Neural Comput & Applic (2018) 30:2643–2660
https://doi.org/10.1007/s00521-017-2874-2

ORIGINAL ARTICLE

Radial basis function neural network-based face recognition using firefly algorithm

Vandana Agarwal[1] · **Surekha Bhanot**[2]

Received: 25 June 2015 / Accepted: 10 February 2017 / Published online: 20 February 2017

Abstract This paper presents an adaptive technique for obtaining centers of the hidden layer neurons of radial basis function neural network (RBFNN) for face recognition. The proposed technique uses firefly algorithm to obtain natural sub-clusters of training face images formed due to variations in pose, illumination, expression and occlusion, etc. Movement of fireflies in a hyper-dimensional input space is controlled by tuning the parameter gamma (γ) of firefly algorithm which plays an important role in maintaining the trade-off between effective search space exploration, firefly convergence, overall computational time and the recognition accuracy. The proposed technique is novel as it combines the advantages of evolutionary firefly algorithm and RBFNN in adaptive evolution of number and centers of hidden neurons. The strength of the proposed technique lies in its fast convergence, improved face recognition performance, reduced feature selection overhead and algorithm stability. The proposed technique is validated using benchmark face databases, namely ORL, Yale, AR and LFW. The average face recognition accuracies achieved using proposed algorithm for the above face databases outperform some of the existing techniques in face recognition.

✉ Vandana Agarwal
vandana@pilani.bits-pilani.ac.in;
vandana.bits.pilani@gmail.com

Surekha Bhanot
surekha@pilani.bits-pilani.ac.in; surekha0057@gmail.com

[1] Department of Computer Science and Information Systems, Birla Institute of Technology and Science (BITS), Pilani, Rajasthan 333031, India

[2] Department of Electrical and Electronics Engineering, Birla Institute of Technology and Science (BITS), Pilani, Rajasthan 333031, India

Keywords Face recognition · Radial basis function neural network · Firefly algorithm · RBF center selection

1 Introduction

Machine-based face recognition has been used effectively in controlled imaging conditions in applications such as passports, credit cards, criminal databases and password authentication [1, 2]. Surveillance-based images capture variations with respect to light source direction, pose, facial expressions, etc. These variations may or may not match with those existing in the training images of persons and pose challenge in face recognition research. The need of the times is to have computationally efficient systems capable of handling these variations under unconstrained environment so as to determine the identity of the person in real time [3]. Many researchers have been working toward handling these variations [4–6] using different techniques. Most studies in face recognition research aim at improving the average recognition accuracy with a focus to make the algorithms more robust for fast face recognition.

Earlier studies on face recognition were based on approaches using principal component analysis (PCA) and geometrical features, etc. [7, 8]. The limitation of geometric feature-based approach was that it was sensitive to scale and dimension differences. The disadvantage of PCA-based approach is that it is sensitive to illumination and face expression-based changes. Wiskott et al. [9] used elastic bunch graph matching (EBGM) for face recognition where a face was represented as a labeled graph of nodes. It was observed that PCA-based eigenfaces and elastic bunch graph matching were sensitive to illumination variations [10]. To overcome these limitations, the advantages of both techniques were combined and used with the back-propagation neural network. Fisher linear discriminant analysis (FLD) method was developed to overcome the limitation of the eigenfaces [11]. Chellappa et al. [2] presented an overview of the challenges and issues in face recognition problem in today's scenario and illustrated that the face recognition problem is hard.

Computational intelligence-based techniques emulating human intelligence, which include neural networks, evolutionary algorithms and fuzzy computing, are reported to efficiently handle challenges of face recognition. Neural networks possess the best approximation capability and have been used in various pattern recognition and classification problems [12–14]. Convolution neural network and multilayer perceptron (MLP) coupled with the back-propagation algorithm, have also been used in face recognition [15, 16]. Back-propagation training algorithm is found to be computationally intensive due to its slow speed of convergence and is not guaranteed to reach global optima [23]. Radial basis function neural networks (RBFNN) consisting of only one hidden layer have been used in face recognition task due to their good approximation and fast computational capabilities [13, 17–19]. Radial basis functions at hidden layer perform nonlinear mapping of input face data to the linearly separable data in hidden hyperspace. The hidden layer design issues include determination of centers of hidden neurons called RBF units, number of hidden neurons, choice and shape of basis functions [1, 13, 23]. The approaches commonly used for obtaining RBFNN centers are based on random subset selection of input data, selection of subset using orthogonal least squares, Gaussian mixture models and clustering algorithms [13].

