Volterra kernel based face recognition using artificial bee colony optimization

Ankush Chakrabartyb,*, Harsh Jain a, Amitava Chatterjeea

a Department of Electrical Engineering, Jadavpur University, Kolkata-700032, West Bengal, India
b Department of Electrical and Computer Engineering, Purdue University, West Lafayette, IN-47906, USA

A B S T R A C T

The present paper describes a novel method of implementation of a stochastic optimization technique for the face recognition problem. The method proposed divides the original images into patches in space, and seeks a non-linear functional mapping using second-order Volterra kernels. The artificial bee colony optimization technique, a modern stochastic optimization algorithm, is used to derive optimal Volterra kernels during training to simultaneously maximize inter-class distances and minimize intra-class distances in the feature space. During testing, a voting procedure is used in conjunction with a nearest neighbor classifier to decide to which class each individual patch belongs. Finally, the aggregate classification results of all patches in an image are used to determine the overall recognition outcome for the given image. The utility of the proposed scheme is aptly demonstrated by implementing it on two popular benchmark face recognition datasets, and comparing the effectiveness of the proposed approach vis-à-vis other statistical learning procedures in facial recognition and also several other methods developed so far. The effectiveness of the artificial bee colony optimization technique and its Levy-mutated variation in optimizing Volterra kernels is conclusively proven in this paper by significantly outperforming many popular contemporary algorithms.

© 2012 Elsevier Ltd. All rights reserved.

1. Introduction

Prolific research has taken place in recent years on face recognition for its application in problems of biometrics, pattern recognition, and computer vision applications (Chellappa et al., 1995; Wechsler et al., 1996; Zhao et al., 2000; Gong et al., 2000). The reason behind the burgeoning of this field of research is that there is a large number of security and forensic applications involving facial recognition. Some of these applications include automated crowd surveillance, face reconstruction, design of human computer interfaces (HCI), multimedia communication (e.g., generation of synthetic faces), and content-based image database management (Lu, 2003).

Although there are a number of face recognition algorithms which work well in constrained environments, face recognition is still an open and very challenging problem in real applications. Many problems arise because of the variability of many parameters: face expression, pose, scale, lighting, and other environmental parameters (Oh, 2005). In this context, many innovative techniques have been formulated for face recognition throughout the scientific world. The face has been widely accepted as an effective biometric indicator owing to its advantage over other biometric indicators in the context of intrusiveness, accuracy, cost and ease of sensing (Lu, 2003).

The face recognition problem can be deconstructed into two sub-problems. The first issue to be answered is: (a) face verification, and the second is (b) face identification (Hjelmars, 2000).

The face verification problem involves checking whether the acquired facial image during a security procedure matches with the template facial image already preexisting in the database. Thus, the parameters which are used to gauge the performance in this type of problem are the verification rate (i.e., the rates at which legitimate users are granted access) as well as the false accept rates (the rate at which impostors are granted access).

The face identification problem is a one-to-many image matching evaluator. In this problem, the technique is used to identify to whom the facial image belongs from among a database of varying facial images. In this case, a large number of factors affect correct identification, namely the angle from which the sensor image is obtained, the lighting condition during image acquisition, facial expression, facial hair and others. In this problem, existing approaches provide scores to different images based on the qualitative similarity of the image acquired to the template image, and subsequently the scores are sorted. The template image corresponding to the highest score is then selected as the match for the acquired image.
In recent times, two major approaches have been proposed for face recognition problems. They are appearance-based and model-based approaches. Fig. 1 shows an overview of the facial recognition techniques that are currently available in a pictorial form. Among face recognition algorithms, appearance-based approaches have been successfully developed and tested as a reference. These approaches utilize pixel intensity or intensity-derived features. However, these methods may not perform well if the test images are significantly different from the training face data, due to variations in pose, illumination or expression. In contrast, model-based approaches are more robust and can incorporate variations in pose, illumination or expression. However, these methods may not perform well if the test images are significantly different from the training face data, due to variations in pose, illumination or expression.

![Fig. 1. Overview of facial recognition techniques.](image.png)

Section 2 discusses the theory of Volterra kernels and their application in biometric recognition as a face recognition tool popularly known as Volterraces (Kumar et al., 2009). Section 3 discusses the artificial bee colony (ABC) optimization procedure and details the algorithm used to compute the optimal Volterra kernel. Section 4 presents the evaluation of the performance of the proposed approach in comparison to previous approaches as well as the comparison of ABC as an effective stochastic optimization technique when compared to existing stochastic search techniques such as genetic algorithms and bacterial foraging optimization. Finally, the conclusions inferred from the results are presented in Section 5.

2. Volterra kernels

Recent advances in signal and image processing have often involved the use of non-linear system modeling and identification tools. One such widely utilized modeling tool is the Volterra series, which is discussed in detail in (Schetzen, 1980). This section briefly describes the salient features of the second-order truncated Volterra series and how it is employed in our present facial recognition problem.

