

Face Recognition Using Bacteria Foraging Optimization-Based Selected Features

Rasleen Jakhar
M.Tech Student
Lovely Professional University
Punjab, India

Navdeep Kaur
Ex-Lecturer
RIEIT
Railmajra, Punjab, India

Ramandeep Singh
Assistant professor
Lovely Professional University
Punjab, India

Abstract— 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. This paper presents a novel feature selection algorithm based on Bacteria Foraging Optimization (BFO). The algorithm is applied to coefficients extracted by discrete cosine transforms (DCT). Evolution is driven by a fitness function defined in terms of maximizing the class separation (scatter index). Performance is evaluated using the ORL face database.

Keywords- Face Recognition; Bacteria Foraging Optimization; DCT; Feature Selection.

I. INTRODUCTION

A. Face Recognition

Face Recognition (FR) is a matching process between a query face's features and target face's features. Face recognition (FR) has emerged as one of the most extensively studied research topics that spans multiple disciplines such as pattern recognition, signal processing and computer vision. [1]. The block diagram of Face Recognition system is shown in figure 1.

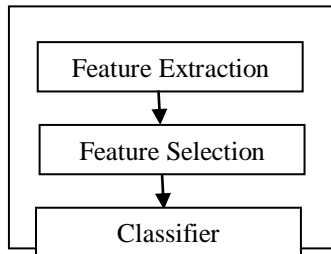


Figure 1: Face Recognition System

1) Feature Extraction

It is known that a good feature extractor for a face recognition system is claimed to select as more as possible the best discriminate features which are not sensitive to arbitrary environmental variations such as variations in pose, scale, illumination, and facial expressions [2].

Feature extraction algorithms mainly fall into two categories: geometrical features extraction and, statistical (algebraic) features extraction [1], [3], [4]. The geometrical approach represents the face in terms of structural measurements and distinctive facial features. These features are

used to recognize an unknown face by matching it to the nearest neighbor in the stored database. Statistical features extraction is usually driven by algebraic methods such as principal component analysis (PCA) [5], and independent component analysis (ICA) [5], [6], [7], [8], [9], [10], [11].

a) Discrete Cosine Transform

DCT has emerged as a popular transformation technique widely used in signal and image processing. This is due to its strong “energy compaction” property: most of the signal information tends to be concentrated in a few low-frequency components of the DCT. The use of DCT for feature extraction in FR has been described by several research groups [10], [11], [12], [13], [14], [15] and [16]. DCT transforms the input into a linear combination of weighted basis functions. These basis functions are the frequency components of the input data.

The general equation for the DCT of an $N \times M$ image $f(x, y)$ is defined by the following equation:

$$F(u, v) = \alpha(u)\alpha(v) \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \cos\left[\frac{\pi \cdot u}{2 \cdot N}(2x+1)\right] \cos\left[\frac{\pi \cdot v}{2 \cdot M}(2y+1)\right] f(x, y) \quad \dots (i)$$

Where $f(x, y)$ is the intensity of the pixel in row x and column y ; $u = 0, 1 \dots N-1$ and $v = 0, 1 \dots M-1$ and the functions $\alpha(u)$, $\alpha(v)$ are defined as:

$$\alpha(u), \alpha(v) = \begin{cases} \sqrt{\frac{1}{N}} & \text{for } u, v = 0 \\ \sqrt{\frac{2}{N}} & \text{for } u, v \neq 0 \end{cases} \quad \dots (ii)$$

2) Feature Selection

The feature selection seeks for the optimal set of d features out of m [17], [18] and [19]. Several methods have been previously used to perform feature selection on training and testing data. Among the various methods proposed for FS, population-based optimization algorithms such as Genetic Algorithm (GA)-based method [20], [21], [22] and Ant Colony Optimization (ACO)-based method have attracted a lot of attention [23]. In the proposed FR system we utilized an evolutionary feature selection algorithm based on swarm intelligence called the Bacteria Foraging Optimization.

B. Bacteria Foraging Optimization

Bacterial Foraging Optimization (BFO) is a novel optimization algorithm based on the social foraging behavior of

E. coli bacteria. The motile bacteria such as *E. coli* and salmonella propel themselves by rotating their flagella. To move forward, the flagella counterclockwise rotate and the organism “swims” (or “runs”). While a clockwise rotation of the flagellum causes the bacterium randomly “tumble” itself in a new direction and then swims again [24], [25].

