Engineering Applications of Artificial Intelligence 34 (2014) 13–22

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

A novel local extrema based gravitational search algorithm and its application in face recognition using one training image per class

Tapabrata Chakraborti [a], Kaushik Das Sharma [b], Amitava Chatterjee [a,*]

[a] Department of Electrical Engineering, Jadavpur University, Kolkata 700032, India
[b] Department of Applied Physics, University of Calcutta, Kolkata 700009, India

ARTICLE INFO

Article history:
Received 23 March 2013
Received in revised form
1 December 2013
Accepted 5 May 2014

Keywords:
Gravitational search algorithm (GSA)
Fisher linear discriminant analysis (FLDA)
Singular value decomposition (SVD)
2-D random local extrema gravitational search algorithm (2-D RLEGSA)

ABSTRACT

In this present paper a new methodology has been presented involving a stochastic optimization based approach to solve the face recognition problem with only one training image per class. Singular value decomposition (SVD) is used to decompose the single training image into two component images in order to compute the within class scatter matrix. The stochastic optimization approach is implemented employing gravitational search algorithm (GSA) which searches for an optimal transform matrix instead of using the traditional solution of general eigenvalue problem as is carried out in Fisher linear discriminant analysis (FLDA). The present paper also proposes two novel variants of GSA, namely the 2-D version of GSA, in order to cater for the 2-D image data, and the other one is a 2-D randomized local extrema based GSA (RLEGSA), which employs a stochastic local neighborhood based search instead of global search, as in basic GSA. Finally, a novel concept of performing an automated selection of projection vectors is incorporated in the 2-D RLEGSA to propose an improved variant, called the Modified RLEGSA (MRLEGSA). Experimental results, based on benchmark Yale A and ORL databases, show that the proposed methods outperform several existing schemes.

1. Introduction

In recent years face recognition has become a widely researched topic since it has numerous real world applications like authentication, identification, advanced human computer interaction and many other emerging fields of research. Face recognition spans the subjects of pattern recognition, image processing, computer vision, machine learning, etc. With the growing importance of biometric recognition systems (Jain and Prabhakar, 2004), due to low susceptibility to security loss, face recognition based biometrics has gained much popularity in recent times.

Many approaches to face recognition have been proposed over the last two decades (Zhao et al., 2003; Jafri and Arabnia, 2009; Chakrabarty et al., 2013) most of which are based on supervised learning. Hence they follow a common sequence of steps. There is a feature extraction step in which a set of discriminating features are extracted from a set of training images (Brunelli and Poggio, 1993). Then if the set of features extracted is dimensionally large, there may be a feature selection/reduction procedure where a reduced set of highly discriminating features are selected employing a suitable algorithm which may attempt to optimize a suitable cost function. For the face recognition problem, since images of different persons are, after all, human faces they have some common characteristics which indicates that some features in a large set of extracted features will not have enough discriminating power. This makes feature selection/reduction an important step as a large feature set might not necessarily result in a higher recognition rate (Tu et al., 2007). This step is followed by the classification step where the final conclusion regarding recognition or authentication is actually performed. Several variations of methods proposed in each of these steps generates new approaches in solving the problem.

A general drawback of the supervised learning method is that for a good classification accuracy rate, the number of training samples needs to be sufficiently large (depending on the number of test images and the number of classes). In those particular methods where inter class and intra class distances are used, the methods do not work at all when there is a single training image of each subject because in this case, the intra class distances are not defined (Gao et al., 2008). This drawback is prominent in several approaches which include the popular methodology of Fisher linear discriminant analysis (FLDA). A few methods have been proposed in recent years to solve this problem of FLDA based face recognition where there is only one training image per person e.g. generalized inverse method (Tian et al., 1988), perturbation based method (Hong and Yang 1991), direct FLDA method (Yu and Yang 2001), null space method (Lu et al., 2003), 2-D FLDA method

* Corresponding author. Tel.: +91 3324146949.
E-mail address: cha_ami@yahoo.co.in (A. Chatterjee).

http://dx.doi.org/10.1016/j.engappai.2014.05.002
0952-1976/

(Ye et al., 2004), singular value decomposition (SVD) based method (Gao et al., 2008) etc.

Our present research concentrates on those more challenging face recognition problems which suffer from small sample size (SSS) problem, typically in those situations where there is only a single training sample available per class/person. The research on face recognition problem using single training sample per person is well known as a very challenging problem and it has gained prominence in recent times (Gao et al., 2008; Tan et al., 2006; Zhu et al., 2012). This situation arises in many real-world scenarios such as utilization of smart cards, airport check-in and check-out, special situations of law enforcement, critical surveillance scenarios and checking for access control etc. (Tan et al., 2006; Zhu et al., 2012). Our present work is inspired by the method proposed by Gao et al. (2008) in which the single training image of a particular class is decomposed into two component images using SVD and then the intra class distance can be conveniently determined using these two resulting images. However, the SVD based FLDA approach in Gao et al. (2008) uses the general eigen value theory to solve the cost function. In this paper we propose a novel method of solving the FLDA cost function using an intelligent iterative stochastic optimization algorithm which can simultaneously solve the feature selection/reduction phase along with the feature extraction phase, thus effectively merging the operations required in two steps. The iterative stochastic optimization problem is solved using a recently proposed method, called gravitational search algorithm (GSA). GSA is a powerful iterative optimization algorithm based on Newton's laws of gravity and motion (Rashedi et al., 2009; Pal et al., 2013). Several interesting applications have recently been proposed using GSA in the domains of e.g. image processing (Sun and Zhang, 2013) and data clustering (Hatamlou et al., 2012; Hatamlou et al., 2011). Three modifications of the GSA have been proposed in this paper with the objective of solving our problem. The first variation proposes a 2-D GSA to adopt the GSA in processing of 2-D images. The second variation introduces a novel random local extrema based GSA (RLEGSA). To the best of our knowledge and belief, although some local best methods have been proposed earlier for a similar swarm intelligence based method called particle swarm optimization (PSO) (Suganthan, 1999; Das Sharma et al., 2012), this is the first such variation developed in the genre of GSA. The third variation incorporates the automated selection of projection vectors within the GSA based cost function optimization framework.

The rest of this paper is organized as follows. Section 2 provides a description of the SVD and FLDA based feature extraction schemes for the single training image per person scenario. Section 3 describes an overview of the traditional gravitational search algorithm and detailed descriptions of the novel variants proposed in this work. Section 4 presents the experiments and simulation results. Section 5 concludes the paper.