2.1. Second-order Volterra (SOV) filters

The ideal Volterra series model, used in the modeling and analysis of nonlinear systems, is represented in the following infinite series form,

$$y_n(k) = \sum_{r_1 = -\infty}^{\infty} \cdots \sum_{r_N = -\infty}^{\infty} h_n(r_1, r_2, \ldots, r_N) x(k-r_1) x(k-r_2) \cdots x(k-r_N)$$

(1)

where $h_n(\cdot)$ is the nth order Volterra kernel. Although such an ideal representation would result in a highly accurate nonlinear model, it is not physically realizable to employ an infinite number of kernels and, in most practical cases, a choice of an order higher than two results in objectionable computational complexity (Kumar et al., 2009). Instead, truncation up to second-order terms are often found in the literature (Kumar et al., 2009; Singh and Chatterjee, 2011), which provides sufficient accuracy without an exponential increase in required computational effort. This second-order approximation is referred to as the second-order Volterra (SOV) series or filter. The mathematical expression for the truncated SOV filters (Zhang and Zhao, 2010) is given by:

$$y(n) = h_0 + \sum_{k=1}^{N} h_k(n)x(n-k+1) + \sum_{k_1=0}^{N-1} h_{k_1,k_2}(n)x(n-k_1)x(n-k_2)$$

(2)

where $h_0(n)$, $h_k(n)$ and $h_{k_1,k_2}(n)$ are constant, first and second-order kernels, respectively. Here, $N$ denotes the system memory size. In matrix form, (2) can be written as follows:

$$y(n) = H^T \chi(n)$$

(3)

Here, $H(n)$ comprises the SOV filter coefficients corresponding to the zero-order, first-order and second-order kernels. $\chi(n)$ represents the expansion of the input signal in terms of constant,
single terms and cross-product terms. The expression given in (3) shows how a nonlinear SOV filter can be finally converted to an expanded linear form where the output is given as a weighted linear combination of the entries in the expanded \( X(n) \) vector. The entries in \( X(n) \) vector are created by suitable combinations of \( x(n) \) at different sampling instants i.e., \( x(n), x(n-1), x(n-2) \) etc. Each of the \( H(n) \) and \( X(n) \) vectors is of length

\[
L = 1 + N + \frac{N(N+1)}{2}
\]

(4)

Here,

\[
H(n) = [h_0(n), h_1(n), \ldots, h_{n-1}(n), h_{n+1}(n), \ldots, h_N(n)]^T
\]

(5)

\[
X(n) = [x(n), x(n-1), \ldots, x(n-N+1), x^2(n), \ldots, x^2(n-N+1)]^T
\]

(6)

The formulation presented in (2)–(6) shows how SOV kernels can be utilized as systems/filters for a one-dimensional input signal time series \( x(n) \). This methodology can be similarly employed for expansion of this concept of employing SOV kernels for two-dimensional signals i.e., for image processing problems, as is carried out in this face recognition problem.

2.2. 2-D quadratic Volterra filters

In the literature, it has already been established that the one-dimensional (1-D) discrete Volterra series can be extended to the \( M \)-dimensional (\( M-D \)) case (Singh and Chatterjee, 2011; Ramponi et al., 1988). In this application, the 2-D quadratic Volterra filter is employed for the facial recognition problem in \( \mathbb{R}^2 \)-space. This quadratic filter, with finite support can be represented by the equation,

\[
y(n_1, n_2) = h_0 + h_1[x(n_1, n_2)] + h_2[x(n_1, n_2)]
\]

where, \( h_0 \) is a constant term, and the linear and quadratic vectors consisting of the coefficients of the kernels of this nonlinear filter \( h_1 \) and \( h_2 \), respectively are,

\[
h_1[x(n_1, n_2)] = \sum_{i=0}^{N_1-1} \sum_{j=0}^{N_2-1} h_{1(i,j)}x(n_1-i, n_2-j)
\]

\[
h_2[x(n_1, n_2)] = \sum_{i=0}^{N_1-1} \sum_{j=0}^{N_2-1} \sum_{k=0}^{N_1-1} \sum_{l=0}^{N_2-1} h_{2(i,j,k,l)}x(n_1-i, n_2-j)x(n_1-k, n_2-l)
\]

(8)

The finite support of this filter is the 2-D space defined by the regions intersected by

\[
0 \leq i, \quad k \leq N_1-1
\]

\[
0 \leq j, \quad l \leq N_2-1
\]

In matrix form, the filter operations may be represented by Ramponi et al. (1988),

\[
h_1[x(n_1, n_2)] = tr[H_1X_1]
\]

\[
h_2[x(n_1, n_2)] = tr[H_2X_2]
\]

(10)

where \( H_1 \) is the matrix of the filter coefficients of \( h_1(i,j) \) and \( X_1 \) is the \( N_1 \times N_2 \) matrix of pixels whose \( (i,j) \)th element corresponds to the \( (n_1-i, n_2-j) \)th input image pixel. \( X_2 \) is an \( N_1^2 \times N_2^2 \) matrix derived by calculating the Kronecker product of the original window of pixels i.e., \( X_1 \otimes X_1 \).