The design of RBFNN has been of research interest in face recognition application. Different approaches such as pruning and growing [3], fuzzy hybrid learning [20], polynomial-based RBF neural networks [21], point symmetry distance [22] and self-adaptive approach [23] were used in RBFNN design for face recognition. Er et al. [3] used PCA for feature extraction and trained the RBFNN centers by splitting a cluster, iteratively obtained. Haddadnia et al. [20] proposed a technique named as fuzzy hybrid learning algorithm (FHLA) in which they combined PCA and shape information to extract features from face images. The FHLA technique used cluster validity number to determine the number of hidden neurons in the RBFNN structure and used fuzzy-C-means (FCM) algorithm to initialize the RBF parameters. The major drawback of FHLA technique is that the network has to be excessively tuned for best performance. Polynomial RBFNN was used in face recognition in which the input space was partitioned using FCM [21]. The FCM technique has limitation of being sensitive to initialization and getting trapped at local optima [24]. RBFNN training for face recognition has been done using a modified *k*-means clustering algorithm using point symmetry distance [22]. A self-adaptive RBFNN design approach for face recognition used each of the training images as hidden neuron initially and imposed a confidence measure with each neuron [23].

Clustering is considered as one of the most difficult and challenging problems in machine learning and is NP-hard [25–27]. The clustering algorithms such as *k*-means, FCM, expectation maximization (EM) get trapped in the local optima, while evolutionary algorithms provide near optimal solution in reasonable time [27]. Various evolutionary algorithms such as genetic algorithm (GA), particle swarm optimization (PSO) and artificial bee colony (ABC) have been used in the design of RBFNN in various applications such as function approximation, web source classification and machine learning tasks [28–32].

A meta-heuristic optimization algorithm named firefly algorithm (FA) was proposed by Yang [34]. The firefly algorithm has been used in various tasks such as breast tumor classification [35], object tracking application [36] and multi-threshold image segmentation [37]. A detailed survey of various other swarm intelligence-based algorithms and variants of firefly algorithms has been presented illustrating the significance of firefly algorithm [38, 39]. Firefly algorithm is efficient for multi-modal, nonlinear optimization problems and leads to true optimality with its ability to automatically divide the search space into subgroups and is reported to perform better than PSO and GA [38]. Senthilnath et al. [33] used firefly algorithm in clustering application and established that the technique was reliable, efficient, robust, and it outperformed PSO and ABC techniques.

It is observed that the potential of firefly algorithm has not been used in RBFNN center selection for face recognition. Therefore, we take advantage of the fast and efficient convergence of the firefly algorithm and combine it with the learning capabilities of RBFNN for high-speed face recognition.

1.1 Contributions of our paper

The proposed firefly inspired RBFNN (FRBFNN) algorithm captures the most natural clustering of high-dimensional face training samples having variations in illumination, pose, expression, accessories, etc. The main contributions of this research are as follows

(i) Obtaining optimal number and centers of RBF units for hyper-dimensional face data.
(ii) Detailed analysis on parameters selection to improve convergence and face recognition performance.
(iii) Testing on popular benchmark face databases such as Olivetti Research Lab (ORL) [42], Yale [43],

AR [44, 45] and Labeled Faces in the Wild (LFW) [46, 47].

The proposed technique has been found to be computationally efficient and converges fast to obtain optimal centers in very less number of iterations. Another strength of the proposed algorithm is reduced feature selection overhead which also makes the face recognition computationally efficient.

2 Radial basis function neural networks

Radial basis function neural networks (RBFNN) are the feedforward neural networks. The architecture of RBFNN shown in Fig. 1 consists of three layers: input layer, hidden layer and output layer. Input layer receives the input and passes the information to the hidden layer [13]. The hidden layer consists of radial basis function (RBF) units, also called as hidden neurons. Each jth RBF unit has an associated center C_j, spread (σ_j) and the basis function ϕ_j.

