Swarm Optimization Algorithms for Face Recognition

Juicy Ray

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela – 769 008, India

Swarm Optimization Algorithms for Face Recognition

Dissertation submitted in

May 2013

to the department of

Computer Science and Engineering

of

National Institute of Technology Rourkela

in partial fulfillment of the requirements

for the degree of

Bachelor of Technology

by

Juicy Ray

(Roll 109CS0180)

under the supervision of

Prof. Banshidhar Majhi

Department of Computer Science and Engineering

National Institute of Technology Rourkela

Rourkela – 769 008, India

Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, India.   www.nitrkl.ac.in

Prof. Banshidhar Majhi
Professor

May 9, 2013

Certificate

This is to certify that the work in the thesis entitled ***Swarm Optimization Algorithms for Face Recognition*** by ***Juicy Ray***, bearing roll number 109CS0180, is a record of her work carried out under my supervision and guidance in partial fulfillment of the requirements for the award of the degree of *Bachelor of Technology* in *Computer Science and Engineering.*

Banshidhar Majhi

Acknowledgment

First and foremost, I would like to thank my supervisor Prof. B. Majhi for introducing me to this exciting area of Biometry. I am especially indebted to him for his guidance, support and patience with me throughout the course of my research. He taught me the essence and principles of research and guided me through until the completion of this thesis. It is due to his faith in me that today I am submitting this thesis. It has been my privilege working with him and learning from him.

I would also like to thank Prof. Ratnakar Dash for showing me innovative research directions for the entire period of carrying out the research. I am indebted to all the professors, batch mates and friends at National Institute of Technology Rourkela for their cooperation.

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

Juicy Ray

Abstract

In this thesis, a face recognition system based on swarm intelligence is developed. Swarm intelligence can be defined as the collective intelligence that emerges from a group of simple entities; these agents enter into interactions, sense and change their environment locally. A typical system for face recognition consists of three stages: feature extraction, feature selection and classification. Two approaches are explored. First, Bacterial Foraging Optimization(BFO), in which the features extracted from Principal Component Analysis(PCA) and Linear Discriminant Analysis(LDA) are optimized. Second, Particle Swarm Optimization(PSO), which optimizes the transform coefficients obtained from the Discrete Cosine Transform(DCT) of the images. PCA, LDA and DCT are all appearance-based methods of feature extraction. PCA and LDA are based on global appearance whereas DCT is performed on a block by block basis exploring the local appearance-based features. Finally, for classification Euclidean distance metric is used. The algorithms that have been applied are tested on Yale Face Database.

Keywords: Swarm intelligence, feature extraction, feature selection, PCA, LDA, DCT, BFO, PSO

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Face recognition is a part of the capability of human beings and is a task that humans perform routinely and effortlessly in daily lives. Though research into this area dates back to the 1960's, face recognition is still an area of active research since a completely successful approach or model has not been proposed to solve the face recognition problem.

Wide availability of powerful and low-cost desktop and embedded computing systems has created an enormous interest in automatic processing of digital images in a variety of applications including biometric authentication, crowd surveillance, human-computer interaction, multimedia management, mug-shot matching, user verification and user access control. Because of its prevalence as an institutionalized and accepted guarantor of identity since the advent of photography, there are also large legacy systems based on face images like voter registration, passports and driver's licenses, all being automated currently [1]. Video indexing is another example of legacy data for which face recognition.

The problem of face recognition can be stated as identifying an individual from images of the face. The inadequacy of automated face recognition systems is especially apparent when compared to our own innate face recognition ability. We perform face recognition, an extremely complex visual task, with in a span of seconds and our own recognition ability is far more robust than any computer can hope to be.

We can recognize a familiar individual under very adverse lighting conditions, from varying angles or viewpoints. Scaling differences or different backgrounds do not change our ability to recognize faces and we can even recognize individuals with just a fraction of their face visible or even after several years have passed. Furthermore, we are able to recognize the faces of several thousand individuals whom we have met during our lifetime. So, its a true challenge to build an automated system which equals human ability to recognize faces.Many face recognition algorithms have been developed. An exhaustive survey on FR techniques [1] is given in Table 1.1

1.1   Face as a biometric

Biometrics are automated methods of recognizing a person based on a physiological or behavioral characteristic. The different features that are measured are face, fingerprints, hand shape, calligraphy, iris, retinal, vein, and voice. Face recognition has a number of strengths to recommend it over other biometric modalities in certain circumstances. Face recognition as a biometric derives a number of advantages from being the primary biometric that humans use to recognize one another. It is well-accepted and easily understood by people, and it is easy for a human operator to arbitrate machine decisions in fact face images are often used as a human-verifiable backup to automated fingerprint recognition systems.