2.3. Posing the kernel optimization problem

The next objective is to calculate the values of the \( n \) Volterra kernels, which would involve the evaluation of multiple integrals. Instead, the authors of Kumar et al. (2009) have suggested a goodness functional which objectively evaluates the effectiveness of a particular kernel on the basis of its classification ability in feature space.

The primary objective of the present problem under consideration is to classify a set of images \( i = \{g_i\} \) into a set of classes \( C = C_k \) where \( k = \{1, 2, \ldots, N_c\} \) such that the classification occurrence is as high as possible. To achieve this optimal performance, a goodness functional is defined that can simultaneously achieve minimization of intra-class distance and maximization of inter-class distance (Kumar et al., 2009).

From the concept of Fisher’s linear discriminant, we know that the intra-class distances can be computed by deriving the scatter matrix, which is a measure of the variance of the data (Kumar et al., 2009; Fisher, 1936). If \( S_W \) is this scatter matrix, then,

\[
S_i = \sum_{x \in C_i} (x-\mu_i)(x-\mu_i)^T
\]

(11)

\[
S_W = \sum_{i=1}^{N_c} S_i
\]

(12)

where, \( N_c \) is the number of classes and \( \omega \) denotes data corresponding to each class.

Similarly, the inter-class distances can be computed in a matrix using the differences in the means of the \( i \)th class and \( j \)th class for all classes in \( N \)-dimensional feature space, or,

\[
S_B = (\mu_i-\mu_j)(\mu_i-\mu_j)^T
\]

(13)

The mathematical representation of the goodness function can be written as:

\[
J(K) = \frac{K^T S_W K}{K^T S_B K}
\]

(14)

In our problem \( K \) is the transformed quadratic Volterra kernel vector, formed by integrating the first order kernel coefficients and the second order kernel coefficients of the Volterra kernel. \( S_W \) is the matrix of intra-class distances, or within-class distances, and \( S_B \) is the matrix of inter-class or between-class distances. The convolution operation in image processing is utilized here to transform the image \( i \) into a new representation \( A_i \) such that the two dimensional convolution operations can be represented as Kumar et al. (2009):

\[
I_1 \otimes K = A_i K
\]

(15)

where \( K \) is the transformed Volterra kernel vector corresponding to the original second order Volterra kernel approximation \( K \). Similarly, \( S_W \) and \( S_B \) can be formed as Kumar et al. (2009):

\[
S_B = \sum_{C_k < C_l < C_k} (A_i - A_j)(A_i - A_j)^T
\]

\[
S_W = \sum_{C_k < C_l < C_k} (A_p - A_q)(A_p - A_q)^T
\]

(16)

\[
S_W \text{ and } S_B \text{ are symmetric matrices with sizes } (I^2 + I^2) \times (I^2 + I^2)
\]

where \( I \) is the size of the original 2-D kernel mask. \( K \) is of the size \((I^2 + I^2) \times 1 \). Thus, the face recognition problem using Volterra kernels is essentially an optimization problem.

3. Artificial bee colony (ABC) optimization

3.1. Conventional ABC algorithm

In this section we describe the artificial bee colony (ABC) algorithm and its application in the design of optimal Volterra kernels. Population based optimization techniques have attracted
attention of researchers in the recent past to achieve global optimum solution for various problems. Swarm intelligence, which can be considered as one of the emerging branches in evolutionary computing, provides researchers an effective tool for solving optimization problems. Swarm intelligence can be broadly defined as an attempt to design an algorithm based on the collective behavior of social insect colonies or other animal societies. The classical example in evolutionary computing used for solving optimization problem is the genetic algorithm (GA) (Goldberg, 1989). Later, many swarm intelligence algorithms are proposed for solving optimization problems such as the particle swarm optimization (PSO) (Kennedy and Eberhart, 1995), the ant colony optimization (ACO) etc. In 2005 D. Karaboga introduced a bee swarm algorithm called artificial bee colony algorithm (ABC) (Karaboga and Basturk, 2007) for numerical optimization problems. ABC has been employed by several researchers to solve various problems in different research areas (Basturk and Karaboga, 2006; Karaboga and BasturkAkay, 2007; Karaboga, 2009).