The nonlinear basis functions ϕ_j are functions of radial distance of input from the center of the jth RBF unit. Commonly used basis functions such as Gaussian, multiquadric, inverse multiquadric and thin spline are used in RBFNN. In this work, the most commonly used Gaussian basis function was represented as

$$\emptyset_j(x) = e^{-\frac{\|x - C_j\|^2}{2\sigma_j^2}} \tag{1}$$

where C_j represents the center of the jth RBF unit. The parameter σ is known as spread of the radial basis function and represents its width. Let the d-dimensional input feature vector be $P \in \mathbb{R}^d$, and the number of RBF units be h, then RBFNN can be described as a mapping from $\mathbb{R}^d \rightarrow \mathbb{R}^h$ ($h \gg d$). The ith output $y_i(x)$ of the RBFNN is

$$y_i(x) = \sum_{j=1}^{h} \emptyset_j(x) \times w_{i,j} \tag{2}$$

where $w_{i,j}$ is the weight of the connection between the jth RBF unit and the ith output neuron.

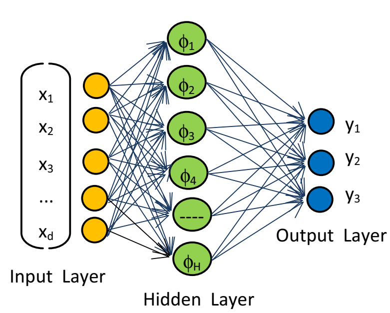

Fig. 1 Three-layered architecture of RBFNN

3 Firefly algorithm

The firefly algorithm (FA) was proposed by Yang [34] to solve nonlinear optimization problems. The algorithm was inspired by the natural fireflies seen around in the tropical summer. The natural fireflies have a bioluminescence that causes a flashing light coming out of the fireflies. A population of fireflies is generated randomly at the beginning, and the fireflies move in the search space. A less brighter firefly is said to move toward a brighter firefly, where brightness of a firefly is defined by its fitness value. Fireflies keep moving till they accumulate at peaks of maximum fitness. The firefly explores the search space due to attractiveness, exploring its neighborhood using randomized steps. Yang established that firefly algorithm was suitable for multi-modal optimization applications. FA was found to be superior to both GA and PSO in terms of efficiency and success rate. Also it was established that FA has more potential to solve NP-hard problems as well. In this study, original firefly algorithm is applied for RBFNN center selection for face recognition.

4 Proposed FRBFNN algorithm for face recognition

In this paper, an adaptive technique is developed for optimal number and center selection of the RBF units which capture variations in illumination, pose, expressions and accessories worn in face images by grouping together approximately similar images to form sub-clusters. Each person's training images are subjected for sub-clustering, and the cumulative number of sub-clusters thus evolved, for all persons, produces the number of neurons in the hidden layer. In this section, the details of the proposed FRBFNN algorithm are presented.

4.1 Firefly design

A firefly is viewed as a hyper-dimensional polygon surface comprising of K d-dimensional vertices representing K possible centers of RBF units, where $K > 1$. A generalized firefly is proposed to be the collection of centers of RBF units as shown below

$$F = \begin{bmatrix} C_1^F \\ C_2^F \\ C_3^F \\ \vdots \\ C_K^F \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} & \cdots & c_{1d} \\ c_{21} & c_{22} & c_{21} & \cdots & c_{2d} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ c_{K1} & c_{K2} & c_{K3} & \cdots & c_{Kd} \end{bmatrix} \tag{3}$$

where center C_i^F is a d-dimensional vector given by $\langle c_{i1} \quad c_{i2} \quad c_{i3} \quad \ldots \quad c_{id} \rangle$. The firefly proposed above is of dimension $K \times d$ which moves in a $[K \times d] + 1$-

dimensional space. The additional dimension represents the dimension of the fitness function.