Face recognition has the advantage ubiquity and of being universal over other major biometrics, in that everyone has a face and everyone readily displays the face. (Whereas, for instance, fingerprints are captured with much more difficulty and a significant proportion of the population has fingerprints that cannot be captured with quality sufficient for recognition.) With some configuration and co-ordination of one or more cameras, it is easy to acquire face images without active participation of the subject. Such passive identification might be desirable for customization of user services and consumer devices, whether that be opening a house door as the

Method	Category	Characteristics
PCA [12]	Appearance-based	PCA for learning eigenfaces
LDA [13]	Appearance-based	LDA for learning fisherfaces
2D-PCA [25]	Appearance-based	For better statistical properties
ICA [16]	Appearance-based	ICA for catch facial independent properties
Laplacianfaces [17]	Holistic-based	Nonlinear dimension reduction for finding bases, LPP
Evolutionary pursuit [18]	Holistic-based	For finding the best projection bases based on generalization error
Kernel PCA And Kernel LDA [19]	Holistic-based	Mapping the image into higher-dimensional space by the kernel function
Sparse representation [20]	Holistic-based	Using L1 minimization for finding sparse representation
Gabor and dynamic link architecture [21]	Feature-based	Gabor features extracted at facial feature locations
Gabor and elastic bunch graph matching [22]	Feature-based	Gabor features extracted at facial feature locations, and obtaining the robust representation by the FBG matching.
LBP [23]	Feature-based	Local binary patterns are introduced
SIFT [24]	Part-based	Using SIFT feature with spatial constraints to compare faces

Table 1.1: Summary of Face Recognition techniques

owner walks up to it, or adjusting mirrors and car seats to the drivers presets when sitting down in their car.

1.2   Face Recognition using Swarm Intelligence

Groups of starlings can form impressive shapes as they travel northward together in the springtime. This is among a group of natural phenomena based on swarm behavior. The behaviour of these intelligent swarms has opened new approaches for feature selection in face recognition as well. The term swarm intelligence describes the collective behaviour of what are usually simple individuals in decentralized systems. The behaviour of individuals allows the entire system to solve complex tasks. Bacterial foraging optimization and Particle swarm optimization are applications of swarm intelligence in the area of optimization. During optimization, one attempts to find the minimum of a target function. A special characteristic of these algorithms is that they generally provide good results, without the computational complexity typically required for finding the optimum solution.

1.3   Face database

There are several publicly available face databases for the research community to use for algorithm development, which provide a standard benchmark when reporting results. Different databases are collected to address a different type of challenge or variations such as illumination, pose, occlusion, etc. In this project, I have used the Yale database [11] [13]which contains 165 gray-scale images in GIF format of 15 subjects.The images are at resolution 243x320 pixels with 24 bits graylevel. There are 11 images of each person, one for each variation such as different facial expression, center-light, with glasses, happy, left-light, with and without glasses, normal, right-light, sad, sleepy, surprised, and wink. Some sample images of this

Figure 1.1: Typical examples of sample face images from the Yale face database

database are shown in Figure.1.1

1.4   Thesis Organisation

The rest of the thesis constitutes the following six chapters-

Chapter 2: Steps in Face Recognition

This chapter outlines different steps of face recognition in detail.

Chapter 3: Feature Extraction

This chapter deals with various feature extraction algorithms for face recognition.

Chapter 4: Feature Selection

This chapter discusses how the swarm intelligence based optimization algorithms are used for optimizing the feature vector set.

Chapter 5: Face Recognition using swarm intelligence

This chapter is based on the details of the face recognition method using swarm optimization based selected features.

Chapter 6: Results and Analysis

This chapter deals with a comparative analysis of the two algorithms carried out followed by results drawn from the research.

Chapter 7: Conclusion

In this chapter, conclusion and possible directions for future work are given.

Chapter 2

Steps in Face Recognition
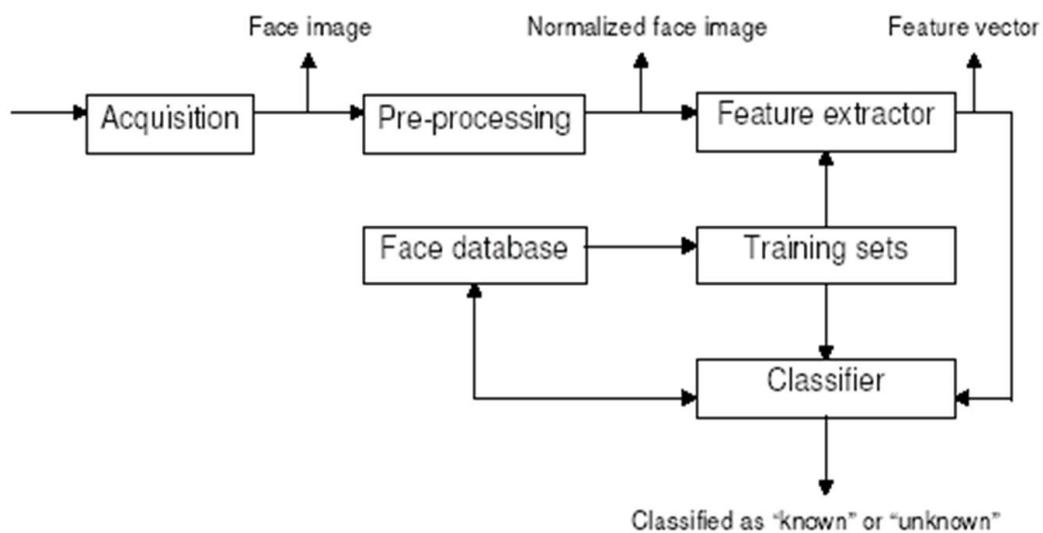

Figure 2.1: Outline of a typical face recognition system

1. **Acquisition module -** It is the module where the face image under consideration is presented to the system. It can request an image from several different environments: The face image can be an image file that is located on a disk or it can be captured by a frame grabber or can be scanned from paper with the help of a scanner.