The ABC algorithm has been developed inspired the behaviors of the real bees in finding the food source, called the nectar, and sharing the information of food sources with the bees in the nest (Karaboga and Basturk, 2007). In ABC, the artificial agents are defined and classified into three types, namely, the employed bee, the onlooker bee and the scout. Each of them plays different role in the process: the employed bees stay on a food source and provides the neighborhood of the source in its memory; the onlooker gets the information of food sources from the employed bees in the hive and select one of the food source to gather the nectar; and the scout is responsible for finding new food, new nectar, sources. In ABC algorithm, the position of a food source represents a possible solution for the optimization problem and the nectar amount corresponds to the quality or fitness of the associated solution. The number of employed bees or the onlooker bees is equal to the number of solutions in the population. The Basic ABC algorithm has three control parameters, namely the size of the colony (NS), number of cycles or iterations (MAXN) and the value of the limit for termination criteria (ξ). These parameters are suitably chosen for a given problem.

The probability of the food source \( p_i \) is calculated as

\[
p_i = \frac{J_i}{\sum_{i=1}^{N} J_i}
\]  

(17)

where \( J_i \) is the quality or fitness of the solution \( i \), proportional to the nectar amount, is present in the food source for this solution position.

The choice of a food source by an onlooker bee is guided by Eq. (17). A possible food position or solution \( x_i \) is initialized as:

\[
x_{ij} = \min_x + \text{rand}[0, 1] \times (\max_x - \min_x)
\]  

(18)

where \( j \in \{1, 2, \ldots, D\} \) and \( D \) is the number of optimization parameters. A new food position is evaluated from its old position i.e., the new position of solution \( i \) is obtained as:

\[
v_{ij} = x_{ij} + \phi_{ij} (x_{ij} - x_{kj})
\]  

(19)

Here, \( k \) is a randomly chosen solution \( (k \neq i) \). The complete bee colony optimization algorithm has been presented in Algorithm 1.

3.2. Levy-mutated ABC algorithm

A variety of additional probabilistic methods for selecting new food sources have been recently presented in the literature, aiming to balance exploitation and exploration required for swift convergence in global optimization algorithms. It is a common notion that the basic ABC algorithm tends to suffer from exhaustive exploration instead of desired exploitation, and thus the need has been felt to introduce variations of the basic ABC algorithm which will attempt to introduce required trade-off between the exploration and exploitation strategies. The authors in (Rajasekhar and Abraham, 2011) introduced a mutation operator into the ABC algorithm based on an alpha-stable distribution known as Levy-distribution, or the Inverse Gaussian distribution. The distribution can be written in closed form as,

\[
L_{\alpha}(z) = \frac{1}{\pi} \int_{0}^{\infty} e^{-q^\alpha} \cos(qz) dq
\]  

(20)

Due to its characteristic tail region, it has a particular property of generating offsprings much further away from the parent. The Levy-mutation is introduced in the ABC algorithm at the step where a new food source is generated from the old one. Instead of the generating equation given in (19), the equation used is of the form,

\[
x_{\text{new}} = x_{\text{best}} + L_{\alpha,0}(z)
\]  

(21)

where, \( L_{\alpha,0}(z) \) signifies a vector, of required dimension, of Levy distributed random numbers or a Levy distributed random sequence in the parameter space. The paper demonstrates how ABC algorithms can be effectively utilized to solve face detection problems using two-dimensional Volterra kernels, and, in particular, how a new variant of ABC algorithm, called Levy-mutated ABC (named in this work as L-ABC) can be utilized for the said purpose to achieve significantly superior performance.

Algorithm 1. Algorithm of artificial bee colony optimization

Initialize the population of honeybees in the search space, \( x_j \)
Evaluate the fitness of the population \( J \)
Initialize cycle number = 1
Start while loop
   Produce new solutions (food source positions) \( v_{ij} \) in the neighborhood of \( x_{ij} \) for the employed bees using (19) or (21) where \( k \) is a solution in the neighborhood of \( i \), \( \phi_{ij} \) is a random number (either uniformly distributed in the range \([-1, 1]\) or Levy distributed)
   Evaluate new solutions \( J_{\text{new}} \)
   Apply the greedy selection process between \( x_i \) and \( v_i \)
   Calculate the probability values \( p_i \) for the solutions \( x_i \) by means of their fitness values using (14) and (17)
   Calculate the fitness values of solutions as:
   \[
   \begin{cases}
   1/\xi, \text{ for } J_i \geq 0 \\
   1 + \text{abs}(J_i), \text{ for } J_i < 0
   \end{cases}
   
   
   \]
   Normalize \( p_i \)
   Produce the new solutions (new positions) \( v_i \) for the onlookers from the solutions \( x_i \) selected depending on \( p_i \), and evaluate them.
   Apply the greedy selection process for the onlookers between \( x_i \) and \( v_i \)
   Determine the abandoned solution (source), if exists, and replace it with a new randomly produced solution \( x_i \) for the scout using the Eq. (18)
   Memorize the best food source position (solution) achieved so far (\( x_{\text{best}} \)).
\text{continue} \text{ until error function } < \xi \text{ or number of iterations equals } \text{NMAX }

