



# Comparitive Study of Face Recognition using ANN and BFO Technique

Shiwani, Dr. Kamal Sharma<sup>1</sup>, Er. Kaushal Kisore<sup>2</sup>

Department of ECE, Emax Group of Institutes, Kurukshetra University, Kurukshetra, Haryana,

**Abstract:** The objective of face recognition involves the extraction of different features of the human face from the face image for discriminating it from other persons. It is the problem of searching a face in reference database to find the matches as a given face. The purpose is to find a face that has highest similarity with a given face in the database. Many face recognition algorithms have been developed and used as an application of access control and surveillance. This paper presents a novel feature selection algorithm based on Bacteria Foraging Optimization (BFO) and then compared the results with the ANN with respect to its accuracy in recognizing a face. BFOA has been widely accepted as a global optimization algorithm of current interest for distributed optimization and control. BFOA is inspired by the social foraging behavior of *Escherichia coli*. BFOA has already drawn the attention of researchers because of its efficiency in solving real-world optimization problems arising in several application domains. In this paper face recognition is done by an efficient way using bacterial foraging optimization which is an optimization technique inspired from *E. coli* bacteria. Hence, this technique helps to reduce the computation complexity and time consuming.

**Keywords:** PCA, ANN, BFOA, rate of swim, rate of elimination

## 1. 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.

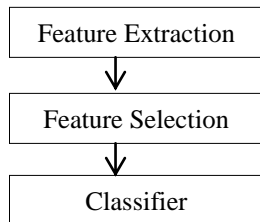


Figure 1: Face Recognition System

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.

Feature extraction algorithms mainly fall into two categories: geometrical features extraction and, statistical (algebraic) features extraction. 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) and independent component analysis (ICA).

In this approach PCA is used for the feature extraction

### 1. Principal Component Analysis

The PCA algorithm is implemented for the extraction of image feature and dimensionality reduction. We assume that  $M$  sample images are being used. Each sample image is referred to as  $A_n$  where  $n$  indicates that we are dealing with  $n$ th sample image ( $1 < n < M$ ). Each  $A_n$  should be a column vector. Images are made of pixels, each having  $(x, y)$  coordinates with  $(0, 0)$  being at the upper left corner. Following are the steps involved in PCA:

Step 1: The size of the resulting  $A_n$  column vector will depend on the size of the sample images. If the sample images are  $x$  pixels across and  $y$  pixels tall, the column vector will be of size  $(x * y) \times 1$ .

Step 2: Calculate the average image,  $\bar{O}$ , as follows. This average image will be a column vector of the same size as the sample images  $((x * y) \times 1)$ .

$$\bar{O} = \sum A_n / M \quad \text{where } 1 < n < M$$

Step 3: Calculate the difference faces by subtracting the average face from each sample image. Each will be a column vector the same size as our sample image vectors  $((x * y) \times 1)$

$$O = A_n - \bar{O}$$

Step 4: Total scatter matrix or covariance matrix is calculated from  $\bar{O}$ . The covariance matrix is defined by  $AA^T$  where  $A$  is

$$A = [O_1 \ O_2 \ O_3 \ \dots \ O_m]$$

The matrix  $A$  will be of size  $(x * y) \times M$ .

### 2) Feature Selection

The feature selection seeks for the optimal set of  $d$  features out of  $m$ . 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 and Ant Colony Optimization (ACO)-based method have attracted a lot of attention. In the proposed FR system we utilized an evolutionary feature selection algorithm based on swarm intelligence called the Bacteria Foraging Optimization.

### 2. Bacteria Foraging Optimization

Bacterial Foraging Optimization (BFO) is a population-based numerical 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. BFO system consists of three principal mechanisms, namely, chemo taxis, reproduction, and elimination-dispersal.

1. **Chemotaxis:** This process simulates the movement of an *E. coli* cell through swimming and tumbling via flagella. Biologically an *E. coli* bacterium can move in two different ways. It can swim for a period of time in the same direction or it may tumble, and alternate between these two modes of operation for the entire lifetime. Suppose  $(j, k, l)$  represents  $i$ -th bacterium at  $j$ th chemotactic,  $k$ -th reproductive and  $l$ -th elimination-dispersal step.  $C(i)$  is the size of the step taken in the random direction specified by the tumble (run length unit). Then in computational chemotaxis the movement of the bacterium may be represented by:

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

Where  $\Delta$  indicates a vector in the random direction whose elements lie in  $[-1, 1]$ .

2. **Reproduction:** The health status of each bacterium is calculated as the sum of the step fitness during its life, that is  $\sum_{j=1}^{N_c} J(i, j, k, l)$ , where  $N_c$  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.

3. **Elimination and Dispersal:** Gradual or sudden changes in the local environment where a bacterium population lives may occur due to various reasons e.g. a significant local rise of temperature may kill a group of bacteria that are currently in a region with a high concentration of nutrient gradients. Events can take place in such a fashion that all the bacteria in a region are killed or a group is dispersed into a new location. To simulate this phenomenon in BFOA some bacteria are liquidated at random with a very small probability while the new replacements are randomly initialized over the search space.