1) Classical BFO Algorithm

The original Bacterial Foraging Optimization system consists of three principal mechanisms, namely, chemo taxis, reproduction, and elimination-dispersal [25]:

a) Chemo taxis

Suppose $\theta^i(j, k, l)$ represents the bacterium at j th chemo tactic, k th reproductive, and l th elimination-dispersal step. $C(i)$ is the chemo tactic step size during each run or tumble (i.e., run-length unit). Then in each computational chemo tactic step, the movement of the i th bacterium can be represented as

$$\theta^i(j+1, k, l) = \theta^i(j, k, l) + C(i) \frac{\Delta(i)}{\sqrt{\Delta^T(i) \Delta(i)}} \quad \dots \text{(iii)}$$

Where $\Delta(i)$ is the direction vector of the j th chemo tactic step. When the bacterial movement is *run*, $\Delta(i)$ is the same with the last chemo tactic step; otherwise, $\Delta(i)$ is a random vector whose elements lie in $[-1, 1]$. With the activity of run or tumble taken at each step of the chemo taxis process, a step fitness, denoted as $J(i, j, k, l)$, will be evaluated.

b) Reproduction

The health status of each bacterium is calculated as the sum of the step fitness during its life, that is, $\sum_{j=1}^{Nc} J(i, j, k, l)$, where Nc is the maximum step in a chemo taxis process. All bacteria are sorted in reverse order according to health status. In the reproduction step, only the first half of population survives and a surviving bacterium splits into two identical ones, which are then placed in the same locations. Thus, the population of bacteria keeps constant.

c) Elimination and Dispersal

The chemo taxis provides a basis for local search, and the reproduction process speeds up the convergence which has been simulated by the classical BFO. While to a large extent, only chemo taxis 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 position to eliminate the accidents of being trapped into the local optima. Then some bacteria are chosen, according to a preset probability Ped , to be killed and moved to another position within the environment.

The original BFO algorithm is briefly outlined step by step as follows.

Step1. Initialize parameters $n, S, Nc, Ns, Nre, Ned, Ped, C(i)$ ($i = 1, 2, \dots, S$), θ_i where

- n : dimension of the search space,
- S : the number of bacteria in the colony,
- Nc : chemo tactic steps,
- Ns : swim steps,

- Nre : reproductive steps,
- Ned : elimination and dispersal steps,
- Ped : probability of elimination,
- $C(i)$: the run-length unit (i.e., the size of the step taken in each run or tumble).
- θ^i : position of i th bacteria

Step2. Elimination-dispersal loop: $l = l + 1$.

Step3. Reproduction loop: $k = k + 1$.

Step4. Chemo taxis loop: $j = j + 1$.

Sub step 4.1. For $i = 1, 2, \dots, S$, take a chemo tactic step for bacterium i as follows:

Sub step 4.2. Compute fitness function, $J(i, j, k, l)$.

Sub step 4.3. Let $J_{last} = J(i, j, k, l)$ to save this value since we may find better value via a run.

Sub step 4.4. Tumble. Generate a random vector $\Delta(i) \in \mathbb{R}^n$ with each element $\Delta m(i)$, $m = 1, 2, \dots, n$, a random number on $[-1, 1]$.

Sub step 4.5. Move. Let

$$\theta^i(j+1, k, l) = \theta^i(j, k, l) + C(i) \frac{\Delta(i)}{\sqrt{\Delta^T(i) \Delta(i)}} \quad \dots \text{(iv)}$$

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

Sub step 4.6. Compute $J(i, j+1, k, l)$ with $\theta^i(j+1, k, l)$.

Sub step 4.7. Swimming

(i) Let $m = 0$ (counter for swim length).

(ii) While $m < Ns$ (if has not climbed down too long), the following hold.

- Let $m = m + 1$.
- If $J(i, j+1, k, l) < J_{last}$, let $J_{last} = J(i, j+1, k, l)$, then another step of size $C(i)$ in this same direction will be taken as (iv) and use the new generated. $\theta^i(j+1, k, l)$ to compute the new $J(i, j+1, k, l)$.
- Else let $m = Ns$.