4. Performance evaluation of the proposed method

In this section, the proposed ABC optimization algorithm based system is implemented on the Yale A and Extended Yale B
The proposed face recognition algorithm

Algorithm 2. The proposed face recognition algorithm

Preprocess the image as required.
Split original facial image into patches of pre-decided size ($N_p$)
Select training set size.
Randomly select patches for training set and the remaining patches form the testing set.

for $k=1$ to 10
  Initialize the population of honeybees, which corresponds to elements in the Volterra kernel, in the search space
  \[ [-1,1], x_i = k_0 \]
  for $p=1:N_p$
    Evaluate the fitness of each bee in the population using the Eq. (14)
    Initialize cycle number $= 1$
    Start while loop
    Produce new solutions $v_i$ using (19) or (21)
    Evaluate new solutions $J_{new}$
    Apply the greedy selection process between $x_i$ and $v_i$
    Calculate the probability values $p_i$ for the solutions $x_i$ using (17)
    Calculate the fitness of each of the solutions as:
    \[
    \frac{1}{1+\text{abs}(J_i)} \quad \text{for} \quad J_i \geq 0
    \]
    \[ 1+\text{abs}(J_i) \quad \text{for} \quad J_i < 0 \]
    Normalize $p_i$
    Produce the new solutions $v_i$ for the onlookers from the solutions $x_i$, selected depending on $p_i$, and evaluate them.
    Apply the greedy selection process for the onlookers between $x_i$ and $v_i$
    Determine the abandoned solution (source), if it exists, and replace it with a new randomly produced solution $x_i$ for the scout using (18)
    Memorize the best food source position (solution) achieved so far ($x_{best}$). This is the optimal Volterra kernel (K) for this iteration.
    Transform the patches using the optimal Volterra kernel using the 2-D Quadratic Volterra Filter with Eq. (7)
    $y(n_1,n_2) = h_0 + h_1[x(n_1,n_2)] + h_2[x(n_1,n_2)]$
    Classify the transformed patches using nearest neighbour method with a majority vote paradigm over all patches and select the class with the highest vote.
    Store the classification error percentage
    continue until error function $< \xi$ or number of iterations equals $NMAX$
end for all patches
end for 10 iterations

4.2. Encoding the ABC algorithm

The ABC optimization was set up with the following parameters. The number of onlooker bees and employed bees was considered to be 20 each, thus a total of 40 bees was created. The food source position in parameter space was analogous to each of the elements in a kernel of size $K$, hence each food source was a vector of size $1 \times (K^2 + 3K^2)/2$, i.e., the number of the coefficients to be independently determined for the transformed kernels of 2D-Volterra filter. The number of times a bee can explore within a neighborhood of its starting point before it becomes a scouting bee is set to 10. The objective function is evaluated by first converting each row vector into a suitable Volterra-Kernel and then computing the objective function value using the Eq. (14). The problem is configured as a minimization problem i.e., the objective is to minimize (14).

Fig. 2. Diagrammatic representation of the proposed ABC-Volterra faced recognition scheme.

Table 1. Unlike many previous approaches developed, no further pre-processing was required for our proposed method to successfully classify the patches. Each image is further preprocessed by performing un-sharp masking. The un-sharp masked versions of each image in each dataset (165 images in the Yale A dataset and 2432 images in the Extended Yale B dataset) are then, respectively, stored as revised datasets in order to evaluate the performance of the proposed technique. The dynamic range of the intensity values are also exponentially expanded to cover the entire gray-level intensity range of the 8-bit resolution images (i.e., from 0 to $2^8 - 1$). The preprocessing techniques result in sharpened high contrast images. Algorithm 2 and Fig. 2 shows how ABC algorithm has been utilized in conjunction with Volterra kernels to develop the proposed face recognition system.

### Table 1: Dataset information.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Image size</th>
<th>Patch size</th>
<th>Number of faces</th>
<th>Images per face</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yale A</td>
<td>64 x 64</td>
<td>8 x 8</td>
<td>11</td>
<td>15</td>
</tr>
<tr>
<td>Yale B</td>
<td>32 x 32</td>
<td>8 x 8</td>
<td>38</td>
<td>64</td>
</tr>
</tbody>
</table>

Fig. 2. Diagrammatic representation of the proposed ABC-Volterra faced recognition scheme.
4.3. Results

The optimization search space for each element of the kernel was chosen as $-1$ to $+1$ and the numbers of iterations were considered to be 2000. This value was chosen after several trials as the optimal iteration number which was required to produce satisfactory classification performance without adversely affecting the computational speed. Results are reported on the basis of 10 test runs using randomly selected training sets and vote-based testing routines. Average and standard deviations of classification error percentages are accordingly computed and tabulated using the proposed ABC algorithm for optimization. These results are compared to other similar stochastic search techniques employed such as genetic algorithms (GA) (Goldberg, 1989) and bacterial foraging optimization (BFO) (Tang et al., 2006; Acharya et al., 2007), used to optimize Volterra kernels to evaluate the classification error as the performance evaluation parameter, and it was observed that the ABC algorithm-based face recognition system outperforms the others as is tabulated in Tables 2 and 3. These tables also show performance comparisons with other face detection algorithms proposed in Kumar et al. (2009). From these results, one can conclude that our proposed ABC based Volterrafaces algorithm is superior to most of the other occasions, while conventional ABC proved to be a winner for the other occasions.