2. SVD and FLDA based feature extraction scheme

Let us consider there are C classes with each having a single image $I_k \in \Re^{m\times n}$ $(k=1,\ldots,C)$. If $m \geq n$, then let $U_k \in \Re^{m\times m}$ and $V_k \in \Re^{n\times n}$ be the eigenvector matrices of $I_k I_k^T$ and $I_k^T I_k$ respectively. Let u_i^k and v_i^k be the ith column of U_k and V_k respectively. Let σ_i^k be the ith singular value of I_k such that σ_i^k is in descending order of magnitude as i increases i.e. $\sigma_1^k \geq \sigma_2^k \geq \cdots \sigma_{i-1}^k \geq \sigma_i^k \geq \sigma_{i+1}^k \cdots \sigma_n^k$. Then the image can be described as (Gao et al., 2008; Golub and Loan, 1983)

$$I_k = \sum_{i=1}^{n} \sigma_i^k u_i^k (v_i^k)^T \tag{1}$$

Hence each image can be thought of being constituted as a summation of n basis images where each basis image corresponds to a particular singular value and the energy content of a basis image is higher if its associated singular value is higher in magnitude. Following the philosophy described in Gao et al. (2008), an image $\hat{I}_k$ is constructed taking the three most significant SVD basis images as

$$\hat{I}_k = \sum_{i=1}^{3} \sigma_i^k u_i^k (v_i^k)^T \tag{2}$$

Thus after obtaining $\hat{I}_k$ we have two image matrices I_k and $\hat{I}_k$ in each class k, and we can also compute a difference image, $\Delta I_k = I_k - \hat{I}_k = \sum_{i=4}^{n} \sigma_i^k u_i^k (v_i^k)^T$. The creation of $\hat{I}_k$ and ΔI_k facilitates the calculation of an approximate within-class scatter matrix, which is, otherwise, not possible when we have only one training image per person/class. Let the within class scatter matrix be denoted as S_w and the between class scatter matrix be denoted as S_b. To compute S_w and S_b we need to compute the global mean image $\bar{I}$ and mean image of the kth class $\bar{I}_k$, which are defined as (Gao et al., 2008)

$$\bar{I}_k = \tfrac{1}{2}(I_k + \hat{I}_k) \tag{3}$$

$$\bar{I} = \frac{1}{2C} \sum_{k=1}^{C} (I_k + \bar{I}_k) \tag{4}$$

Then S_w and S_b can be computed as

$$S_b = \frac{1}{C} \sum_{k=1}^{C} (\bar{I}_k - \bar{I})^T (\bar{I}_k - \bar{I}) \tag{5}$$

$$S_w = \frac{1}{C} \sum_{k=1}^{C} [(I_k - \bar{I}_k)^T (I_k - \bar{I}_k) + (\hat{I}_k - \hat{I}_k)^T (\hat{I}_k - \bar{I}_k)] \tag{6}$$

Using (3) and (4), one can obtain (Gao et al., 2008)

$$S_w = \frac{1}{2C} \sum_{k=1}^{C} (I_k - \hat{I}_k)^T (I_k - \hat{I}_k) \tag{7}$$

From the theory of two-dimensional FLDA (Ye et al., 2004), then our goal will be to seek a set of d optimal discriminating column vectors w_j $(j=1, 2, \ldots, d)$ constituting $m \times d$ optimal projection matrix W so as to minimize the cost function

$$J(W) = \frac{trace(W^T S_w W)}{trace(W^T S_b W)} \tag{8}$$

Once W is determined, all the training images are projected on to W to obtain the feature matrices Z_k

$$Z_k = I_k \times W, \quad k = 1, 2, \ldots, C \tag{9}$$

Then, for each class k we have a feature matrix Z_k. If we have an input test image I then the corresponding feature matrix is $Z = I \times W$

Then we can utilize the nearest neighbor classifier method, where we calculate the Euclidean distance D_k of feature matrix Z from each feature matrix Z_k pertaining to class k, given as

$$D_k = \|Z_k - Z\|, \quad k = 1, \ldots, c \tag{10}$$

Then, I is identified to belong to that class k for which D_k is minimum.

Now in Gao et al. (2008) the optimal value of W has been determined by using the general eigen value theorem and for that the cost function considered is the inverse of (8), which is the traditional cost function considered in Ye et al. (2004), as they solved a maximization problem. In this work, as we attempt to solve a minimization function by utilizing the gravitational search algorithm and its proposed variants, we have utilized a form of $J(W)$ which is inverse of that considered in Ye et al. (2004) and Gao et al. (2008).

3. Gravitational search algorithm and the proposed variants

3.1. Gravitational search algorithm (GSA)

GSA is a relatively recently proposed optimization algorithm based on the Newtonian law of gravity (Rashedi et al., 2009). Let us consider an isolated system of p objects called agents or particles. This can be considered a universe consisting of only these p agents which obey (1) Newton's Law of Motion and (2) Newton's Law of Gravity. Let the position of the ith agent ($i=1, 2, \ldots, p$) in n dimensional space at time t be given by

$$X_i(t) = \{x_i^1(t), x_i^2(t), \ldots, x_i^n(t)\} \tag{11}$$

Also we define three types of masses of an agent (Rashedi et al., 2009):

(1) Active gravitational mass (M_a) – it is the cause of the gravitational field due to an object. Greater the active gravitational mass of an object greater will be the strength of the gravitational field due to that object.
(2) Passive gravitational mass (M_p) – greater the passive gravitational mass of an object greater will be the gravitational force that it will experience in the same gravitational field.
(3) Inertial mass (M_i) – inertial mass determines the resistance of an object to change its state of motion when a force is applied on it. When the same force is applied, an object with larger inertial mass will change its position more slowly.