Sub step 4.8. Go to next bacterium ($i+1$). if $i \neq S$, go to Sub step 4.2 to process the next bacterium.

Step5. If $j < Nc$, go to Step 3. In this case, continue chemo taxis since the life of the bacteria is not over.

Step6. Reproduction

Sub step 6.1. For the given k and l , and for each $i = 1, 2, \dots, S$, let

$$J_{health}^i = \sum_{j=1}^{Nc+1} J(i, j, k, l) \quad \dots \text{(v)}$$

be the health of the bacteria. Sort bacteria in order of ascending values (J_{health})

Sub step 6.2. The Sr bacteria with the highest J_{health} values die and the other Sr bacteria with the best values split and the copies that are made are placed at the same location as their parent.

Step7. If $k < Nre$, go to Step 2. In this case the number of specified reproduction steps is not reached and start the next generation in the chemotactic loop.

Step8. Elimination-dispersal: for $i = 1, 2, \dots, S$, with probability Ped , eliminate and disperse each bacterium, which results in keeping the number of bacteria in the population constant. To do this, if a bacterium is eliminated, simply disperse one to a random location on the optimization domain. If $l < Ned$, then go to Step 2; otherwise end.

II. BFO- BASED FEATURE SELECTION

In this proposed work, features of image are extracted using DCT technique. The extracted features are reduced further by using Bacteria Foraging Optimization to remove redundancy and irrelevant features. The resulting feature subset (obtained by BFO) is the most representative subset and is used to recognize the face from face gallery.

A. Bacteria Representation

Each bacteria's position represent one possible solution (feature subset) required for face recognition. The number of dimensions of search space is m where m is the length of feature vector extracted by DCT. In each dimension of search space, bacteria position is 1 or 0, where 1 or 0 indicates that this feature is selected or not selected, respectively, as required feature for next generation. In the each iteration of chemo taxis step, each bacteria tumbles to the new random position. Position of i th bacteria in j th chemo taxis and k th reproduction step is defined as:

$$\Theta^i(j, k) = F_1 F_2 \dots F_m \quad \dots(vi)$$

Where, m is the length of feature vector extracted by DCT. Each $F_z = 1$ or 0 ($z=1,2,\dots,m$) Depending upon whether z th feature is selected or not for the next iteration.

1) Fitness Function

In each generation, each bacterium is evaluated, and a value of *goodness* or *fitness* is returned by a fitness function. This evolution is driven by the fitness function F [26]. Let w_1, w_2, \dots, w_L and N_1, N_2, \dots, N_L denote the classes and number of images within each class, respectively. Let M_1, M_2, M_L and M_0 be the means of corresponding classes and the grand mean in the feature space, M_i can be calculated as:

$$M_i = \frac{1}{N_i} \sum_{j=1}^{N_i} W_j^{(i)}, \quad i = 1, 2, \dots, L \quad \dots (vii)$$

Where $W_j^{(i)}$, $j=1,2,\dots,N_i$, represents the sample image from class w_i and grand mean M_0 is:

$$M_0 = \frac{1}{N} \sum_{i=1}^L N_i M_i \quad \dots (viii)$$

Where N is the total number of images of all the classes. Thus the between class scatter fitness function F is computed as follows:

$$F = \sqrt{\sum_{i=1}^L (M_i - M_0)' (M_i - M_0)} \quad \dots (ix)$$

2) Classifier

After the training phase, a typical and popular Euclidean distance is employed to measure the similarity between the test vector and the reference vectors in the gallery. Euclidean distance is defined as the straight-line distance between two points. For N -dimensional space, the Euclidean distance between two any points' p_i and q_i is given by:

$$D = \sqrt{\sum_{i=1}^N (p_i - q_i)^2} \quad \dots(x)$$

Where p_i (or q_i) is the coordinate of p (or q) in dimension i .

III. PROPOSED BFO-BASED FEATURE SELECTION ALGORITHM

The algorithm proposed for feature extraction using BFO is discussed in figure 2. There are certain variations in BFO algorithm used in this work. Firstly, step 6.6 of the proposed algorithm moves the bacteria back to its previous position if current position is less suitable (checked using fitness function). So in this algorithm, bacteria have "memory" as they remember their previous position. Secondly, as there are chances that bacteria may get stuck in local optima, elimination dispersal removes bacteria from its current position and moves it to "random" new position. In the proposed algorithm, position of bacteria is decided randomly in the each iteration. There is no need of using Elimination Dispersal.