Tables 2 and 3 demonstrate these results in a mean/standard deviation format for 10 test runs. While a lower mean value indicates highly accurate face recognition systems with low errors and lower standard deviation justifies the inherent robustness of the system developed. In most cases, our proposed system not only produced the lowest error percentage, but also showed high robustness because of lower standard deviation values with random training sets. One interesting observation can be made from these results is that the L-ABC tends to improve the more difficult scenarios, when the number of training sets is miniscule, whereas conventional ABC tend to slightly outperform L-ABC for relatively larger training sets. From Tables 2 and 3, it can be easily observed that, in general, the errors of the ABC algorithm and the L-ABC algorithm are less than those reported for several competing algorithms, as shown. These results conclusively demonstrate the utility of using ABC based algorithms for solving these face recognition problems.

The improvement of the performance with the increment of training sets is also demonstrated in the pictorial representation given in Figs. 3 and 4 which is representative of the fact that the rate of decrease of error with the increase of the number of training sets is much higher for the proposed ABC-Volterra approach than the other approaches compared in this paper.

Table 2

<table>
<thead>
<tr>
<th>Classification error</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>S-LDA (Cai et al., 2007)</td>
<td>42.4/NA</td>
<td>27.7/NA</td>
<td>22.2/NA</td>
<td>18.3/NA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UVF (Shan and Cottrell, 2008)</td>
<td>27.1/NA</td>
<td>17.4/NA</td>
<td>11.7/NA</td>
<td>8.2/NA</td>
<td>6.3/NA</td>
<td>5.1/NA</td>
</tr>
<tr>
<td>TANMM (Wang and Zhang, 2007)</td>
<td>44.7/NA</td>
<td>29.6/NA</td>
<td>18.4/NA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OLAP (Cai et al., 2006)</td>
<td>44.3/NA</td>
<td>29.9/NA</td>
<td>22.7/NA</td>
<td>17.9/NA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Eigenfaces (Cai et al., 2006)</td>
<td>56.5/NA</td>
<td>51.1/NA</td>
<td>47.8/NA</td>
<td>45.2/NA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fisherfaces (Cai et al., 2006)</td>
<td>54.3/NA</td>
<td>35.5/NA</td>
<td>27.3/NA</td>
<td>22.5/NA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Laplacianfaces (Cai et al., 2006)</td>
<td>43.5/NA</td>
<td>31.5/NA</td>
<td>25.4/NA</td>
<td>21.7/NA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Paper (Kumar et al., 2009)</td>
<td>22.2/NA</td>
<td>13.4/NA</td>
<td>15.8/NA</td>
<td>10.2/NA</td>
<td>10.0/NA</td>
<td>9.7/NA</td>
</tr>
<tr>
<td>Program (Volterrafaces)</td>
<td>21.6/7.8</td>
<td>17.3/6.2</td>
<td>15.7/7.9</td>
<td>14.4/8.5</td>
<td>14.6/11.0</td>
<td>13.3/8.1</td>
</tr>
<tr>
<td>GA-Volterra</td>
<td>27.4/10.6</td>
<td>16.7/4.4</td>
<td>11.8/4.9</td>
<td>10.9/7.3</td>
<td>7.4/1.8</td>
<td>6.3/2.4</td>
</tr>
<tr>
<td>BFO-Volterra</td>
<td>22.5/10.7</td>
<td>18.7/8.1</td>
<td>12.3/4.3</td>
<td>9.4/6.1</td>
<td>7.6/2.5</td>
<td>5.9/3.0</td>
</tr>
<tr>
<td>ABC-Volterra</td>
<td>18.2/5.8</td>
<td>13.2/4.9</td>
<td>11.7/6.8</td>
<td>9.4/5.2</td>
<td>7.3/7.4</td>
<td>4.0/3.4</td>
</tr>
<tr>
<td>L-ABC-Volterra</td>
<td>17.3/10.6</td>
<td>13.8/5.3</td>
<td>10.5/5.9</td>
<td>9.4/5.2</td>
<td>7.8/6.7</td>
<td>5.1/3.2</td>
</tr>
</tbody>
</table>

Note: All values have been given as mean/standard deviation of classification error percentages.