2. **Preprocessing -** This is done to enhance the images to improve the recognition performance of the system:

   - **Image normalization -** It is done to change the acquired image size to a default size say, 128 X 128 On which the system can operate.

   - **Histogram equalization-** For images that are too dark or too light, it modifies the dynamic range and improves the contrast of the image so that facial features become more apparent.

   - **Median filtering -** For noisy images especially obtained from a camera or from a frame grabber, median filtering can clean the image without much loss of information.

   - **Background removal -** In order to deal primarily with facial information itself, face background can be removed. This is important for face recognition systems where entire information contained in the image is used. It is obvious that, for background removal, the preprocessing module should be capable of determining the face outline.

   - **Translational and rotational normalizations-** In some cases, it is possible that in the face image the head is somehow shifted or rotated. The head plays a major role in the determination of facial features. The pre- processing module determines and normalizes the shifts and rotations in the head position.

   - **Illumination normalization -** Face images taken under different illuminations can degrade recognition performance especially for face recognition systems based on the principal component analysis in which entire face information is used for recognition.

3. **Feature Extraction -** This module finds the key features in the face that will be used in classification. It is responsible for composing a feature vector that is well enough to represent the image.

4. **Classification Module -** In this module, the extracted features of the face image is compared with the ones stored in the face library with the help of a pattern classifier.

5. **Training sets -** It is used during the learning phase of face recognition process

6. **Face database -** On being classified as unknown by the classification module, the face can be added to the library with their feature vectors for future comparisons

To the steps described above, an additional important step is added as shown in figure 2.2, the step of feature selection:
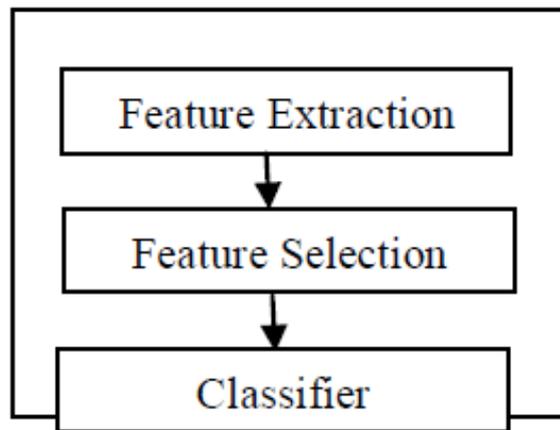
Figure 2.2: Feature Extraction process

Feature Selection-The feature selection seeks for the optimal set of d features out of m features obtained from feature extraction module. Several methods have been previously used to perform feature selection on training and testing data. In the developed FR system I have utilized evolutionary feature selection algorithms based on swarm intelligence called the Bacteria Foraging Optimization and Particle Swarm Optimization

Chapter 3

Feature Extraction

Face recognition algorithms can be classified into two broad categories according to feature extraction schemes for face representation: feature-based methods and appearance-based methods [1]

3.1 Feature-based Methods

Properties and geometric relations such as the areas, distances, and angles between the facial feature points are used as descriptors for face recognition in this approach [2]. These methods try to define a face as a function and attempts to find a standard template of all the faces. The features can be defined independent of each other. For example, a face can be divided into eyes, face contour, nose 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. These standard patterns are compared to the input images to detect faces. This approach is simple to implement, but its inadequate for face detection. It cannot achieve good results with any kind of pose angle variation, scale difference and shape change.

3.2   Appearance-based Methods

Appearance-based methods consider the global properties of the face image intensity pattern [8]. Typically appearance-based face recognition algorithms proceed by computing basis vectors to represent the face data efficiently. In the next step, the faces are projected onto these vectors and the projection coefficients can be used for representing the face images. Popular algorithms such as PCA, LDA, ICA, LFA, Correlation Filters, Manifolds and Tensorfaces are based on the appearance of the face. Holistic approaches to face recognition have trouble dealing with pose variations. In general, appearance-based methods rely on techniques from statistical analysis and machine learning to find the relevant characteristics of face images. For feature extraction purpose in this project, appearance-based methods like Principle Component Analysis (PCA), Linear Discriminant Analysis (LDA), Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT) have been used. They are described in detail in the next subsection.

3.2.1   Principal Component Analysis(PCA)

PCA is an orthogonal linear transformation that transforms the data to a new coordinate system such that greatest variance by any projection of the data comes to lie on the rst coordinate, the second greatest variance comes up in the second coordinate, and so on. The idea of PCA is illustrated in figure 3.1. Eigenfaces [12] also known as Principal Components Analysis (PCA) find the minimum mean squared error linear subspace that maps from the original N dimensional data space into an M-dimensional feature space. By doing this, Eigenfaces (where typically $M << N$) achieve dimensionality reduction by using the M eigenvectors of the covariance matrix corresponding to the largest eigenvalues. The resulting basis vectors are obtained by finding the optimal basis vectors that maximize the total variance of the projected data (i.e. the set of basis vectors that best describe the data). Usually the mean x is extracted from the data, so that PCA is equivalent
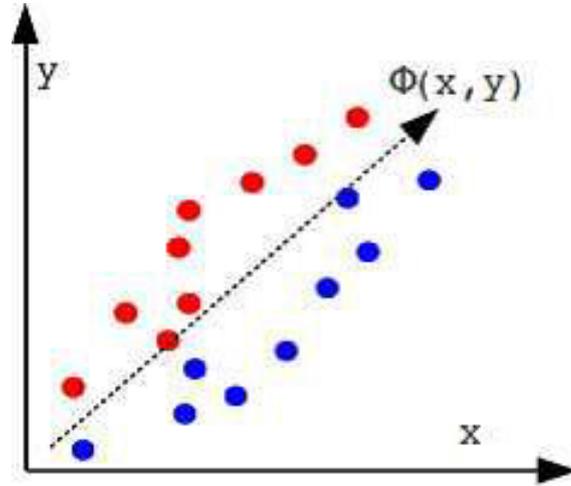
Figure 3.1: PCA. x and y are the original basis. ϕ is the first principal component