4.2 Fitness function

The fitness of a firefly in the context of RBFNN design is negative of the sum of squared distances of d-dimensional input feature vectors from the nearest center. Consider the total number of training images to be M in each person class. Each face image T_u, for $u = 1, 2, \ldots, M$, is represented as a d-dimensional feature vector $\langle f_1^u \quad f_2^u \quad f_3^u \quad \ldots \quad f_d^u \rangle$. A sub-cluster S_i is constructed as the collection of all feature vectors nearest to the ith center C_i^F as

$$S_i = \left\{ \bigcup_{1 \le u \le M} T_u \,|\, ||C_i^F - T_u|| < ||C_j^F - T_u|| \quad \forall j = 1, 2, \ldots, K,\ i \ne j \right\} \tag{4}$$

The fitness of firefly F, computed as G^F, is given by

$$G^F = -\sum_{i=1}^{K} \sum_{\substack{u=1 \\ T_u \in S_i}}^{M} \left\| C_i^F - T_u \right\|^2 \tag{5}$$

where $||C_i^F - T_u||$ is the Euclidean distance between the ith center represented by firefly F, i.e., C_i^F, and the training image feature vector. The mean vector of the face feature vectors belonging to each sub-cluster is considered as the center of the RBF unit.

4.3 Firefly movement

A firefly, while moving in the $K \times d$-dimensional search space, acquires different positions and changes its brightness. The fireflies acquire new positions within the search space boundary and are not allowed to go outside the boundary. If a new position falls beyond the boundary, the firefly is given a position at the boundary. The attractiveness (β) between the two fireflies at distance r apart is expressed as

$$\beta = \beta_0 \exp^{-\gamma r^2} \tag{6}$$

where $\beta_0 = 1$ is the maximum attractiveness at $r = 0$ and γ is the light absorption coefficient. The attractiveness β is relative and is seen in the eyes of other firefly at a distance r. It varies with distance between fireflies F_i and F_j. The firefly F_i moves to a brighter firefly F_j by a distance and acquires a new position F_i^{t+1} at $(t + 1)$th iteration which is defined by

$$F_i^{t+1} = F_i^t + \beta_0 \exp^{-\gamma r^2} ||F_j - F_i|| + \alpha \in_i \tag{7}$$

where F_i^t is the previous position of the firefly [34]. The second term is due to the attractiveness between the two fireflies F_i and F_j. The randomization parameter α, with values varying in [0 1], in the third term causes randomness in the movements, and $\in_i$ is a random number drawn from Gaussian distribution. The firefly movement consists of two types of shifts from the current position of the firefly-*heuristic shift* and *random shift*, respectively, characterized by the second and third terms of (7). Heuristic shift is the distance travelled by a firefly due to the attractiveness of the other firefly and depends mainly on the parameter γ and the distance r between the two fireflies. Therefore, the value of γ is required to be known appropriately.

4.4 Algorithm parameters

The parameters γ (gamma), number of fireflies (Q) and number of iterations (I) play an important role in algorithm convergence resulting in improved face recognition.

(i) *Gamma (γ)* It controls the speed of the movement of fireflies. When the value of γ is large, the speed of the firefly is less and it explores the search space more densely. When the value of γ is small, the speed of the movement of firefly is more and it explores the search space less densely. If the firefly moves faster, it may miss the optima, and if it moves very slow, then it may never reach the optima.

(ii) *Number of fireflies (Q)* The number of fireflies significantly affects the efficiency and performance of the algorithm. A sufficiently large population of fireflies prevents the algorithm to get trapped in the local optimal solution, but it is computationally expensive to work with large number of fireflies, while very small number of fireflies may lead to sub-optimal solution.

(iii) *Iterations (I)* An iteration consists of pairwise interaction of all fireflies where fireflies look at each other and move if the other firefly is brighter. For high-speed face recognition, the parameters γ and Q are tuned in such a way that the fireflies explore the search space in less number of iterations efficiently.