IV. EXPERIMENTAL RESULTS

The performance of the proposed feature selection algorithm is evaluated using the standard Cambridge ORL gray-scale face database. The ORL database of faces contains a set of face images taken between April 1992 and April 1994 at the AT&T Laboratories (by the Oliver Research Laboratory in Cambridge, UK) [13] and [23].

The database is composed of 400 images corresponding to 40 distinct persons. The original size of each image is 92x112 pixels, with 256 grey levels per pixel. Each subject has 10 different images taken in various sessions varying the lighting, facial expressions (open/ closed eyes, smiling/ not smiling) and facial details (glasses/ no glasses). Four images per person were used in the training set and the remaining six images were used for testing.

The parameters used for BFO-based Feature Selection is shown in table 1.

TABLE I. BFO PARAMETER SETTING

Bacteria Size(S)	30
Number of chemo taxis steps (Nc)	10
Number of reproduction steps (Nre)	10

In this work, we test the BFO-based feature selection algorithm with feature vectors based on various sizes of DCT coefficient. The 2-dimensional DCT is applied to the input image and only a subset of the DCT coefficients corresponding to the upper left corner of the DCT array is retained. Subset sizes of 50x50, 40x40, 30x30 and 20x20 of the original 92x112 DCT array are used in this work.

1. Feature Extraction: Obtain the DCT array by applying Discrete Cosine Transformation to image.
2. Take the most representative features of size $n \times n$ from upper left corner of DCT Array.
3. Feature Selection: Define the BFO parameters: S, N_c, N_{re}, n
Where S : Number of bacteria in the colony
 N_c : chemo tactic steps
 N_{re} : reproductive steps
 n : dimension of the search space
4. Place each bacteria at random position.
5. (Reproduction step $k+1$) For $k = k + 1$
6. (Chemo taxis step $j+1$) For $j = j + 1$
 - 6.1 For $i = 1, 2, \dots, S$, take chemotaxis step for bacteria i as follows:
 - 6.2 Compute the fitness function of the previous (j^{th}) chemo taxis step as defined in formula (vii), (viii), (ix).
 - 6.3 (Tumble) Tumble to random new position, $\theta^i(j+1, k)$, as defined in formula (vi).
 - 6.4 (Move) Move the bacteria to new position
 $\theta^i(j+1, k)$.
 - 6.5 Compute $J(i, j+1, k)$ using $\theta^i(j+1, k)$ as in step 6.2.
 - 6.6 If $J(i, j+1, k) < J(i, j, k)$ then:
 - (i) (Move Back) Move bacteria back to its previous position:
 $\theta^i(j+1, k) = \theta^i(j, k)$
 - (ii) Update Fitness function
 $J(i, j+1, k) = J(i, j, k)$
 - 6.7 Go to next bacteria $i + 1$ (for of step 6.1 ends)
 - 6.8 Store the current fitness of i^{th} bacteria in $J_c(i)$. (chemo taxis loop of step 6 ends).
7. (reproduction step) for given k and for each $i = 1, 2, \dots, S$, Let
 $J_{health}^i = J_c(i)$
be the health of bacteria. Sort the bacteria in descending order of J_{health} .
8. The bacteria with S_r lowest J_{health} values die and other bacteria with S_r best J_{health} values are split and copies that are made are placed at the same location as their parents. (reproduction loop of step 5 ends)
9. Pick up the position of bacteria B with $\max(J_{health})$ value. This position represents the best feature subset of the features defined in step 2. (Feature Selection Ends)
10. Classification: calculate the difference between the feature subset (obtained in step 9) of each image of facial gallery and the test image with the help of Euclidean Distance defined in Formula (v). The index of the image which has the smallest distance with the image under test is considered to be the required index.

Figure 2: Face Recognition using BFO based Feature Selection

Each of 2- dimensional subset DCT array is converted to a 1-dimensional array using raster scan. This is achieved by processing the image row by row concatenating the consecutive rows into a column vector. This column vector is

the input to the subsequent feature selection algorithm.