to Karhunen-Loeve Transform (KLT). So, let Xnxm be the the data matrix where
x1,..., xm are the image vectors (vector columns) and n is the number of pixels per
image. The KLT basis is obtained by solving the eigenvalue problem [3]:

$$C_x = \phi \Lambda \phi^T \qquad (3.1)$$

where C_x is the covariance matrix of the data:

$$C_x = \frac{1}{m} \sum_{i=1}^{m} x_i x_i^T \qquad (3.2)$$

$\phi = [\phi1,..., \phin]$ is the eigenvector matrix of C_x. Λ is a diagonal matrix,the
eigenvalues $\Lambda1, ..., \Lambdan$ of C_x are located on its main diagonal. $\Lambdai$ is the variance of
the data projected on $\phii$. The application of pca on an image from the yale database
is shown in figure 3.2

The primary advantage of this technique is that it can reduce the data needed
to identify the individual to 1/1000th of the data presented. PCA is good for data
representation but not necessarily for class discrimination as we will discuss next.

Figure 3.2: (Starting from top-left)(a)Class Mean image (b) Overall Mean Image (c)Test Image (d)Reconstructed Test image)

3.2.2 Linear Discriminant Analysis(LDA)

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 point from the same class is close. Consequently, LDA obtains differenced projection vectors for each class. Suppose we have m samples x1,. . . ,xm belonging to c classes; each class has mk elements. We assume that the mean has been extracted from the samples, as in PCA. The ratio of between -class scatter to the within-class scatter is calculated which is the optimizing criterion in LDA:

$$Criterion(W) = inv(Sw) \times (Sb) \tag{3.3}$$

Between-class Scatter:

$$Sw = \sum_{i=1}^{c} |\chi_i|(\mu_i - \mu)(\mu_i - \mu)^T \tag{3.4}$$

Within-class Scatter:

$$Sb = \sum_{i=1}^{c} \sum_{x_k \varepsilon \chi_i} (x_k - \mu_i)(\mu_k - \mu_i)^T \tag{3.5}$$
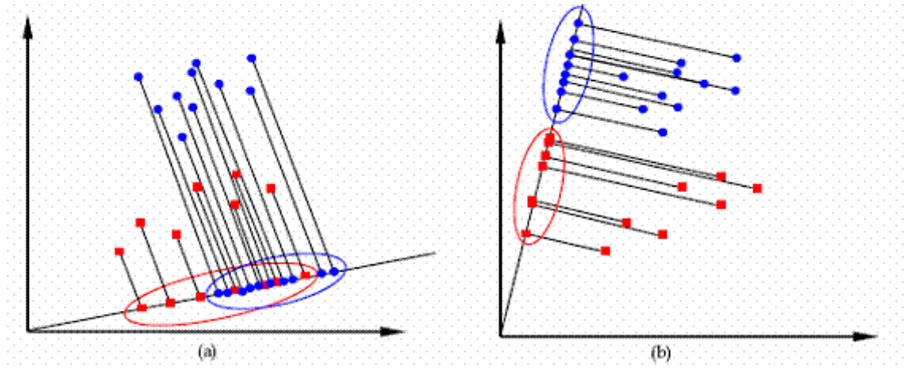
Figure 3.3: (a) The two classes are not well separated when projected onto this line (b) This line succeeded in separating the classes as well as reduces dimensionality from two features (x1,x2) to only a value y [4]

Typically when dealing with face images (and most other image based pattern recognition problems) the number of training images is smaller than the number of pixels (or equivalently dimensionality of the data), thus the within-class scatter matrix Sw is singular causing problems for LDA. To address this issue [13] first performs PCA to reduce the dimensionality of the data in order to overcome this singular-matrix problem and then applies LDA in this lower-dimensional PCA subspace.

3.2.3   Discrete Cosine Transform(DCT)

The Discrete Cosine Transform (DCT) expresses a sequence of data points in terms of a sum of cosine functions oscillating at different frequencies. 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.