4.5 Convergence metric

Consider a set of Q fireflies as $\{F_1, F_2, \ldots, F_Q\}$ searching the solution space. The fireflies in this process accumulate near the fireflies having peaks of maximum fitness. Let S_{conv} be the set of fireflies which converge close to the brightest firefly and is given as

$$S_{conv} = \left\{ F_i | 1 \le i \le Q \text{ and } \left| G^{F_i} - G^{F_{best}} \right| < T_{Lim} \right\} \tag{8}$$

where T_{Lim} is the tolerance limit within which fireflies converge. We define the convergence as follows.

(i) *Firefly convergence* The population of fireflies is said to have converged, if the size of set S_{conv}, i.e., $|S_{conv}|$ is greater than or equal to the user-defined threshold (say T_F).

$$|S_{conv}| \geq T_F \tag{9}$$

(ii) *Sub-cluster convergence* Firefly movement is continued, and the formation of sub-clusters is observed. Each time the brightest firefly produces the same sub-clusters as that produced in the previous evaluation, a count (say *count*) is increased. If a new sub-cluster is obtained using the best firefly, then the value of *count* is reinitialized to zero. When *count* reaches a threshold (say T_{count}), the firefly movement is terminated. Using firefly fitness $G^{F_{best}}$ of the brightest firefly and the value of *count*, an error is computed at each evaluation as

$$\text{Error} = G^{F_{best}} \times \left(1 - \frac{count}{T_{count}}\right) \tag{10}$$

(iii) *Algorithm convergence:* The algorithm converges when the square of error reduces below a defined threshold, i.e., $\text{Error}^2 < 0.0001$ (threshold), number of converged fireflies is greater than or equal to the threshold T_F or the number of iterations reaches its maximum.

4.6 Algorithm development details

The proposed FRBFNN algorithm is shown as a flowchart in Fig. 2. The centers of hidden neurons or RBF units are obtained from the sub-clusters obtained using the proposed algorithm. The number of sub-clusters for each person is not required as an input for face recognition as the algorithm learns from the given training data and evolves the sub-clusters. The stepwise details are as follows:

Step 1 Feature extraction

The feature vectors (T_u: $u = 1, 2, \ldots, M$) of each person's training images are obtained using discrete cosine transform (DCT). The features are extracted from the upper left corner of the transformed images of all training images, and the proposed algorithm does not have the feature selection overhead. The variables h and R are used to store the number and centers of the evolved neurons.

Step 2 Computation of the best firefly containing optimal centers

Each person's training feature vectors are used to obtain the best firefly. For each person, identified as *personIndex*, Q fireflies are initially generated randomly, which interact pairwise and move to the brighter firefly in each iteration. An iteration consists of $Q \times Q$ evaluations each comprising of the fitness comparison and firefly movement, if required. The pairwise interaction of the fireflies is implemented using two nested loops over two variables i and j as shown in Fig. 2. The error is computed using (10) at each evaluation, and the fireflies keep moving till the algorithm converges. The best firefly F_{Best} with maximum fitness is computed for sub-cluster formation in the next step.

Step 3 Sub-cluster formation using best firefly and mean vector computation

Each feature vector is used to find its belongingness to one of the sub-clusters with feasible centers represented by firefly F_{Best}, at C_p: $p = 1, 2, \ldots, K$. Let S_p be the set of such feature vectors belonging to C_p, then if S_p is empty, it is ignored. If S_p is not empty, then a mean vector C_{Mean} is obtained by averaging the feature vectors in S_p. C_{Mean} is appended as a row of matrix *Cen_Mat*, and the count of evolved neurons e is incremented by one. The process is repeated for $p = 1, 2, \ldots, K$.

Step 4 Evolution of hidden neurons for each person class (RBF units)

The centers and number of RBF units obtained as *Cen_Mat* and e add to the variables R and h, respectively. The rows of matrix R represent the centers of the RBF units, and h is the total number of evolved neurons.

Step 5 Repeat Steps 2, 3 and 4 for all person classes

The output of the algorithm is the total number of hidden neurons or RBF units and their optimal centers.

4.7 Performance measure

The performance of the proposed algorithm is measured by the percentage of correctly recognized images from the total number of test images. Average recognition accuracy is computed as

$$\text{Average recognition accuracy} = \frac{\sum_{i=1}^{i=q}\left(\frac{n_c^i}{n_t}\right)}{q} \times 100 \tag{11}$$

where q is the total number of runs (taken as 10 in this study). The terms n_c^i and n_t represent the number of correctly recognized faces at ith iteration and total number of test images taken in each iteration, respectively.