To calculate average recognition rate for each problem instance, 5 test images are randomly chosen from 40 classes. Average recognition is measured by knowing how many times correct faces were identified out of 5 trials. The average recognition rate is measured together with the CPU training time and the average number of selected features for each problem instance. The algorithm has been implemented in Matlab 7 and the result for each problem instance (20X20, 30X30, 40X40, and 50X50 DCT Array) is shown in table 2

TABLE II. RESULTS OF BFO-FS ALGORITHM

DCT Feature Vector Size	Number of Features input to BFO-FS	Average no. of features selected by BFO-FS	Training time (in seconds)	Average Recognition Rate
20X20	400	215	102.610	100%
30X30	900	454	132.531	100%
40X40	1600	807	209.969	100%
50X50	2500	1281	272.110	100%

Following are the faces recognized by the proposed Algorithm for various number of features input to BFO-FS.

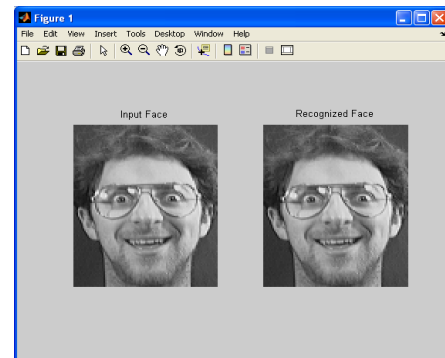


Figure: 3 Input Face and the Recognized Face for DCT Feature vector of 20X20

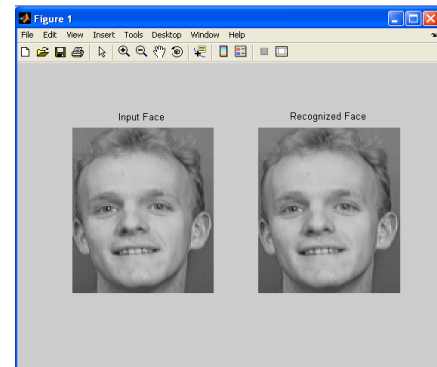


Figure: 4 Input Face and the Recognized Face for DCT Feature vector of 30X30

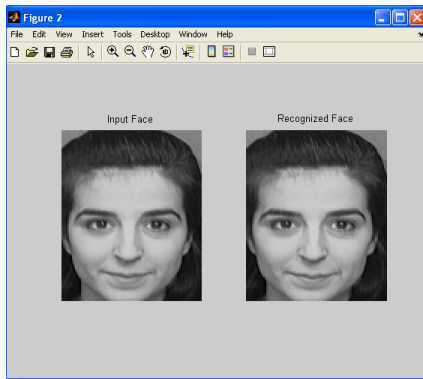


Figure: 5 Input Face and the Recognized Face for DCT Feature vector of 40X40

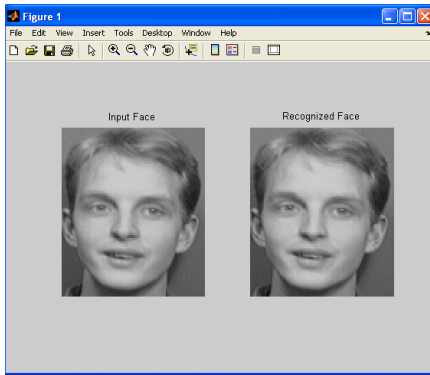
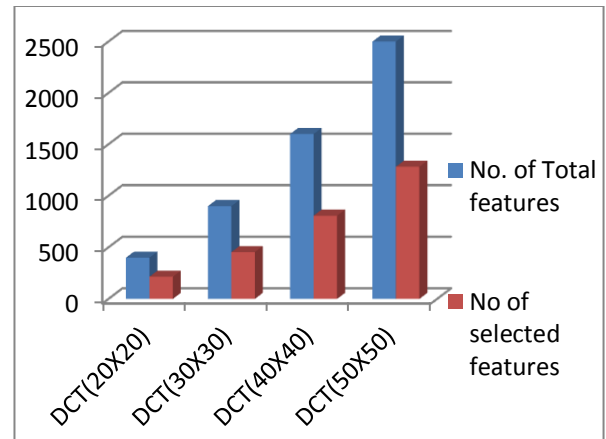