For an M × N image, where each image corresponds to a 2D matrix, DCT
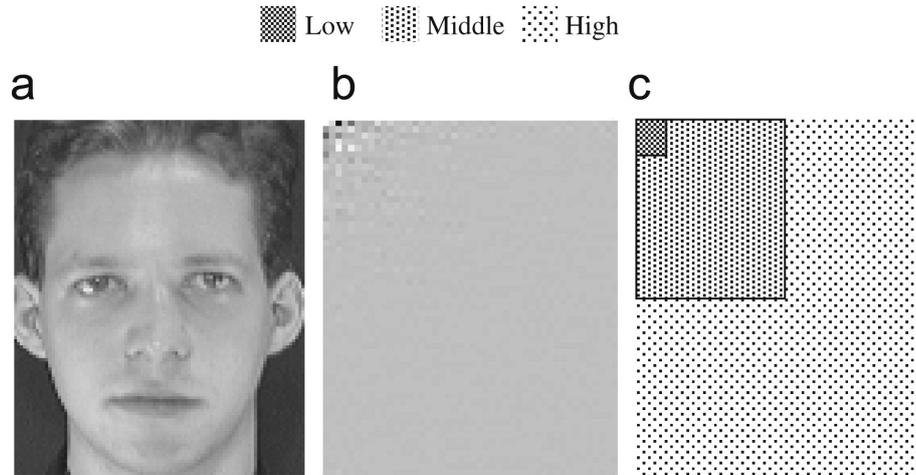
Figure 3.4: (a) A typical face image (b) its DCT transformed image and (c) typical division of the DCT coefficients into low, middle and high frequencies (Source: [14])

coefficients are calculated as follows [5]:

$$F(u,v) \quad = \quad \frac{1}{\sqrt{MN}}\alpha(u)\alpha(v)\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)\times cos\frac{(2x+1)\mu\pi}{2M}\times cos\frac{(2y+1)\mu\pi}{2N} \quad (3.6)$$

where u=1,...,M v=1,...,N

$$\alpha(\omega) \quad = \quad \begin{cases} 1 & \text{if } \omega = 0, \\ \sqrt{(2)} & \text{if } otheriwse, \end{cases} \quad (3.7)$$

f(x,y) – The image intensity function

F(u,v) – A 2D matrix of DCT coefficients The matrix F(u,v) can be truncated, retaining the upper-left area, which has the most information, reducing the dimensionality of the problem.This is illustrated in figure 3.4.

Chapter 4

Feature Selection

Feature selection (FS) is a global optimization problem in machine learning, which reduces the number of features, removes irrelevant, noisy and redundant data, and results in acceptable recognition accuracy [8]. Although feature selection is primarily performed to select relevant and informative features, it can have other motivations, including:

- General data reduction, to limit storage requirements and increase algorithm speed;

- Feature set reduction, to save resources in the next round of data collection or during utilization;

- Performance improvement, to gain in predictive accuracy;

- Data understanding, to gain knowledge about the process that generated the data or simply visualize the data

In section 3, all the methods depicted use the top n principal components (or transform coefficients) when used directly for dimension reduction and eliminate the lower order. However, there may be some useful information in lower order principal components leading to a significant contribution in improving the recognition rate. Feature selection thus becomes an important step, affecting the performance of

a pattern recognition system. Here, search methods based on swarm intelligence algorithms are developed to select the appropriate principal components or transform coefficients from the set of extracted feature vectors. They are discussed in detail in the subsections below.

4.1 Bacterial Foraging Optimization(BFO)

BFO is based on foraging strategy of bacteria Escherichia coli. After many generations, bacteria with poor foraging strategies are eliminated while; the individuals with good foraging strategy survive signifying survival of the fittest. The whole process can be divided into three sections, namely, chemotaxis, reproduction, and elimination and dispersal [15].

4.1.1 Chemotaxis

Chemotaxis can be defined as foraging behavior of bacteria in which it try to avoid noxious substances and search for nutrient rich substances by climbing up the nutrient concentration. This process involves two actions; either a run (in the same direction as the previous step) or tumble (in an absolutely different direction from the previous one). In order to explore whole search space there is a limit on run steps in a particular direction. So, bacteria tumble after some run steps. Suppose $\theta i(j,k,l)$ be the position of ith bacterium at jth chemotactic, kth reproductive and lth elimination & dispersal loop. Then chemotactic movement of the bacterium may be mathematically represented by following equation.

In the above expression, C(i) is the size of the step taken in random direction and $\Delta(i)$ indicates a vector in the arbitrary direction. When the bacterial movement is run, $\Delta(i)$ remains unchanged; otherwise, $\Delta(i)$ is a random vector whose elements lie in [-1, 1]. Fitness function, denoted as J(i,j,k,l), will be evaluated for each step of run or tumble in the chemotactic process.

4.1.2 Reproduction

The health of each bacterium is calculated as the sum of the step fitness during its life, namely,

$$J_{health}^{i} \quad = \quad \sum_{j=1}^{N_c+1} J(i, j, k, l) \tag{4.1}$$

where N_c is number of chemotactic steps. All bacteria are sorted in increasing order according to health status. In the reproduction step, only the first half of population survives and a surviving bacterium reproduces by splitting into two daughter bacteria, which are then placed at the same locations. Thus, the population of bacteria keeps constant.

4.1.3 Elimination & Dispersal

The chemotaxis provides a basis for local search, and the reproduction process speeds up the convergence of the algorithm. However, only chemotaxis and reproduction are not enough for global optima searching. Since bacteria may get stuck around the initial positions or local optima, it is possible for the diversity of BFO to change either gradually or suddenly to eliminate the accidents of being trapped into the local optima. In BFO, according to a preset probability Ped, bacteria are eliminated and dispersed after a certain number of reproduction steps. After elimination, they are moved to another position within the environment.