At a certain instant of time t, the gravitational force experienced by an object i due to the object j in the dimension d is given by Newton's Law of Gravity as follows (Rashedi et al., 2009):

$$F_{ij}^d(t) = G(t)\frac{M_{pi}(t) \times M_{aj}(t)}{R_{ij}(t)+\varepsilon}(x_j^d(t) - x_i^d(t)) \tag{12}$$

where M_{aj} is the active gravitational mass of agent j, M_{pi} is the passive gravitational mass of agent i, ε is a small positive constant ($\varepsilon > 0$), and $R_{ij}(t)$ is the Euclidean distance between two agents i and j.

$$R_{ij}(t) = \|X_i(t), X_j(t)\| \tag{13}$$

Now the gravitational constant $G(t)$ at time t is given by (Rashedi et al., 2009; Mansouri et al., 1999)

$$G(t) = G(t_0) \times \left(\frac{t_0}{t}\right)^{\beta}, \quad \beta < 1 \tag{14}$$

where t_0 is the initial time/iteration. Hence $G(t)$ gradually decreases over time, representing the effect of ageing. The introduction of this concept facilitates stronger and stronger exploitations in later parts of iterations as it is desired that the changes should be lesser and lesser as the system converges toward a solution. The total gravitational force on a particle i in the dth dimension is (Rashedi et al., 2009)

$$F_i^d(t) = \sum_{j=1, j\neq i}^{p} rand_j F_{ij}^d(t) \tag{15}$$

where $rand_j$ is an uniformly distributed random number in the interval [0,1]. This introduces the stochastic element into the algorithm. Next, by Newton's Law of Motion, the acceleration of the agent i at time t in dimension d is calculated as

$$a_i^d(t) = \frac{F_i^d(t)}{M_{ii}(t)} \tag{16}$$

where M_{ii} is the inertial mass of the ith agent.

Now the velocity, position and mass update equations are as follows:

$$v_i^d(t+1) = rand_i \times v_i^d(t) + a_i^d(t) \tag{17}$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \tag{18}$$

$$M_{ai} = M_{pi} = M_{ii} = M_i \tag{19}$$

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \tag{20}$$

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^{p} m_j(t)} \tag{21}$$

where $fit_i(t)$ is the fitness value of the agent i at time t.

So the $best(t)$ and the $worst(t)$, the best and worst fitness among the p particles at time t, respectively, are given as

$$best(t) = \min_{j \in \{1,2,\ldots,p\}} fit_j(t) \tag{22}$$

$$worst(t) = \max_{j \in \{1,2,\ldots,p\}} fit_j(t) \tag{23}$$

After the stopping criterion is met the position of the agent with the largest mass gives the solution of the search. The larger the mass of an agent, more slowly will it change its position and this is what is desired as it is nearer to the solution. Also it can be noted here that unlike the classical PSO, the basic GSA is a memory less algorithm since unlike PSO there is no need to keep record of the best position of each particle and also the globally best position. Algorithm 1 describes the implementation of the traditional GSA.

Algorithm 1. Traditional GSA

BEGIN
Create p particles and make randomized initialization of their n dimensional positions X.
Initialize iteration number $t=1$
REPEAT:
FOR $i=1$ to p
Calculate fitness $fit_i(t)$
END FOR
Calculate global best fitness: $best(t) = \min_{i \in \{1,2,\ldots,p\}} fit_i(t)$
Calculate global worst fitness: $worst(t) = \max_{i \in \{1,2,\ldots,p\}} fit_i(t)$
FOR $i=1$ to p
Calculate mass: $m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)}$, $M_i(t) = \frac{m_i(t)}{\sum_{j=1}^{p} m_j(t)}$
Calculate gravitational constant:
$G(t) = G(t_0) \times (\frac{t_0}{t})^{\beta}, \quad \beta < 1, \ t_0 = 1$
FOR $j=1$ to p, $j \neq i$
Calculate distance between two particles:
$R_{ij}(t) = X_i(t), X_j(t)$
Calculate force between two particles:
FOR $d=1$ to n
$F_{ij}^d(t) = G(t)\frac{M_i(t) \times M_j(t)}{R_{ij}(t)+\varepsilon}(x_j^d(t) - x_i^d(t))$
END FOR
END FOR
FOR $d=1$ to n
Calculate total force on ith particle:
$F_i^d(t) = \sum_{j=1, j\neq i}^{p} rand_j F_{ij}^d(t)$
Calculate acceleration: $a_i^d(t) = \frac{F_i^d(t)}{M_i(t)}$
Calculate velocity: $v_i^d(t+1) = rand_i \times v_i^d(t) + a_i^d(t)$
Calculate position: $x_i^d(t+1) = x_i^d(t) + v_i^d(t+1)$
END FOR
END FOR
UNTIL termination criterion is satisfied
END

3.2. Two dimensional gravitational search algorithm (2-D GSA)

The first modification of GSA proposed in this work is developed in order to account for a two dimensional solution space. In this case the position variable for the ith particle ($i=1, 2, \ldots, p$) becomes an $n \times n$ matrix (corresponding to the size of the W matrix) as follows:

$$X_i(t) = \{x_i^{kl}(t)\} \quad i = 1, \ldots, p;\ k = 1, \ldots, n;\ l = 1, \ldots, n \tag{24}$$

Thus the modified force calculation equation between two particles is developed as

$$F_{ij}^{kl}(t) = G(t)\frac{M_{pi}(t) \times M_{aj}(t)}{R_{ij}(t)+\varepsilon}(x_j^{kl}(t) - x_i^{kl}(t)) \tag{25}$$

where $R_{ij}(t)$ is the Euclidean distance between two agents i and j, given as

$$R_{ij}(t) = \|X_i(t), X_j(t)\| \tag{26}$$

The total gravitational force on a particle i in the dimension k, l is

$$F_i^{kl}(t) = \sum_{j=1, j \neq i}^{p} rand_j F_{ij}^{kl}(t) \tag{27}$$

Accordingly, the modified acceleration, velocity and position update equations are given as

$$a_i^{kl}(t) = \frac{F_i^{kl}(t)}{M_{ii}(t)} \tag{28}$$

$$v_i^{kl}(t+1) = rand_i \times v_i^{kl}(t) + a_i^{kl}(t) \tag{29}$$

$$x_i^{kl}(t+1) = x_i^{kl}(t) + v_i^{kl}(t+1) \tag{30}$$

The mass update equations remain the same as in the traditional GSA.

3.3. 2-D random local extrema gravitational search algorithm (2-D RLEGSA)

This is the second proposed variation of GSA in this paper, where instead of using the global best and global worst values in the mass update equation, the local best and local worst values are used. For each particle, these are the best and worst fitness values in the 'locality' or neighborhood of that particle. The process of determination of this neighborhood is the main contribution of this method. The method proposes incorporation of a random element in the determination of neighborhood to boost the rate of convergence.

Like before let there be p particles with n dimensional position vectors $X_i(t)$, $i=1, 2, \ldots, p$. Now a normalized Euclidian distance of ith particle from any other particle is calculated as

$$d_{ij} = \frac{\|X_i - X_j\|}{\max_j(\|X_i - X_j\|)} \text{where } j = 1 \text{ to } p \text{ and } j \neq i \tag{31}$$

A corresponding distance vector for particle i is created as

$$D_i = \{d_{ij}\}, \quad j = 1, 2, \ldots, p \tag{32}$$

Now, let the vector D_i be rearranged such that its individual elements appear in ascending order of magnitude and let this new vector be denoted as D_i^{asc}. At iteration number t we shall consider those particles in the neighborhood of particle i which have distances corresponding to the first $q_i(t)$ number of elements of D_i^{asc}, where $q_i(t)$ is given as

$$q_i(t) = round\left(\left[K \times \frac{t}{iter_{max}}\right] \times p \times rand[0, 1]\right) \tag{33}$$

If $q_i(t) > p$ then $q_i(t) = p$, and K the scaling constant.