Table 3

<table>
<thead>
<tr>
<th>Classification error</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>MASSO (Pham and Venkatesh, 2008)</td>
<td>58.0/NA</td>
<td>54.0/NA</td>
<td>50.0/NA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SR (Cai et al., 2007)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12.0/NA</td>
<td>4.7/NA</td>
</tr>
<tr>
<td>RDA (Cai et al., 2007)</td>
<td></td>
<td>11.6/NA</td>
<td>4.2/NA</td>
<td>1.8/NA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ORO (Hua et al., 2007)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>9.0/NA</td>
<td></td>
</tr>
<tr>
<td>KLPSO (An et al., 2008)</td>
<td>24.7/NA</td>
<td>9.9/NA</td>
<td>3.2/NA</td>
<td>1.4/NA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RHR (An et al., 2007)</td>
<td>23.9/NA</td>
<td>11.0/NA</td>
<td>3.7/NA</td>
<td>1.4/NA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CTA (Fu and Huang, 2008)</td>
<td>17.0/NA</td>
<td>7.6/NA</td>
<td>5.0/NA</td>
<td>2.5/NA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Eigenfaces (Fu and Huang, 2008)</td>
<td>54.7/NA</td>
<td>36.1/NA</td>
<td>31.2/NA</td>
<td>27.7/NA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fisherfaces (Fu and Huang, 2008)</td>
<td>37.6/NA</td>
<td>18.9/NA</td>
<td>16.8/NA</td>
<td>14.9/NA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Laplacianfaces (Fu and Huang, 2008)</td>
<td>34.1/NA</td>
<td>18.0/NA</td>
<td>30.3/NA</td>
<td>20.2/NA</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Volterrafaces (Kumar et al., 2009)</td>
<td>40.8/NA</td>
<td>20.5/NA</td>
<td>14.4/NA</td>
<td>13.0/NA</td>
<td>4.0/NA</td>
<td>1.3/NA</td>
</tr>
<tr>
<td>Volterrafaces (Volterrafaces)</td>
<td>47.9/17.4</td>
<td>24.7/8.2</td>
<td>21.3/8.2</td>
<td>14.9/5.7</td>
<td>5.5/3.3</td>
<td>1.6/6.7</td>
</tr>
<tr>
<td>GA-Volterra</td>
<td>42.4/11.9</td>
<td>34.4/10.5</td>
<td>23.9/7.6</td>
<td>17.8/3.5</td>
<td>11.2/4.7</td>
<td>3.1/1.4</td>
</tr>
<tr>
<td>BFO-Volterra</td>
<td>30.2/6.3</td>
<td>19.8/6.0</td>
<td>16.8/5.0</td>
<td>11.3/3.1</td>
<td>3.8/2.0</td>
<td>1.5/0.8</td>
</tr>
<tr>
<td>ABC-Volterra</td>
<td>28.6/10.7</td>
<td>21.0/5.6</td>
<td>13.3/3.9</td>
<td>10.9/4.2</td>
<td>3.6/1.2</td>
<td>1.2/0.7</td>
</tr>
<tr>
<td>L-ABC-Volterra</td>
<td>28.6/5.0</td>
<td>18.2/3.6</td>
<td>13.3/3.7</td>
<td>10.8/4.4</td>
<td>3.3/1.0</td>
<td>1.4/0.4</td>
</tr>
</tbody>
</table>

Note: All values have been given as mean/standard deviation of classification error percentages.
The results reported in Tables 2 and 3 and Figs. 3 and 4 are computed with \( 3 \times 3 \) Volterra kernels employed. Next we make a detailed analysis of the performance of the proposed ABC-Volterra system and the computation time taken with variation in the kernel size. The classification performances for different training sets in different kernel sizes are represented graphically in Figs. 5 and 6 for the two benchmark image datasets. The corresponding computation times involved in these system implementations are presented in Figs. 7 and 8, respectively. It may be noted that computational times have been mentioned for the conventional ABC only, because the L-ABC works in identical computational time. It is apparent from these results that the general trend of the classification error is higher for increased kernel size, which is an added advantage since the average classification computation times of the \( 5 \times 5 \) kernel is significantly lower (almost 1/10th) than the average classification computation times of the \( 3 \times 3 \) kernel for both datasets. Facial recognition requires in online situations, quicker processing speeds, and so the small kernel size is conducive to facial recognition applications for its accuracy as well as its processing speed.
An, S., Liu, W., Venkatesh, S.. Exploiting Side Information in Locality Preserving
Acharya, D.P., Panda, G., Mishra, S., Lakshmi, Y.V.S., Bacterial foraging based
Extended Yale B datasets, and the proposed method has been
optimal kernel designed is then used for classification in two
non-linear mapping functional called the Volterra kernel. The
tional, where the decision variables correspond to elements of a
optimization algorithm called the Artificial Bee Colony optimization
outperformed the other commonly used stochastic optimization
tecnhiques such as the genetic algorithm, and the bacterial foraging
optimization, used for similar optimization of Volterra kernels. An
Further scope exists in attempting to select an optimal itera-
tion number as well as termination criteria in order to apply this
work to future on-line face recognition systems for quicker and
more accurate classification.