4.1.4 BFO Algorithm

The algorithm [6] is as stated below:

Step1 Initialize parameters p, S, N_c, N_s, $N_r e$, $N_e d$, $P_e d$, C(i)(i=1,...,S), Θ_i

Step2 Elimination-dispersal loop: l=l+1

Step3 Reproduction loop: k=k+1

Step4 Chemotaxis loop: j=j+1

- For i =1,...,S take a chemotactic step for bacterium i as follows.

- Compute fitness function, J (i, j, k, l):

$$J(i,j,k,l) \quad = \quad J(i,j,k,l) + J_{cc}(\Theta^i(j,k,l), P(j,k,l)) \qquad (4.2)$$

  (i.e. add on the cell-to cell attractantrepellant profile to simulate the swarming behavior)

- Let

$$J_{last} \quad = \quad J(i,j,k,l) \qquad (4.3)$$

  to save this value since we may find a better cost via a run.

- Tumble: generate a random vector $\Delta(i)$ ε Rp with each element $\Delta_m(i)$,m=1,...,p, a random number on [-1, 1].

- Move: Let

$$\Theta_i(j+1,k,l) \quad = \quad \Theta_i(j,k,l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \qquad (4.4)$$

  This results in a step of size C(i) in the direction of the tumble for bacterium i.

- Compute J (i,j+1,k,l) and let

$$J(i,j,k,l) \quad = \quad J(i,j,k,l) + J_{cc}(\Theta^i(j,k,l), P(j,k,l)) \qquad (4.5)$$

- Swim

  - Let m=0 (counter for swim length).

  - While m $< N_s$ (if have not climbed down too long)

    (i) Let

$$m \quad = \quad m+1. \qquad (4.6)$$

(ii) If J(i,j+1,k,l)$< J_{last}$(if doing better), let

$$J_{last} = J(i, j + 1, k, l) \tag{4.7}$$

and let

$$\Theta_i(j + 1, k, l) = \Theta_i(j, k, l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \tag{4.8}$$

And use this Θ_i(j+1,k,l) to compute the new J(i,j+1,k,l)

(iii) Else, let This is the end of the while statement.

- Go to next bacterium (i+1) if i $\neq$ S

Step5 If j $< N_c$, go to step 4. In this case continue chemotaxis since the life of the bacteria is not over.

Step6 Reproduction:

- For the given k and l, and for each i = 1,..., S , let

$$J_{health}^i = \sum_{j=1}^{N_c+1} J(i, j, k, l) \tag{4.9}$$

be the health of the bacterium i (a measure of how many nutrients it got over its lifetime and how successful it was at avoiding noxious substances). Sort bacteria and chemotactic parameters C(i) in order of ascending cost J_{health} (higher cost means lower health).

- The S_r bacteria with the highest J_{health} values die and the remaining S_r bacteria with the best values split (this process is performed by the copies that are made are placed at the same location as their parent).

Step7 If k $< N_{re}$, go to step 3. In this case, we have not reached the number of specified reproduction steps, so we start the next generation of the chemotactic loop.

Step8 Elimination-dispersal: For i= 1,...,S with probability P_{ed} , eliminate and disperse each bacterium (this keeps the number of bacteria in the population constant). To do this, if a bacterium is eliminated, simply disperse another one to a random location on the optimization domain. If ed l N , then go to step 2; otherwise end.

4.2 Particle Swarm Optimization(PSO)

Particle Swarm Optimization is an algorithm capable of optimizing a non-linear and multidimensional problem which usually reaches good solutions efficiently while requiring minimal parameterization. The basic concept of the algorithm is to create a swarm of particles which move in the space around them (the problem space) searching for their goal, the place which best suits their needs given by a fitness function [9]. A nature analogy with birds is the following: a bird flock flies in its environment looking for the best place to rest. An example of PSO is shown in figure 4.1

4.2.1 PSO Algorithm

The PSO Algorithm [10] is stated as follows:

- Initialize the particle position by assigning location p = (p0,p1,...,pN) and velocities v = (v0,v1,...,vN).

- Determine the fitness value of all the particles: f(p) = (f(p0),f(p1),...,f(pN)).

- Evaluate the location where each individual has the highest fitness value so far: p = (p0best,p1best,...,pNbest).

- Evaluate the global fitness value which is best of all pbest :G(p) = max(f(p)).
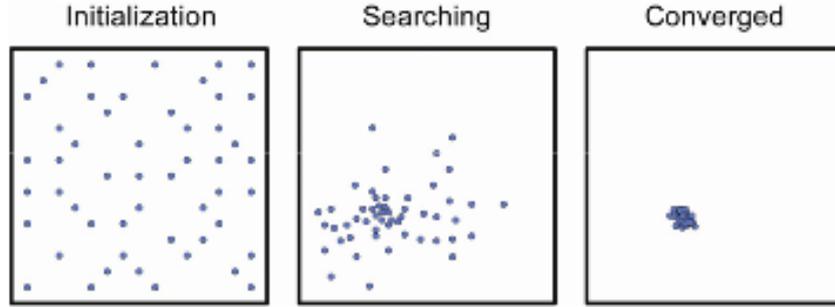
Figure 4.1: Example of a particle swarm optimisation swarms progression in two-dimensions. Initially, the particles are scattered randomly in the search space. As they search for the optimum value, the particles balance the objectives of following the value gradient and random exploration. Over time they begin to congregate in the general area of the optimum value. Finally, the particles converge to the optimum value

- The particle velocity is updated based on the pbest and gbest

$$vi^{new} = v1 + c1 \times rand() \times (pibest - pi) + c2 \times rand() \times (pgbest - pi) \quad (4.10)$$

  for $1 < i < N$. where c1 and c2 are constants known as acceleration coefficients and rand () are two separately generated uniformly distributed random numbers in the range [0, 1].