Once the neighborhood of the ith particle is thus found out, the best and worst fitness values are calculated over the neighborhood only, hence the name of the method. As the number of iterations increases, the neighborhood of each particle grows in size and it finally approaches the traditional global extrema case, where the entire search space becomes the neighborhood of each particle. Algorithm 2 describes the implementation of the proposed Random Local Extrema based GSA (RLEGSA). The entire algorithm has been developed as a modification of the 2-D GSA described in Section 3.2 and hence it is called 2-D RLEGSA.

Algorithm 2. RLEGSA.

BEGIN
Create p particles and make randomized initialization
of their positions X. $X_i(t) = \{x_i^{kl}(t)\}$ $i = 1, \ldots, p;\ k = 1, \ldots, n;\ l = 1, \ldots, n$
Initialize iteration number $t=1$
REPEAT:
FOR $i=1$ to p
FOR $j=1$ to p, $j \neq i$
Calculate normalized distance between two particles:
$d_{ij} = \frac{\|X_i - X_j\|}{\max_j(\|X_i - X_j\|)}$
END FOR
Form distance vector: $D_i = \{d_{ij}\}, j = 1, \ldots, p$ $D_i^{asc} = sort_{ascend}(D_i)$
Calculate neighborhood:
N_i^{asc} = particles corresponding to first q_i^t elements of D_i^{asc}
$q_i^t = \left[K \times \frac{t}{iter_{max}}\right] \times p \times rand[0, 1]$
if $q_i^t > p$ then $q_i^t = p$
Calculate fitness $fit_i(t)$
Calculate local best fitness: $lbest_i(t) = \min_{l \in \{N_i^{asc}\}} fit_l(t)$
Calculate global best fitness: $lworst_i(t) = \max_{l \in \{N_i^{asc}\}} fit_l(t)$
Calculate mass: $m_i(t) = \frac{fit_i(t) - lworst_i(t)}{lbest_i(t) - lworst_i(t)}$, $M_i(t) = \frac{m_i(t)}{\sum_{j=1}^{p} m_j(t)}$

Calculate gravitational constant: $G(t)=G(t_0)\times\left(\frac{t_0}{t}\right)^{\beta},\quad \beta<1,\ t_0=1$
FOR $j=1$ to p, $j\neq i$
Calculate distance between two particles: $R_{ij}(t)=X_i(t),X_j(t)$
Calculate force between two particles:
FOR $k=1$ to n
FOR $l=1$ to n
$F_{ij}^{kl}(t)=G(t)\frac{M_i(t)\times M_j(t)}{R_{ij}(t)+\varepsilon}(x_j^{kl}(t)-x_i^{kl}(t))$
END FOR
END FOR
END FOR
FOR $k=1$ to n
FOR $l=1$ to n
Calculate total force on ith particle: $F_i^{kl}(t)=\sum_{j=1,j\neq i}^{p} rand_j F_{ij}^{kl}(t)$
Calculate acceleration: $a_i^{kl}(t)=\frac{F_i^{kl}(t)}{M_i(t)}$
Calculate velocity: $v_i^{kl}(t+1)=rand_i\times v_i^{kl}(t)+a_i^{kl}(t)$
Calculate position: $x_i^{kl}(t+1)=x_i^{kl}(t)+v_i^{kl}(t+1)$
END FOR
END FOR
END FOR
UNTIL termination criterion is satisfied
END

3.4. Modified random local extrema gravitational search algorithm (MRLEGSA) incorporating automated selection of number of projection vectors

In Gao et al. (2008), the scheme actually utilized different subsets of projection vectors from the optimal transform matrix solution W obtained by solving the generalized eigenvalue problem. The selection of number of f such projection vectors from the n vectors in W matrix ($f\leq n$) and which such f vectors should be chosen were both carried out manually. Their extensive experimentations showed that, for each separate face database, the performance varies individually with variation in number and choices of f vectors and no correlation can be obtained in determining their optimal choices which means they had to make an extensive manual search for the best combination of f vectors for each face database. This inspired us to propose an improved algorithm which can perform an additional task of an automated selection of the number of optimal projection vectors to be chosen from the W matrix and which candidate f vectors should be chosen that will lead to better performance accuracy. This has been referred to here as the third proposed modification of GSA. It actually augments the RLEGSA proposed in the previous subsection by incorporating the automated selection of number of projection vectors and also which projection vectors to be chosen from the W matrix.

Here for the ith agent, the solution space contains, besides the $n\times n$ matrix $X_i(t)$, an $n\times 1$ vector $S_i(t)$, given as

$$X_i(t)=\{x_i^{kl}(t)\}_{n\times n}\quad i=1,\ldots,p;\ k=1,\ldots,n;\ l=1,\ldots,n \tag{34}$$

$$S_i(t)=\{s_i^{l}(t)\}\quad i=1,\ldots,p;\ l=1,\ldots,n \tag{35}$$

The vector $S_i(t)$ is named the selection vector since it will be used to select the optimal number of projection vectors. Each element in the selection vector $S_i(t)$ is initialized by a number pulled from a uniform random distribution in [0, 1] and then it undergoes modifications in its velocity and position in each subsequent iteration. At any iteration t, if the position of the lth element of the vector $S_i(t)$ i.e. $s_i^l(t)$ has a value greater than 0.5 then the lth column of the position matrix $X_i(t)$, i.e., the lth projection vector, is selected, otherwise, if this value is less than 0.5, then the lth projection vector is not selected. This process is repeated for each lth dimension of the vector $S_i(t)$ in each iteration and, at the completion of the implementation of the modified GSA algorithm, from the best $X_i(t)$ solution i.e. the final W solution, those projection vectors are finally selected that are chosen according to the best solution of the $S_i(t)$ vector.