4.8 Time complexity

The worst-case time complexity of the proposed technique is given by

Variable	*Detail*
Q	Number of Fireflies
K	Maximum number of sub-clusters in each person class
M	Total number of training face images
R	RBF center matrix with each row containing a center vector
I_{max}	Number of maximum iterations in which fireflies keep moving
Cen_Mat	Sub- Cluster Matrix with each row containing a center
e	The number of evolved neurons in each person class
h	Total Number of evolved neurons(RBF units)
F_{best}	Brightest Firefly
n	Total number of persons in the face database
Error	Error term defined in (10)

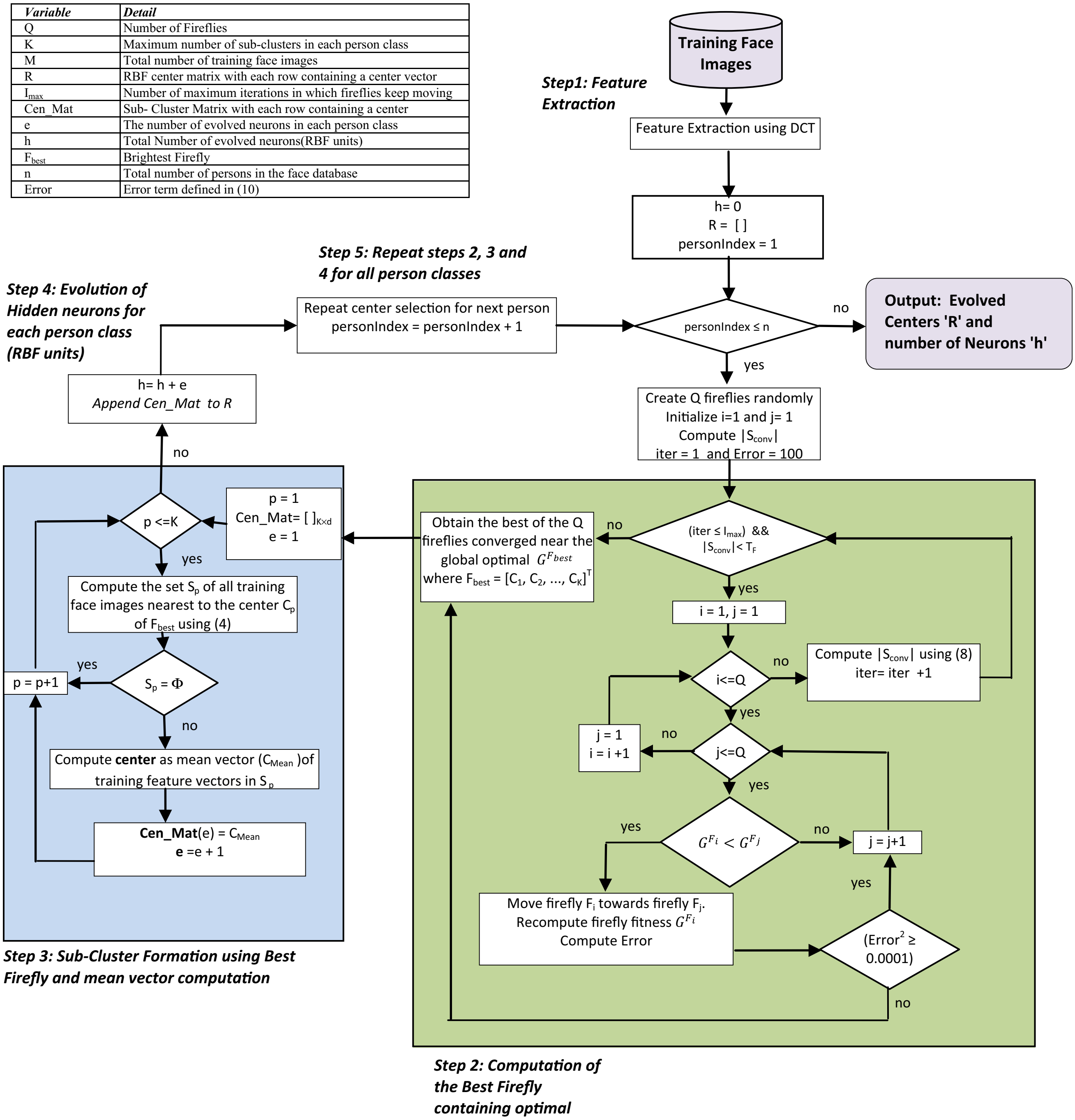

Fig. 2 Flowchart of the proposed FRBFNN algorithm for number of neurons in hidden layer and their centers training for face recognition

$$\text{Time complexity} = O\left(I_{\max} K Q^2 n^3\right) \qquad (12)$$

given $n^2 \gg d$ and $M \ll n$, where Q, K, n, $I_{\max}$, M and d are number of fireflies, maximum number of sub-clusters, total number of persons in the face database, maximum number of iterations, number of training images per person and feature dimension, respectively.

5 Face databases used

Four benchmarked face databases namely Olivetti Research Lab (ORL, also known as AT&T) (Fig. 3a), AR (Fig. 3b), Yale (Fig. 3c) and Labeled Faces in the Wild (LFW) (Fig. 3d) are used. The details of the face databases are mentioned in Table 1.

The experiments were performed on Intel i5-2430 M CPU (processor speed of 2.40 GHz and RAM of 8 GB). Ten independent runs were used for each observation. An independent run includes training using proposed FRBFNN technique followed by testing of unseen face images using a disjoint set of randomly selected test images. The images in the experimentation were used as per the description given in Table 2. The DCT has been used as the feature extraction method [41], and only upper left triangular coefficients were used as features. The details of face images used in this study are shown in Table 2.