Figure: 6 Input Face and the Recognized Face for DCT Feature vector of 50X50

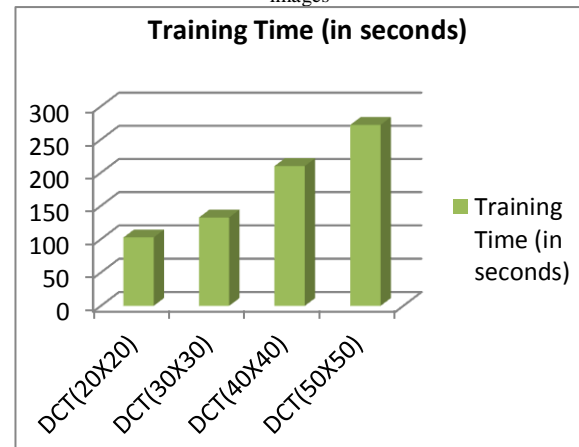
For each of the problem instance (20X20, 30X30, 40X40, and 50X50), algorithm is run 5 times and each time, random test image is chosen to be matched with face gallery. The test face matches with image in face gallery in each trial and average recognition rate is 100 % for each problem instance. The BFO-selection algorithm reduces the size of original feature vector to 53.7%, 50%, 50%, 51% for problem instance of 20X20, 30X30, 40X40, and 50X50 respectively. For example, if the DCT of an image is calculated and 20X20 DCT subset is taken from upper left of DCT array, there are total 400 features which are given as an input to BFO-FS algorithm. BFO-FS reduces the 400 features to 215 which means only 215 features are required to recognize the face from facial gallery.

A. Comparison of BFO with PSO

If the proposed algorithm is compared with PSO-based feature selection described in [2], the average recognition rate of the proposed algorithm is better than that of PSO-based feature selection. The number of selected features by proposed algorithm is comparable to those selected by PSO-based feature selection. On the other hand, in terms of computational time, PSO-based selection algorithm takes less training time than the BFO-based selection algorithm in all tested instances which indicates that BFO is computationally expensive than PSO but the effectiveness' of BFO in finding the optimal feature subset compared to PSO compensates its computational inefficiency.



Graph: 1 showing the total no of features and the selected features for various images



Graph2: Showing the training time for different images

V. CONCLUSION

In this paper, a novel BFO-based feature selection algorithm for FR is proposed. The algorithm is applied to feature vectors extracted by Discrete Cosine Transform. The algorithm is utilized to search the feature space for the optimal feature subset. Evolution is driven by a fitness function defined in terms of class separation. The classifier performance and the length of selected feature vector were considered for performance evaluation using the ORL face database. Experimental results show the superiority of the BFO-based feature selection algorithm in generating excellent recognition accuracy with the minimal set of selected features.

REFERENCES

- [1] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face Recognition: A Literature Survey," *ACM Computing Surveys*, vol. 35, no. 4, pp. 399-458, 2003.
- [2] R.M. Ramadan and R. F. Abdel - Kader, "Face Recognition Using Particle Swarm Optimization - Based Selected Features", *International Journal of Signal Processing, Image Processing and Pattern Recognition*, Vol. 2, No. 2, June 2009.
- [3] R. Brunelli and T. Poggio, "Face Recognition: Features versus Templates," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, no. 10, pp. 1042-1052, 1993.