As the Modified RLEGSA algorithm now has to evolve both the candidate $X_i(t)$ matrices and the candidate $S_i(t)$ vectors in each iteration, the corresponding equations for calculating force between two particles are modified as

$$Fx_{ij}^{kl}(t)=G(t)\frac{M_{pi}(t)\times M_{aj}(t)}{Rx_{ij}(t)+\varepsilon}(x_j^{kl}(t)-x_i^{kl}(t)) \tag{36}$$

$$Fs_{ij}^{l}(t)=G(t)\frac{M_{pi}(t)\times M_{aj}(t)}{Rs_{ij}(t)+\varepsilon}(s_j^{l}(t)-s_i^{l}(t)) \tag{37}$$

$$Rx_{ij}(t)=\|X_i(t),X_j(t)\| \tag{38}$$

$$Rs_{ij}(t)=\|S_i(t),S_j(t)\| \tag{39}$$

Now, the total gravitational force on a particle i is calculated as

$$Fx_i^{kl}(t)=\sum_{j=1,j\neq i}^{p} rand_j Fx_{ij}^{kl}(t) \tag{40}$$

$$Fs_i^{l}(t)=\sum_{j=1,j\neq i}^{p} rand_j Fs_{ij}^{l}(t) \tag{41}$$

Accordingly, new acceleration update equations are developed as

$$ax_i^{kl}(t)=\frac{Fx_i^{kl}(t)}{M_{ii}(t)} \tag{42}$$

$$as_i^{l}(t)=\frac{Fs_i^{l}(t)}{M_{ii}(t)} \tag{43}$$

This gives the new velocity update equations as

$$vx_i^{kl}(t+1)=rand_i\times vx_i^{kl}(t)+ax_i^{kl}(t) \tag{44}$$

$$vs_i^l(t+1) = rand_i \times vs_i^l(t) + as_i^l(t) \tag{45}$$

Hence the new position and selection update equations are given as

$$x_i^{kl}(t+1) = x_i^{kl}(t) + vx_i^{kl}(t+1) \tag{46}$$

$$s_i^l(t+1) = s_i^l(t) + vs_i^l(t+1) \tag{47}$$

$$smax_i(t+1) = \max\{S_i(t+1)\} \tag{48}$$

$$smin_i(t+1) = \min\{S_i(t+1)\} \tag{49}$$

$$\hat{s}_i^l(t+1) = \frac{s_i^l(t+1) - smin_i(t+1)}{smax_i(t+1) - smin_i(t+1)} \tag{50}$$

$$sbin_i^l(t+1) = \begin{cases} 0, & \left|\hat{s}_i^l(t+1) < 0.5\right. \\ 1, & \left|\hat{s}_i^l(t+1) \geq 0.5\right. \end{cases} \tag{51}$$

Now normalized Euclidian distance of the ith particle from any other particle is given as

$$d_{ij} = \frac{\|X_i - X_j\|}{\max_j(\|X_i - X_j\|)} \quad \text{where } j = 1 \text{ to } p \text{ and } j \neq i \tag{52}$$

The corresponding distance vector for particle i is given as

$$D_i = \{d_{ij}\}, \quad j = 1, 2, \ldots, p \tag{53}$$

$$D_i^{asc} = sort_{ascend}(D_i) \tag{54}$$

Thus the neighborhood of the ith particle is given as

N_i^{asc} = particles corresponding to first q_i^t elements of D_i^{asc}

where

$$q_i(t) = \left[K \times \frac{t}{iter_{max}}\right] \times p \times rand[0,1] \tag{55}$$

If $q_i(t) > p$ then $q_i(t) = p$, and K the scaling constant.

$fit_i(t)$ is the fitness value of the agent i at time t.

Now, the selection variables introduced will become operative while calculating the fitness $fit_i(t)$. The feature matrix $W_i(t)$ used for evaluation of the fitness function corresponding to the candidate solution $X_i(t)$ is extracted as

$$W_i(t) = \{x_i^{kl}(t)\}_{n \times r_i^t} \quad i = 1, \ldots, p;\ k = 1, \ldots, n;\ l \in \{l; sbin_i^l(t+1) = 1\} \tag{56}$$

where r_i^t is the number of elements in $sbin_i(t+1)$ (over $l = 1, \ldots, n$) which are equal to 1.

The formulations of $fit_i(t)$, $lbest_i(t)$, $lworst_i(t)$, $m_i(t+1)$ and $M_i(t+1)$ remain unchanged as given in Section 3.3. Algorithm 3 describes the implementation of the Modified RLEGSA.

Algorithm 3. MRLEGSA.

BEGIN
Create p particles and make randomized initialization of position matrices X and selection vectors S.
Initialize iteration number $t = 1$
REPEAT:
FOR $i = 1$ to p
FOR $j = 1$ to p, $j \neq i$
Calculate normalized distance between two particles:
$d_{ij} = \frac{\|X_i - X_j\|}{\max_j(\|X_i - X_j\|)}$
END FOR
Form distance vector: $D_i = \{d_{ij}\}, j = 1, \ldots, p$
$D_i^{asc} = sort_{ascend}(D_i)$
Calculate neighborhood:
N_i^{asc} = particles corresponding
to first q_i^t elements of D_i^{asc}

$q_i^t = \left[K \times \frac{t}{iter_{max}}\right] \times p \times rand[0,1]$
if $q_i^t > p$ then $q_i^t = p$
$smax_i(t) = \max\{S_i(t)\}$
$smin_i(t) = \min\{S_i(t)\}$

FOR $l = 1$ to dim $\{S_i(t)\}$
$\hat{s}_i^l(t+1) = \frac{s_i^l(t+1) - smin_i(t+1)}{smax_i(t+1) - smin_i(t+1)}$
$sbin_i^l(t+1) = \begin{cases} 0, & \left|\hat{s}_i^l(t+1) < 0.5\right. \\ 1, & \left|\hat{s}_i^l(t+1) \geq 0.5\right. \end{cases}$
END FOR
Compose the feature matrix $W_i(t)$:
$W_i(t) = \{x_i^{kl}(t);\ x_i^{kl}(t) \in X_i(t)\}$

$i = 1, \ldots, p;\ k = 1, \ldots, n;\ l \in \{l; sbin_i^l(t+1) = 1\}$

$fit_i(t) = \frac{trace(W_i(t)^T S_w W_i(t))}{trace(W_i(t)^T S_b W_i(t))}$
END FOR
Calculate local best fitness: $lbest_i(t) = \min_{l \in \{N_i^{asc}\}} fit_l(t)$
Calculate global best fitness: $lworst_i(t) = \max_{l \in \{N_i^{asc}\}} fit_l(t)$
FOR $i = 1$ to p
Calculate mass:
$m_i(t) = \frac{fit_i(t) - lworst_i(t)}{lbest_i(t) - lworst_i(t)}$
$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^{p} m_j(t)}$