6 Parameter selection for the proposed FRBFNN technique

In this study, the tuning of the parameters gamma (γ), number of fireflies (Q) and number of iterations (I) for the face recognition problem is done through experiments as discussed below. The effect of the above parameters on the algorithm convergence is investigated using 20 iterations. The convergence thresholds used are $T_{Lim} = 0.01$, $T_F = 12$ and $T_C = 2000$. The automatic tuning of the parameters is a very tough hyper-optimization problem [40] and is not the focus of present study. Total number of features used for images from ORL, Yale, AR and LFW face databases are 45, 30, 260 and 65, respectively (based on analysis reported in Sect. 7.1).

6.1 Effect of parameter γ on algorithm convergence

Effect of γ on all databases ORL, AR, Yale and LFW was investigated for improved high-speed face recognition (Fig. 4). The number of fireflies is taken as 20, and value of γ was changed from 1 to 21 for ORL face database, 1–11 for Yale and LFW face databases and from 0.00001 to 1.0 for the AR face database. These are the ranges of γ values in which the proposed algorithm was found to converge.

AR face database is a special database which captures a huge variation in illumination, expression and occlusion, making feature input space very large in dimension, so fireflies need faster movements to cover such huge hyperspace to converge; hence, γ value is taken very low (0.00001–1.0). It is evident from Fig. 4 that large values of γ lead to slow convergence. As compared to other methods reported in [22] and [23], where the algorithm converges in 6500 and 15,000 epochs, respectively, the proposed FRBFNN algorithm converges very fast in 5–15 iterations. Very large values of γ reduce the speed of fireflies, and they are less likely to reach the brightest firefly in the given number of iterations.