### II. BFO BASED ALGORITHM

BFO Algorithm is briefly outlined as follows:

Step 1: Initialise parameters  $p, S, N_c, N_s, N_{re}, N_{ed}, P_{ed}, C(i)(i=1, 2, \dots, S), \theta^T$ .

Step 2: Elimination-dispersal loop:  $l=l+1$

Step 3: Reproduction loop:  $k=k+1$

Step 4: Chemotaxis loop:  $j=j+1$

[a] For  $i=1, 2, \dots, S$  take a chemotactic step for bacterium  $i$  as follows.

[b] Compute fitness function,  $J(i, j, k, l)$ . Let,

$$J(i, j, k, l) = J(i, j, k, l) + J_{cc}(\theta^T(j, k, l)P(j, k, l)).$$

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

[d] Tumble: generate a random vector

$\Delta(i) \in \mathcal{R}$  with each element  $\Delta_m(i), m=1, 2, \dots, p$  a random number on  $[-1, 1]$ .

[e] Move: Let

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

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

[f] Compute  $J(i, j+1, k, l)$  and let

$$J(i, j, k, l) = J(i, j, k, l) + J_{cc}(\theta^T(j, k, l)P(j, k, l)).$$

[g] Swim

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

ii) While  $m < S$  (if have not climbed down too long).

• Let  $m=m+1$ .

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

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

And use this  $\theta^i(j+1, j, k)$  to compute the new  $J(i, j+1, k, l)$ .

Step 5: If  $c_j < N$ , go to step 4. In this case continue chemotaxis since the life of the bacteria is not over.

Step 6: Reproduction:

[a] For the given  $k$  and  $l$ , and for each  $i = 1, 2, \dots, S$ , let

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

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).

[b] The  $S_r$  bacteria with the highest  $J_{health}$  values die and the remaining  $S_r$  bacteria with the best values split.

Step 7: 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.

Step 8: Elimination-dispersal: For  $i = 1, 2, \dots, S$  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  $l < N_{ed}$ , then go to step 2; otherwise end.

Step 9: Pick up the position of bacteria B with  $max J$  (health) value. This position represents the best feature subset of the features.

Step10: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. The index of the image which has the smallest distance with the image under test is considered to be the required index.

**III.EXPERIMENTAL RESULTS and ANALYSIS**

The system proposed here first select the image from the given dataset of the face images. After the image is selected segmentation is done followed by the thresholding. Then the feature extraction is performed using BFO and finally the matching is done to obtain the desired output. We have used two generally used database for face recognition i.e. ORL database and FVC database. ORL database include 400 face images taken from 40 subjects, with each subjects providing ten face images. For some subjects, the images were taken at different times, with varying lighting, facial expressions (open /closed eyes, smiling/not smiling) and facial details (glasses/ no glasses). Each face image from the ORL database has been resized to a  $40 \times 56$  matrix by using the down sampling algorithm. In this some images are used for training samples and the remaining serves as the test samples. FVC database takes some images as training samples and the remaining images are designated as the test samples. All images were cropped and resized to  $92 \times 112$  pixels. All the simulation results of the above steps are shown in this section.