Calculate gravitational constant:
$G(t) = G(t_0) \times \left(\frac{t_0}{t}\right)^{\beta}, \quad \beta < 1, \ t_0 = 1$
FOR $j = 1$ to p, $j \neq i$
Calculate distance:
$Rx_{ij}(t) = X_i(t), X_j(t)$
$Rs_{ij}(t) = S_i(t), S_j(t)$

Calculate force:
FOR $l = 1$ to n
FOR $k = 1$ to n
$Fx_{ij}^{kl}(t) = G(t)\frac{M_i(t) \times M_j(t)}{Rx_{ij}(t) + \varepsilon}(x_j^{kl}(t) - x_i^{kl}(t))$
END FOR
$Fs_{ij}^{l}(t) = G(t)\frac{M_i(t) \times M_j(t)}{Rs_{ij}(t) + \varepsilon}(s_j^{l}(t) - s_i^{l}(t))$
END FOR
END FOR
FOR $k = 1$ to n
FOR $l = 1$ to n

Calculate total force:
$Fx_i^{kl}(t) = \sum_{j=1, j \neq i}^{p} rand_j Fx_{ij}^{kl}(t)$
$Fs_i^{l}(t) = \sum_{j=1, j \neq i}^{p} rand_j Fs_{ij}^{l}(t)$

Calculate acceleration:

$$ax_i^{kl}(t) = \frac{Fx_i^{kl}(t)}{M_i(t)}$$

$$as_i^l(t) = \frac{Fs_i^l(t)}{M_i(t)}$$

Calculate velocity:

$$vx_i^{kl}(t+1) = rand_i \times vx_i^{kl}(t) + ax_i^{kl}(t)$$

$$vs_i^l(t+1) = rand_i \times vs_i^l(t) + as_i^l(t)$$

Calculate position and selection variables:

$$x_i^{kl}(t+1) = x_i^{kl}(t) + vx_i^{kl}(t+1)$$

$$s_i^l(t+1) = s_i^l(t) + vs_i^l(t+1)$$

END FOR
END FOR
END FOR
UNTIL termination criterion is satisfied
END

4. Experimental results

Extensive experimentations have been carried out to determine the effectiveness of the proposed algorithms. At first the 2-D version of the traditional GSA proposed in Section 3.2 and the local extrema based variation (RLEGSA) proposed in Section 3.3 have been separately evaluated on two well known benchmark face databases, namely Yale A (Yale University) and ORL (ORL database) databases and the recognition accuracies have been compared with four existing comparable methods, as shown in Table 1. The Yale A database published by Yale University has 11 images each from 15 individuals with lighting, occlusion and facial emotion variations. Fig. 1 shows the 11 different images for one person in the Yale A database. The ORL database published by Cambridge

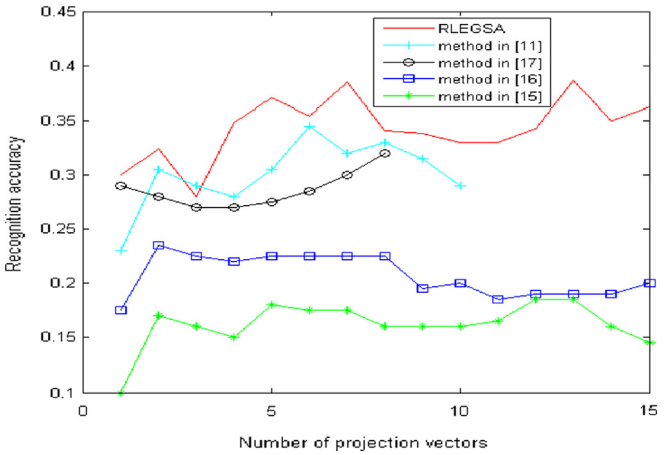

Fig. 3. A comparison of recognition accuracy vs. number of projection vectors for five competing methods, for Yale A dataset.

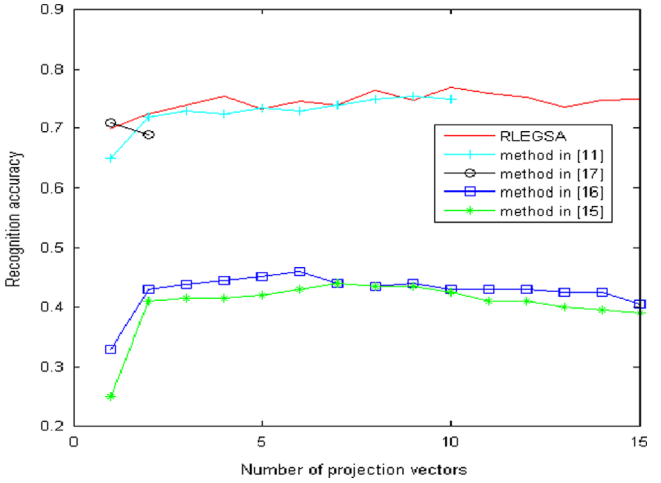

Fig. 4. A comparison of recognition accuracy vs. number of projection vectors for five competing methods, for ORL dataset.

Table 1
Top recognition accuracies for six competing methods on two benchmark face databases.

Database	Method 1 (S.C. Chen et al., 2004)	Method 2 (Zhang et al., 2005)	Method 3 (S. Chen et al., 2004)	Method 4 (Gao et al., 2008)	Proposed 2-D GSA	Proposed 2-D RLEGSA
Yale A	18.67	23.33	32.00	34.67	32.73	38.57
ORL	44.17	46.39	70.83	75.56	75.56	76.92

Fig. 1. A few sample images of one person from Yale A database.

Fig. 2. A few sample images of one person from the ORL database.

University has 10 images each from 40 individuals. Here there are also variations of occlusion and facial expressions along with changes in scale and tilting. Fig. 2 shows the 10 different images for one person in the ORL database.

During training, the first image of each individual is taken as the single training image. All the remaining images have been taken as testing samples. For classification purpose a simple nearest neighbor classifier is used (Gao et al., 2008; Code for Gao et al.). These conditions are chosen to be identical with the experimental conditions considered in other methods chosen here for comparison so that we can perform a fair comparison in an identical platform. As mentioned, the performance of the proposed method is compared with four state-of-the-art, competing, existing methods, all of whom use different variations of 2-D FLDA for a similar face recognition problem using single training image per person. The first method, proposed by S.C. Chen et al. (2004), is based on the projection approach. It reported a top recognition accuracy of 18.67% for Yale A database and 44.17% for ORL database. The second method, proposed by Zhang et al. (2005), is based on the singular value perturbation approach. It reported a top recognition accuracy of 23.33% for Yale A database and 46.39% for ORL database. The third method, proposed by S. Chen et al. (2004), is based on the non-overlapping image block approach. It reported a top recognition accuracy of 32.00% for Yale A database and 70.83% for ORL database. The fourth method, proposed by Gao et al. (2008), reported a top recognition accuracy of 34.67% for Yale A database and 75.56% for ORL database.