5. Conclusion

The present paper proposes an improved Volterrafaces method-
dology for face recognition by employing a modern stochastic
optimization algorithm called the Artificial Bee Colony optimization
technique for the minimization of a pre-defined goodness func-
tional, where the decision variables correspond to elements of a non-linear mapping functional called the Volterra kernel. The
optimal kernel designed is then used for classification in two standard face recognition problem datasets known as Yale A and
Extended Yale B datasets, and the proposed method has been proven to outperform many existing technologies reported so far
(Kumar et al., 2009). Also, the bee colony optimization technique has
outperformed the other commonly used stochastic optimization
techniques such as the genetic algorithm, and the bacterial foraging
optimization, used for similar optimization of Volterra kernels. An
additional Levy-mutated modification of the ABC has been imple-
mented and comparison has been made to the conventional ABC.

Acknowledgements

The authors would like to thank Professor Stanislaw Zak for his
tremendously helpful insights on the improvement of this paper.
Professor Zak is associated with the Electrical and Computer
Engineering department at Purdue University. His interests are in
control, intelligent systems, and optimization.

References

Acharya, D.P., Panda, G., Mishra, S., Lakshmi, Y.V.S., Bacterial foraging based
independent component analysis. In: Proc. Int. Conf. Computational Intelli-
An, S., Liu, W., Venkatesh S. Face Recognition Using Kernel Ridge Regression. In
An, S., Liu, W., Venkatesh, S. Exploiting Side Information in Locality Preserving

Basturk, B., Karaboga, D, An artificial bee colony (ABC) algorithm for numeric
function optimization. IEEE Swarm Intelligence Symposium 2006, Indianapolis,
IN, USA, (2006).
Cai, D., He, X., Han, J., Zhang, HJ., 2006. Orthogonal Laplacianfaces for facial
Cai, D., He, X., Hu, Y., Han, J., Huang, T., Learning a Spatially Smooth Subspace for
Cai, D., He, X., Han, J, Spectral Regression for Efficient Regularized Subspace
Chellappa, R., Wilson, CL.,hirosey, S., 1995. Human and machine recognition of
Data Mining and Knowledge Discovery Handbook, 469–504.
218–237.
14 (11), 1151–1164.
 Eugen., 7, 179–188.
TIP 17 (2).
Recognition. Imperial College Press and World Scientific Publishing.
Hjelmas, E., Biometric Systems: A Face Recognition Approach (www.researchin
dex.org), (2000).
Hua, G., Viola, P., Drucker S. Face Recognition Using Discriminatively Trained
Karaboga, D., Basturk, B., 2007. A powerful and efficient algorithm for numerical
function optimization: artificial bee colony (ABC) algorithm. J. Global Optim.
Karaboga, D., Basturk Akay, B., An artificial bee colony (ABC) algorithm on training
artificial neural networks In: 15th IEEE Signal Processing and Communications
Kumar, R., Banerjee, A., Vemuri, B.C., Volterrafaces: discriminant analysis using
Recognition, (2009).
OB, R.J, Face recognition using radial basis function network based on LDA In:
Pham, D.S., Venkatesh, S. Robust Learning of Discriminative Projection for Multi-
Rajasekhar, A., Abraham, A., Pant, M., Levy, mutated artificial bee colony algorithm
for global optimization. Syst. Man Cybern. (SMC), 2011 IEEE International
Conference on 9–12 Oct, pp.655–662. 2011 http://dx.doi.org/10.1109/ICSMC.
2011.6083786.
Ramponi, G., Sicuranza, G.L., Uckovich, W., 1988, A computational method for the
Wiley, New York.
Shan, H., Cottrell, G.W. Looking Around the Backyard Helps to Recognize Faces and
for nonlinear system identification based on second order Volterra and
Tang, W.J., Wu, Q.H., Saunders, J.R., Bacterial Foraging Algorithm for Dynamic Environ-
Wang, F., Zhang, C, Feature Extraction by Maximizing the Average Neighborhood
Wechsler, H., Phillips, P., Bruce, V., Soulie, F., Huang, T., 1996. Face Recognition:
From Theory to Applications. Springer-Verlag.
Yale, A. Dataset. (http://www.face-rec.org/databases/).
Extended Yale B. Dataset. (http://cvl.yale.edu/projects/yalefaces/yalefaces.html).
Zhao, W., Chellappa, R., Rosenfeld, A., Phillips, P.J., Face Recognition: A literature

Fig. 8. Classification time variation for variations in kernel size (Ext. Yale B).