- Update the particle location by:

$$pi^{new} = pi + vi^{new} \quad (4.11)$$

  for $1 < i < N$.

- Terminate if maximum number of iterations is attained or minimum error criteria is met.

- Go to 2nd step.

4.2.2   Binary PSO

Consider a database of L subjects or classes, each class W1, W2,...,WL with N1,N2,...,NL number of samples. Let M1,M2,...,ML be the individual class mean and M0 be mean of feature vector. Fitness function is defined so as to increase the class separation equation. By minimizing the fitness function, class separation is increased. With each iteration, the most important features are selected. Binary value of 1 of its position implies that the feature is selected as a distinguishing feature for the succeeding iterations and if the position value is 0 the feature is not selected. The expressions for class, individual mean and mean of feature of feature vector are shown below:

$$Wj(i), for j \quad = \quad 1, 2, \ldots, Ni \tag{4.12}$$

$$Mi \quad = \quad \frac{1}{Ni} \sum_{j=1}^{Ni} Wj(i) \tag{4.13}$$

$$Mo \quad = \quad \frac{1}{N} \sum_{i=1}^{L} Ni \times Mi \tag{4.14}$$

Chapter 5

Face Recognition Using Swarm Intelligence

Swarm intelligence is a family of decentralized stochastic algorithms inspired by the behavior of swarms. Swarm intelligence algorithms include Particle Swarm Optimization (PSO) [26], Bacterial Foraging Optimization (BFO) [15], and Ant Colony Optimization (ACO). The ultimate goal of any of these optimization algorithms is to find the global best fitness as efficiently as possible. Since fitness/objective function calls are often the most resource-intensive component of an optimization algorithm, efficiency is often defined as using the fewest number of fitness function calls as possible, i.e., fastest convergence to the global optimum.

In this project, the face images of the Yale face database are used to generate the training set and the test set. The Yale face database composes of 165 face images, 11 different face images for each of 15 people. The training set is generated by 75 face images, 5 different images for each of 15 people. The test set is generated by rest of the images in the database.

Two different approaches are followed and a comparison of the two is made. They are described in the subsection below:

5.1 BFO-based Feature Selection Algorithm

Feature Extraction Perform PCA on the images to obtain the optimal bases before LDA. Then generate the eigenvectors as the feature vector set (which will be input to the BFO) through LDA.

Feature Selection Apply the BFO algorithm on the feature vector set as stated in section 4.1. Pick up the position of bacteria B with max (Jhealth) value. This position represents the best feature subset of the features defined in feature extraction step.

Classification Calculate the difference between the feature subset (obtained through feature selection) of each image of facial gallery and the test image with the help of Euclidean Distance defined below. The index of the image which has the smallest distance with the image under test is considered to be the required index. For an N-dimensional space, the Euclidean distance between two any points, pi and qi is given by:

$$D = \sum_{i=1}^{n} sqrt(pi - qi)^2 \tag{5.1}$$

Where pi (or qi) is the coordinate of p (or q) in dimension i.

5.2 PSO-based Feature Selection Algorithm

Feature Extraction Obtain the DCT array by applying Discrete Cosine Transformation to image. Take the most representative features of size 50 × 50 from upper left corner of DCT Array.

Feature Selection In this algorithm, each particle represents a range of possible candidate solutions. The evolution of each generation is accomplished through the fitness function. The fitness function is as follow:

$$F = \sqrt{\sum_{i=1}^{l}(Mi - Mo)(Mi - Mo)^T} \tag{5.2}$$

Where Mi represents number of classification and Mo represents the grand mean in the sample space.

The most representative features obtained through DCT processing are provided as input to PSO algorithm stated in section 4.2.1. PSO algorithm finds out the optimal set of coefficients.

Classification Euclidean distance classification is done to judge the category of each testing sample.

Chapter 6

Results & Analysis

6.1 Face Database

The two recognition approaches has been tested on publicly available YALE face database [11]. It comprises images from 15 subjects. For each subject, 11 different images are recorded, one for each variation such as different facial expression, center-light, with glasses, happy, left-light, with and without glasses, normal, right-light, sad, sleepy, surprised, and wink. In total, the database consists of 165 images.

6.2 Experiment & Results

In order to construct the training set, 7 images per person was used and the remaining 4 images were used for testing purpose. All 15 classes of persons in the database were considered. The same training and testing data sets were used in both the approaches. The results are displayed in table 6.1

Feature Extraction	Feature Selection	Training Time	Recognition Rate
PCA and LDA	BFO	257.13 sec	95.14%
DCT	PSO	161.86 sec	93.7%

Table 6.1: Results on Yale Database

6.3   Comparative Analysis

On comparing BFO-based approach with PSO-based approach for feature selection, it is found that:

- The average recognition rate of BFO is better than that of PSO-based feature selection. Also, on analysis it is found that number features required by BFO are less than that required for recognition using PSO.

