):1450–1464, November 2002.

[59] M.-H. Yang, “Face recognition using kernel methods”, Advances in Neural Information Processing Systems, 14:8, 2002.

[60] B. Moghaddam, “Principal manifolds and probabilistic subspaces for visual recognition”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(6):780–788, June 2002.

[61] V. Blanz and T. Vetter,“Face recognition based on fitting a 3-D morphable model”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 25(9):1063–1074, September 2003.

[62] N. Ahmed, T. Natarajan, and K. R. Rao, “Discrete cosine transform”, IEEE Transactions on Computers, 23:90–93, 1974.

[63] D. Cai, X. He, and J. Han, “Semi-supervised discriminant analysis” Computer Vision. IEEE 11th International Conference on Computer Vision, 14:1–7, 2007.

[64] N. Intrator, D. Reisfeld, and Y. Yeshurun, “Face recognition using a hybrid supervised/unsupervised neural network”, Pattern Recognition Letters, 17:67–76, 1995.

[65] A. A. Bhuiyan and C. H. Liu, “On face recognition using gabor filters”, In Proceedings of World Academy of Science, Engineering and Technology, volume 22, pages 51–56, 2007.

[66] Marco Dorigo, Ant Colony Optimization, Scholarpedia, 2(3):1461, 2007.

[67] D. Bratton and J. Kennedy, “Defining a Standard for Particle Swarm Optimization”, Proceedings of the 2007 IEEE Swarm Intelligence Symposium.

[68] N. Holden and A.A. Freitas, “A hybrid particle swarm/ant colony algorithm for the classification of hierarchical biological data”, In: Proc. 2005 IEEE Swarm Intelligence Symposium (SIS-05), pp. 100 107, IEEE, 2005.

# Appendix A
# Abbreviations

PCA – Principal Component Analysis

BBO – Biogeography Based Optimization

PSO – Particle Swarm Optimization

ACO – Ant Colony Optimization

DCT – Discrete Cosine Transformation

ROC – Receiver Optimizing Characteristic

ORL – Olivetti Research Laboratory

ICA – Independent Component Analysis

LDA – Linear Discriminant Analysis

FLD – Fisher Linear Discriminant

HMM – Hidden Markov Model

SVM – Support Vector Machine

SIV – Suitability Index Variable

HSI – Habitat Suitability Index

FFNN – Feed Forward Neural Network

AS – Ant System

FAR – False Acceptance Rate

FRR – False Rejection Rate

KPCA – Kernel PCA

PC – Principal Component

MSE – Mean Square Error

HTER – Half Total Error Rate

DE – Differential Evolution

# Appendix B
# An Introduction to MATLAB

MATLAB is short for “matrix laboratory.” MATLAB is a powerful software package that has built-in functions to accomplish a diverse range of tasks, from mathematical operations to three-dimensional imaging. Additionally, MATLAB has a complete set of programming constructs that allows users to customize programs to their own specifications. MATLAB is commonly used for Math and computation, Algorithm, simulation and prototyping, Modeling, simulation and prototyping, Data analysis, exploration and visualization, Scientific and engineering and visualization and Application development, including graphical user interface building.

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations in a fraction of the time it would take to write a program in a scalar non interactive language such as C or FORTRAN.

MATLAB was originally written to provide easy access to matrix software developed by the LINPACK and EISPACK projects. Today MATLAB uses software developed by the LAPACK and ARPACK projects, which together represent the state-of-the-art in software for matrix computation.

MATLAB features a family of application-specific solutions called toolboxes. Very important to most users of MATLAB toolboxes allow you to learn and apply specialized technology. Toolboxes are comprehensive collections of MATLAB functions (M-files) that extend the MATLAB environment to solve particular classes of problems. Areas in which toolboxes are available include signal processing, control system neural networks, fuzzy logic, wavelets, simulation and many others.

**FILE TYPES**

MATLAB can read and write several types of files. There are mainly five different types of files used in MATLAB which is used for storing data or programs.

- **M-FILES** - They are the standard ASCII files, with a .m extension to the file name. There are basically two types of files and they are SCRIPT and FUNCTION file. In general, mostly MATLAB files are saved as M-FILES.
- **MAT-FILES** - They are the binary data-files, with a .mat extension to the filename. These files are created when you save the MATLAB data with the save command. The data which you save in MATLAB can only be read by mat lab as it gets saved in a special format.
- **FIG-FILES** - They are the binary figure-file, with a .fig extension to the filename. Such files are created by saving a figure in this format by using the save and saveas option in it. These files basically create all kind of information which is used for again recreating a figure and can be opened by filename.fig.
- **P-FILES** - These are the compiled M-File, with a .p extension to the filename. These file can be executed directly without using any compiler and parsed in it. These files are created with the P-CODE command.
- **MEX-FILES** - These are MATLAB-callable Fortran and C programme, with the .mex extension to the filename.