In comparison, by using the 2-D version of the traditional GSA, the top recognition rates achieved are 32.73% for Yale A database (comparable to method 3) and 75.56% for ORL database (comparable to method 4). However using the proposed 2-D RLEGSA top recognition rates achieved are 38.57% for Yale A database and 76.92% for ORL database which could comfortably outperform all its five competing algorithms. This shows that the proposed local extrema based version of the 2-D GSA could prove to be indeed effective in achieving better recognition accuracies for a variety of face recognition problems using single training images per person.

To make a performance comparison of the recognition rates for different competing methods with variations in the manually chosen number of projection vectors, for different face databases, the detailed performance evaluation results are plotted in Figs. 3 and 4, for Yale A database and ORL database respectively, where the results of the other four comparing methods are taken from Gao et al. (2008). The curves clearly show that the proposed RLEGSA outperforms the other competing methods in almost all choices of number of projection vectors, for both databases. This results in an ample justification of the fact that RLEGSA can be claimed as an overall superior method in comparison with the four other competing methods considered. Also, these plots signify the variability of the recognition rate with number of projection vectors for each algorithm considered and no trend from these curves can be determined which can facilitate any easy determination of choice of the optimal number of projection vectors. This fact further strengthens our logic of development of improved methods, using stochastic optimization techniques, which can make an additional search for automated selection of number of projection vectors in order to maximize the recognition rate, as proposed in MRLEGSA.

The concept of top recognition rates, as mentioned in Table 1, arises for all those methods as experiments are conducted by varying the dimension of the discriminating feature matrix that is by choosing different numbers of projection vectors. However, as mentioned before, the possibility of automated selection of the dimension of the discriminating feature matrix to obtain higher recognition rate had not been considered in the earlier four methods. Our proposed MRLEGSA has already been described in Section 3.4 which is armed with this additional feature of selection/automation. Table 2 shows the recognition performance comparisons for RLEGSA and MRLEGSA for the two benchmark databases considered. It is seen that the top recognition rates thus achieved with MRLEGSA are almost similar to the results obtained with RLEGSA except for minor deviations which may be attributed to the stochastic nature of activation of each variant of GSA. However, it should be kept in mind that the performance obtained with RLEGSA was obtained after a tedious, manual procedure of selecting different number of projection vectors by turn, activating the algorithm for each such chosen number f, noting down their performances, and then determining for which f the best performance was achieved. In MRLEGSA this entire procedure is automated involving significant reduction in computational time and

Table 2
Top recognition accuracies for two variants of GSA for two benchmark face databases.

Database	Proposed 2-D RLEGSA	Proposed 2-D MRLEGSA
Yale A	38.57	38.52
ORL	76.92	76.90

Table 4
Computation time complexity comparison of the three GSA variants.

Algorithm	Average (approx) computation time (s)	
	Yale A	ORL
2-D GSA	31.05	33.25
RLEGSA	60.88	62.79
MRLEGSA	71.53	73.46

Table 3
Quantitative performance evaluation for 2-D RLEGSA for Yale A and ORL databases.

Parameters	Significance	Yale A database	ORL database
True positive (TP)	Samples belonging to class i correctly classified as belonging to class i (summed over number of classes)	63	307
False negative (FN)	Samples belonging to class i incorrectly classified as not belonging to class i (summed over number of classes)	102	93
True negative (TN)	Samples not belonging to class i correctly classified as not belonging to class i (summed over number of classes)	715	10180
False positive (FP)	Samples not belonging to class i incorrectly classified as belonging to class i (summed over number of classes)	102	93
Precision (P)	TP/(TP+FP)	0.382	0.786
Specificity (S)	TN/(TN+FP)	0.875	0.991

effort and yet it could produce almost identical performance as RLEGSA which means it was successful, in each case, to automatically determine the best performing f optimal projection vectors, because of which the top performance of RLEGSA could also be approached using MRLEGSA. Hence the objective of introducing the innovation in MRLEGSA to automate a manual, tedious procedure and yet achieve the desired, best performance was fully achieved.

The results obtained for the two benchmark databases employing the proposed 2-D RLEGSA are further analyzed in greater detail by calculating several popular performance indices like true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN). Further analyses are also carried out to calculate other popular performance measures like precision and specificity. These results are presented in Table 3. These results demonstrate that for the ORL database the results obtained are quite encouraging as both precision and specificity values calculated are quite high. However, for the Yale A database, the precision value is quite low, although the specificity value obtained is quite satisfactory. These results are, overall, in conformation with the results presented in Table 1 where it has been demonstrated that, generally speaking, the overall performance achievable for ORL database is much better than for Yale A database, for single training sample per person situation. However, it has also been shown in Table 1 that, for the Yale A database, compared to other competing algorithms, our proposed 2-D RLEGSA could achieve quite superior performance.

Finally, a detailed discussion on the computation time complexity comparison of the proposed methods is provided in Table 4. For each algorithm, the average computation time consumed over 10 independent runs is calculated and reported here, for each of Yale A and ORL databases. These algorithms were simulated in MATLAB 2011b environment with system specifications being 8 GB RAM, INTEL i7 processor and Windows 8 OS.

The results show that 2-D GSA consumes least computation time. This is understandable because, among the three variants of GSA developed in this work, this variant is least complex and also the accuracy achievable with this method is not satisfactory. Another notable feature is that, for each run of MRLEGSA, the average time consumed is not more than 20% of the average time consumed for each run of RLEGSA. On the other hand, as mentioned before, RLEGSA requires several trial runs before the best performance can be achieved, whereas this process is automated in MRLEGSA where it can find the optimized performance in a single run. Hence it can be easily appreciated that MRLEGSA can produce satisfactory and comparable performance with respect to RLEGSA and, overall, it requires less computational time compared to RLEGSA. Hence, this analysis further justifies the utility of MRLEGSA over that of RLEGSA, as mentioned before.