- However, in terms of computational time, PSO-based selection algorithm takes less training time than the BFO-based selection algorithm. Hence, BFO is computationally expensive than PSO.

Therefore, it can be said that the effectiveness of BFO in finding the optimal feature subset compared to PSO compensates its computational inefficiency.

Chapter 7

Conclusion

In this thesis, face recognition was performed by application of the swarm optimization algorithms. It was found out that the underlying foraging principle and the swarm optimization can be integrated into evolutionary computational algorithms to provide a better search strategy for finding optimal feature vectors for face recognition. Finally, it is believed that the two swarm optimization methods, namely bacterial foraging optimization and particle swarm optimization, may be useful for the development of face recognition system.

Bibliography

[1] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Comput. Surv.*, 35(4):399–458, December 2003.

[2] Marios Savvides, Jingu Heo, and SungWon Park. Face recognition. In AnilK. Jain, Patrick Flynn, and ArunA. Ross, editors, *Handbook of Biometrics*, pages 43–70. Springer US, 2008.

[3] M. Kirby and L. Sirovich. Application of the karhunen-loeve procedure for the characterization of human faces. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(1):103–108, January 1990.

[4] F.Z. Chelali, A. Djeradi, and R. Djeradi. Linear discriminant analysis for face recognition. In *Multimedia Computing and Systems, 2009. ICMCS '09. International Conference on*, pages 1–10, 2009.

[5] B. Schwerin and K. Paliwal. Local-dct features for facial recognition. In *Signal Processing and Communication Systems, 2008. ICSPCS 2008. 2nd International Conference on*, pages 1–6, 2008.

[6] Rutuparna Panda, Manoj Kumar Naik, and B.K. Panigrahi. Face recognition using bacterial foraging strategy. *Swarm and Evolutionary Computation*, 1(3):138 – 146, 2011.

[7] S. Arivalagan and K. Venkatachalapathy. Article: Face recognition based on a hybrid meta-heuristic feature selection algorithm. *International Journal of Computer Applications*, 55(17):18–22, October 2012. Published by Foundation of Computer Science, New York, USA.

[8] Navdeep Kaur Rasleen Jakhar and Ramandeep Singh. Face recognition using bacteria foraging optimization-based selected features. *(IJACSA) International Journal of Advanced Computer Science and Applications,Special Issue on Artificial Intelligence*, 2011.

[9] Rabab M. Ramadan and Rehab F. Abdel. Face Recognition Using Particle Swarm Optimization-Based Selected Features. 2009.

[10] Guojian Cheng, Caiyun Shi, Kai Zhu, and Kevin Gong. The application of binary particle swarm algorithm in face recognition. In *Computational Intelligence and Security (CIS), 2011 Seventh International Conference on*, pages 1229–1233, 2011.

[11] `http://cvc.yale.edu/projects/yalefaces/yalefaces.html`.

[12] Matthew Turk and Alex Pentland. Eigenfaces for recognition. *J. Cognitive Neuroscience*, 3(1):71–86, January 1991.

[13] Peter N. Belhumeur, Joo P. Hespanha, and David J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection.

[14] Saeed Dabbaghchian, Masoumeh P. Ghaemmaghami, and Ali Aghagolzadeh. Feature extraction using discrete cosine transform and discrimination power analysis with a face recognition technology. *Pattern Recognition*, 43(4):1431 – 1440, 2010.

[15] K. M. Passino. Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Systems Magazine*, 22:52–67, 2002.

[16] M.S. Bartlett, Javier R. Movellan, and T.J. Sejnowski. Face recognition by independent component analysis. *Neural Networks, IEEE Transactions on*, 13(6):1450–1464, 2002.

[17] Xiaofei He, Shuicheng Yan, Yuxiao Hu, P. Niyogi, and Hong-Jiang Zhang. Face recognition using laplacianfaces. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(3):328–340, 2005.

[18] Chengjun Liu and Harry Wechsler. Evolutionary pursuit and its application to face recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(6):570–582, June 2000.

[19] Ming-Hsuan Yang. Kernel eigenfaces vs. kernel fisherfaces: Face recognition using kernel methods. In *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*, pages 215–220, 2002.

[20] J. Wright, A.Y. Yang, A. Ganesh, S.S. Sastry, and Yi Ma. Robust face recognition via sparse representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(2):210–227, 2009.

[21] M. Lades, J.C. Vorbruggen, J. Buhmann, J. Lange, C. Von Der Malsburg, R.P. Wurtz, and W. Konen. Distortion invariant object recognition in the dynamic link architecture. *Computers, IEEE Transactions on*, 42(3):300–311, 1993.

[22] L. Wiskott, J.-M. Fellous, N. Kuiger, and C. Von der Malsburg. Face recognition by elastic bunch graph matching. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):775–779, 1997.

[23] Timo Ahonen, Abdenour Hadid, and Matti Pietik?inen. Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):2037–2041, 2006.

[24] Jun Luo, Y. Ma, E. Takikawa, Shihong Lao, M. Kawade, and Bao-Liang Lu. Person-specific sift features for face recognition. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 2, pages II–593–II–596, 2007.

[25] A. F. Frangi J. Yang D. Zhang and J. Y. Yang. Two-dimensional pca: A new approach to appear-ance-based face representation and recognition.

[26] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 4, pages 1942–1948 vol.4, 1995.