- [4] X. Yi-qiong, L. Bi-cheng and W. Bo, "Face Recognition by Fast Independent Component Analysis and Genetic Algorithm," *Proc. Of the 4th International Conference on Computer and Information Technology (CIT'04)*, pp. 194-198, Sept. 2004.
- [5] M. A. Turk and A. P. Pentland, "Face Recognition using Eigenfaces," *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 586-591, June 1991.
- [6] H. R. Wilson, D. Levi, L. Maffei, J. Rovamo, and R. DeValois, "The Perception of Form: Retina to Striate Cortex", *Visual Perception: The Neurophysiological Foundations*, Academic Press, 1990.
- [7] S. Chien, and I. Choi, "Face and Facial Landmarks location based on Log-Polar Mapping", *Lecture Notes in Computer Science – LNCS 1811*, pp. 379-386, 2000.
- [8] S. Minut, S. Mahadevan, J. Henderson, and F. Dyer, "Face Recognition using Foveal Vision", *Lecture Notes in Computer Science – LNCS 1811*, pp. 424-433, 2000.
- [9] M. Tistarelli, and E. Grosso, "Active Vision-Based Face Authentication", *Image and Vision Computing*, no. 18, pp. 299-314, 2000.
- [10] A. S. Samra, S. E. Gad Allah, R. M. Ibrahim, "Face Recognition Using Wavelet Transform, Fast Fourier Transform and Discrete Cosine Transform," *Proc. 46th IEEE International Midwest Symp. Circuits and Systems (MWSCAS'03)*, vol. 1, pp. 272- 275, 2003.
- [11] Z. Yankun and L. Chongqing, "Efficient Face Recognition Method based on DCT and LDA", *Journal of Systems Engineering and Electronics*, vol. 15, no. 2, pp. 211-216, 2004.
- [12] C. Podilchuk and X. Zhang, "Face Recognition Using DCT-Based Feature Vectors," *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'96)*, vol. 4, pp. 2144-2147, May 1996.
- [13] F. M. Matos, L. V. Batista, and J. Poel, "Face Recognition Using DCT Coefficients Selection," *Proc. of the 2008 ACM Symposium on Applied Computing, (SAC'08)*, pp. 1753-1757, March 2008.
- [14] M. Yu, G. Yan, and Q.-W. Zhu, "New Face recognition Method Based on DWT/DCT Combined Feature Selection," *Proc. 5th International Conference on Machine Learning and Cybernetics*, pp. 3233-3236, August 2006.
- [15] Z. Pan and H. Bolouri, "High Speed Face Recognition Based on Discrete Cosine Transform and Neural Networks," Technical Report, Science and Technology Research Center (STRC), University of Hertfordshire.
- [16] Z. M. Hafeed and M. D. Levine, "Face Recognition Using Discrete Cosine Transform", *International Journal of Computer Vision*, vol. 43, no. 3, pp. 167-188, 2001
- [17] C.-J. Tu, L.-Y. Chuang, J.-Y. Chang, and C.-H. Yang, "Feature Selection using PSO-SVM," *International Journal of Computer Science (IAENG)*, vol. 33, no. 1, IJCS_33_1_18.
- [18] E. Kokiopoulou and P. Frossard, "Classification-Specific Feature Sampling for Face Recognition," *Proc IEEE 8th Workshop on Multimedia Signal Processing*, pp. 20-23, 2006.
- [19] A. Y. Yang, J. Wright, Y. Ma, and S. S. Sastry, " Feature Selection in Face Recognition: A Sparse Representation Perspective," submitted for publication, 2007.
- [20] X. Fan and B. Verma, "Face recognition: a new feature selection and classification technique," *Proc. 7th Asia-Pacific Conference on Complex Systems*, December 2004.
- [21] D.-S. Kim, I.-J. Jeon, S.-Y. Lee, P.-K. Rhee, and D.-J. Chung, "Embedded Face Recognition based on Fast Genetic Algorithm for Intelligent Digital Photography," *IEEE Trans. Consumer Electronics*, vol. 52, no. 3, pp. 726-734, August 2006.
- [22] M. L. Raymer, W. F. Punch, E. D. Goodman, L.A. Kuhn, and A. K Jain, "Dimensionality Reduction Using Genetic Algorithms," *IEEE Trans. Evolutionary Computation*, vol. 4, no. 2, pp. 164-171, July 2000.
- [23] H. R. Kanan, K. Faez, and M. Hosseinzadeh, "Face Recognition System Using Ant Colony Optimization-Based Selected Features," *Proc. IEEE Symp. Computational Intelligence in Security and Defense Applications (CISDA 2007)*, pp 57-62, April 2007.
- [24] J. Adler (1966), "Chemotaxis in bacteria," *Science*, vol. 153, pp. 708–716.
- [25] H. Chen, Y. Zhu and R. Hu, " Cooperative Bacterial Foraging Optimization", Research Article, Key Laboratory of Industrial Informatics, Shenyang Institute of Automation, Chinese Academy of Sciences, China., 2009.
- [26] C. Liu and H. Wechsler, "Evolutionary Pursuit and Its Application to Face Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 6, pp. 570-582, 2000.