5. Conclusions

In this work, a new approach to solve SVD based face recognition problems involving single training image per person is proposed using stochastic optimization approaches. The problem is solved using GSA, a contemporary algorithm recently proposed under the category of heuristic optimization methods, which attempts to determine an optimal transform matrix W such that a cost functional $J(W)$ is minimized. In this context two new variants of GSA, called the 2-D version of GSA, and a 2-D randomized local extrema based GSA (RLEGSA) have also been proposed. Then further sophistication is incorporated in this approach, by automating the process of selection of projection vectors to propose another new variant called Modified RLEGSA (MRLEGSA). Several experiments carried out for two benchmark databases i.e. Yale A and ORL databases, demonstrated that both RLEGSA and MRLEGSA produced comparable recognition accuracies and could outperform several state-of-the-art algorithms already produced for the identical problem genre. In one hand this proved the utility of proposing the random local extrema based version of GSA for this class of face recognition problems. At the same time it was aptly demonstrated that the process of automation proposed in MRLEGSA could significantly reduce the computational effort involved in RLEGSA to determine the top accuracy and yet it was able to approach the top accuracy obtained with RLEGSA.

Acknowledgments

This work was supported by University Grants Commission (UGC), India, under University with Potential for Excellence (UPE) – Phase II Scheme awarded to Jadavpur University, Kolkata, India.

References

Brunelli, R., Poggio, T., 1993. Face recognition: features versus templates. IEEE Trans. Pattern Anal. Mach. Intell. 15 (10), 1042–1052.
Chakrabarty, A., Jain, H., Chatterjee, A., 2013. Volterra kernel based face recognition using artificial bee colony optimization. Eng. Appl. Artif. Intell. 26 (3), 1107–1114.
Chen, S., Liu, J., Zhou, Z.H., 2004. Making FLDA applicable to face recognition with one sample per person. Pattern Recognit. 37 (1), 553–1555.
Chen, S.C., Zhang, D.Q., Zhou, Z.H., 2004. Enhanced (PC) ^{2}A for face recognition with one training image per person. Pattern Recognit. Lett. 25 (10), 1173–1181.
Code for Gao et al., 2008. ⟨http://www4.comp.polyu.edu.hk/~cslzhang/code.htm⟩.
Das Sharma, K., Chatterjee, A., Rakshit, A., 2012. A random spatial lbest PSO-based hybrid strategy for designing adaptive fuzzy controllers for a class of nonlinear systems. IEEE Trans. Instrum. Meas. 61 (6), 1605–1612.
Gao, Q., Zhang, L., Zhang, D., 2008. Face recognition using FLDA with single training image per person. Appl. Math. Comput. 205 (2), 726–734.
Golub, G., Loan, C., 1983. Matrix Computations. John Hopkins University Press, Baltimore, MD.
Hatamlou, A., Abdullah, S., Nezamabadi-Pour, H., 2011. Application of gravitational search algorithm on data clustering, rough sets and knowledge technology. Lect. Notes Comput. Sci. 6954, 337–346.
Hatamlou, A., Abdullah, S., Nezamabadi-Pour, H., 2012. A combined approach for clustering based on K-means and gravitational search algorithms. Swarm Evol. Comput. 6, 47–52.
Hong, Z.Q., Yang, J.Y., 1991. Optimal discriminant plane for a small number of samples and design method of classifier on the plane. Pattern Recognit. 24 (4), 317–324.
Jafri, R., Arabnia, H.R., 2009. A survey of face recognition techniques. J. Inf. Process. Syst. 5 (2), 41–68.
Jain, A.K., Prabhakar, S., 2004. An introduction to biometric recognition. IEEE Trans. Circuits Syst. video Technol. 14 (1), 4–20.
Lu, J., Plataniotis, K.N., Venetsanopoulos, A.N., 2003. Face recognition using LDA-based algorithms. IEEE Trans. Neural Netw. 14 (1), 195–200.
Mansouri, R., Nasseri, F., Khorrami, M., 1999. Effective time variation of G in a model universe with variable space dimensionPhys. Lett. A 259 (3–4), 194–200.
ORL database, 2002. ⟨http://www.cam-orl.co.uk/facedatabase.html⟩.
Pal, K., Saha, C., Das, S., Coello, C.A. Coello, 2013. Dynamic constrained optimization with offspring repair based gravitational search algorithm. In: Proceedings of 2013 IEEE Congress on Evolutionary Computation (CEC 2013), pp. 2414–2421.
Rashedi, E., Nezamabadi-pour, H., Saryazdi, S., 2009. GSA: a gravitational search algorithm. Inf. Sci. 179 (13), 2232–2248.
Suganthan, P.N., 1999. Particle swarm optimizer with neighborhood operator. In: Proceedings of the Congress on Evoutionary Computation. Washington, DC, pp. 1958–1962.
Sun, G., Zhang, A., 2013. A hybrid genetic algorithm and gravitational search algorithm for image segmentation using multilevel thresholding. Pattern Recognit. Image Anal. Lect. Notes Comput. Sci. 7887, 707–714.
Tan, X., Chen, S., Zhou, Z., Zhang, F., 2006. Face recognition from a single image per person: a survey. Pattern Recognit. 39 (9), 1725–1745.
Tian, Q., Fainman, Y., Lee, Sing H., 1988. Comparison of statistical pattern-recognition algorithms for hybrid processing II. Eigenvector-based algorithm. J. Opt. Soc. Am. A 5 (10), 1670–1682.
Tu, C.J., Chuang, L.Y., Chang, J.Y., Yang, C.H., 2007. Feature selection using PSO-SVM. Int. J. Comput. Sci. (IAENG) 33 (1) (IJCS_33_1_18).
Yale University, 2002. Face Database. ⟨http://cvc.yale.edu/projects/yalefaces/yalefaces.html⟩.

Ye, J., Janardan, R., Li, Q., 2004. Two-dimensional linear discriminant analysis, In: Proceedings of Neural Information Processing Systems (NIPS). Vancouver, BC, Canada.

Yu, H., Yang, J., 2001. A direct LDA algorithm for high-dimensional data with application to face recognition. Pattern Recognit. 34, 2067–2070.

Zhang, D., Chen, S.C., Zhou, Z.H., 2005. A new face recognition method based on SVD perturbation for single example image per person. Appl. Math. Comput. 163 (2), 895–907.

Zhao, W., Chellappa, R., Phillips, P.J., Rosenfeld, A., 2003. Face recognition: a literature survey. ACM Comput. Surv. 35 (4), 399–458.

Zhu, P., Zhang, L., Hum, Q., Shiu, S.C.K., 2012. Multi-scale patch based collaborative representation for face recognition with margin distribution optimization. In: Proceedings of the Computer Vision – ECCV 2012, LNCS, vol. 7572. Springer, Berlin Heidelberg, pp. 822–835.