Figure 1: Training Database

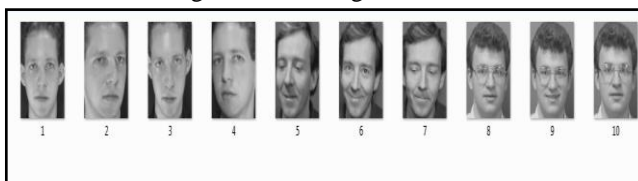


Figure 2: Testing Database

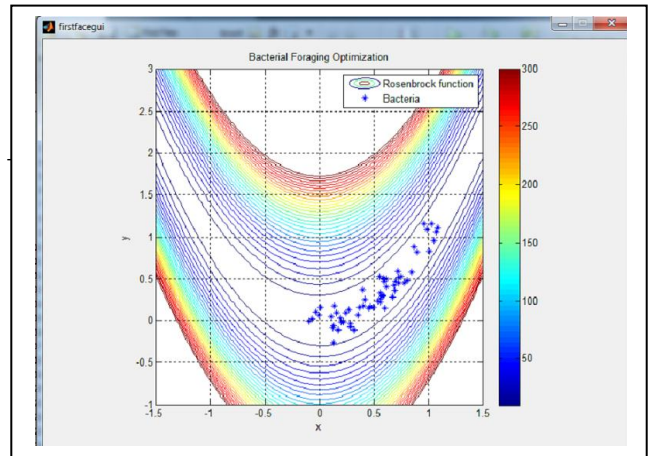


Figure 3: Bacterial swarm on multi-modal objective surface function

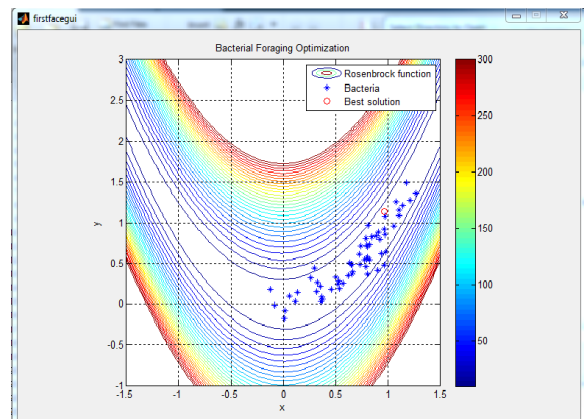
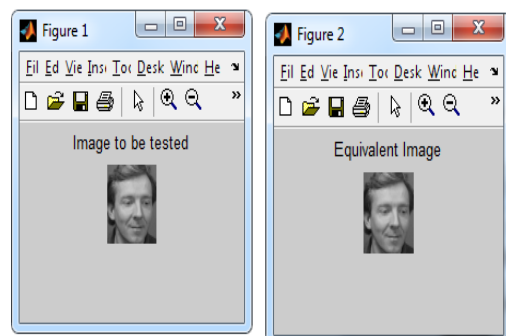


Figure 4: BFO algorithm giving optimum solution after foraging



a) b)

Figure 5: a) Input image b) Output image

To calculate average recognition rate for each problem instance, 4 test images are randomly chosen from 40 classes and the results are then compared with the ANN. The comparative results of recognition rate and elapsed time of ANN and BFO is shown in the following table.



	Elapsed time in sec	Recognition Rate in %
ANN	33.608	97.25
	32.536	94.00
	30.691	86.60
	34.720	94.17
	147.103	100.00
BFO	128.270	100.00
	130.091	100.00
	128.121	100.00

The table shown above clearly indicates that if the proposed algorithm is compared with ANN-based feature selection described in, the average recognition rate of the proposed algorithm is better than that of ANN-based feature selection. The number of selected features by proposed algorithm is comparable to those selected by ANN-based feature selection. On the other hand, in terms of computational time, ANN-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 ANN but the effectiveness' of BFO in finding the optimal feature subset compared to ANN compensates its computational effectiveness.

**IV. CONCLUSION AND FUTURE SCOPE**

This paper proposed a novel BFO-based feature selection algorithm for face recognition. The algorithm is applied to feature extracted by a widely used technique known as PCA. Bacterial Foraging Algorithm is based on a computational intelligence technique that is not largely affected by the size and non-linearity of the problem and has converged to the optimal solution to many problems where the most analytical methods fail to converge 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 face database. Experimental results show the superiority of the BFO-based feature selection algorithm in generating excellent recognition accuracy with less computational burden.

The research can be further extended to implement a face recognition technique using a hybrid approach of two widely used population-based optimization algorithm i.e. BFO and PSO.

**REFERENCES**

[1]Manzoor Ahmad Lone, S. M. Zakariya and Rashid Ali, “Automatic Face Recognition System by Combining Four Individual Algorithms”, International Conference on Computational Intelligence and Communication Systems IEEE-2011, pp. 222-226.

[2]HengFuiLiau ,HengFuiLiau , KahPhooi Seng, “A Multiview Face Recognition System Based on Eigenface Method”, IEEE-2008, pp. 431-436.

[3]. 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

[4] 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.

[5]Yoshihiko Nankaku and Keiichi Tokuda, “Face Recognition Using Hidden Markov EigenfaceModels”, IEEE-2007, pp.269 272

[6]SoumitraKar, Swati Hiremath, and Dilip.G, “A Multi Algorithmic Face Recognition System”, IEEE-2006, pp.321-326.

[7] Wei-Shi Zheng, Jian-Huang Lai, and Pong C, “A New LDA Based Face Recognition Algorithm With Selection of Principal Components”, IEEE-2005, pp. 230-234.

[8] Dr. Hassan Fahmy Hashem, “Comparison BetweenEigenface and Wavelet Technique as Methods of Face Recognition”, IEEE-2004, pp. 367-372.

[9]HimanshuS.Bhatt, Samarth Bhardwaj, RichaSingh and MayankVatsa, “Recognizing Surgically Altered Face Images Using MultiobjectivEvolutionary Algorithm”, IEEE-2013, pp. 89-100.

[10]StefanosZafeiriou, Gary A. Atkinson, Mark F. Hansen, William A.P. Smith ,VasileiosArgyriou , Maria Petrou, Melvyn L.Smith , Lyndon N. Smith ,” Face Recognition and Verification Using Photometric Stereo: The Photoface Database and a Comprehensive Evaluation”, IEEE-2013, pp. 121-135.

[11]Shengcai Liao, Anil K.Jain, Stanz Z. Li,” Partial Face Recognition: Alignment Free Approach”, IEEE-2013, pp. 1193-1205

[12] 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

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] M. Tistarelli, and E. Grosso, “Active Vision-Based Face Authentication”, *Image and Vision Computing*, no. 18, pp. 299-314, 2000.

[18] 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.

[19] 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.

[20] 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.