6.2 Effect of parameter γ on average recognition accuracy

Performance in terms of average recognition accuracy, obtained for test images, and stability of proposed

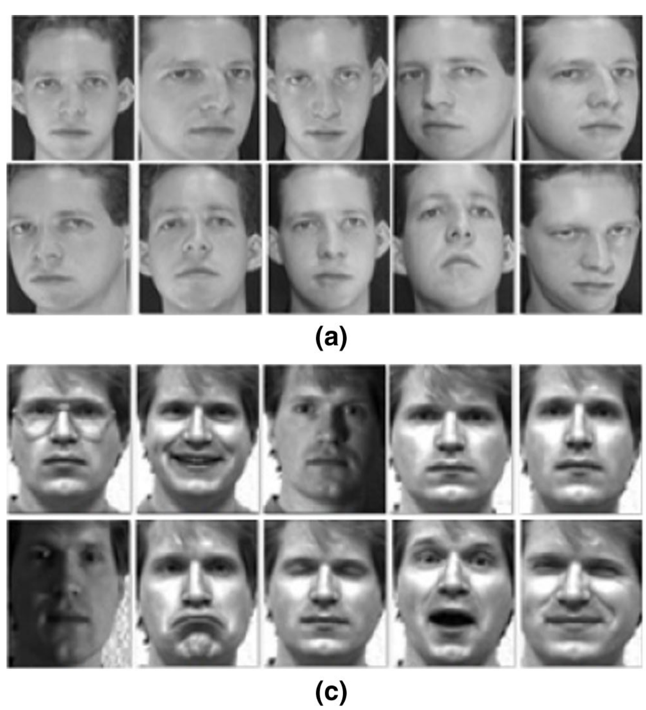

Fig. 3 Sample face images of one person each from face databases **a** ORL, **b** AR, **c** Yale face databases and **d** unconstrained faces from LFW face database

Table 1 Face databases

Face database	Developed at	Year	Number of persons	Total face images per person	Size (in pixels)	Image format	Gray/color depth	Background	Variations captured during face image acquisition				
									Pose	Illumination	Emotions/expression	Accessories/occlusion	Temporal
ORL [42]	AT&T laboratories, Cambridge, UK	1992–1994	40	10	92 × 112	PNG	8-Bit gray level	Dark, homogeneous	Upright, frontal (minor side poses)	Minor	Eyes open/closed, smiling/not smiling	Glasses	Some images
Yale [43]	Yale University	–	15	11	243 × 320	GIF	Gray	–	Frontal	Center light, left light, right light	Normal, sad, happy, sleepy, surprised, wink	Glasses, no glasses	–
AR [44, 45]	Computer Vision Center at U.A.B.	1998	126	26	120 × 165	BMP	24-Bit depth	–	Frontal	Left light, right light, all side light	Neutral, smile, anger, scream	Sun glasses, scarf	Two sessions
LFW [46, 47]	MIT	2007	5749	Varying Numbers	250 × 250	JPG	Color	Natural	Unconstrained				

algorithm is studied for varying values of γ which would converge in fixed number of iterations (maximum 20) and is shown in Table 3. The detail of each face database in Table 3 is mentioned as *name of the database/number of persons/number of training images per person/number of testing images used per person/number of fireflies/maximum number of iterations/number of features*. It is observed that as γ increases, average number of iterations increases and average number of converged fireflies decreases, implying slow convergence. In order to obtain near optimal sub-clusters, it is essential that a large number of fireflies converge in moderate number of iterations. If a large number of fireflies converge in very less iterations, this may imply a sub-optimal solution.

The stability, represented by standard deviation from the average recognition accuracy, is also of concern while arriving at the most suitable value of γ. The best average recognition accuracies ($\pm$standard deviation) obtained for ORL, Yale, AR and LFW face databases are 97.35% ($\pm$0.97), 99.83% ($\pm$0.53), 93.15% ($\pm$3.25) and 55.50% ($\pm$8.62), respectively. The values of γ producing best accuracies are 5 for ORL, 3, 5, 7, 9 for Yale, 0.00001 for AR and 7 for LFW face databases. Average numbers of converged fireflies to reach above accuracies are 16.23, 14.07, 19.10 and 1.05, respectively, for the four databases. However, this number for LFW, which is 1.05, is less than the threshold T_F; therefore, we select the second best accuracy as 52.50% for LFW obtained with 16.90 converged fireflies for γ equal to one. The above accuracies were arrived in 6.41, 8.78, 5.95 and 7.08 average number of iterations where average is taken over all person classes and all independent runs. Standard deviations for ORL and Yale are 0.97 and 0.53, respectively, which are very low and support the stability of algorithm for these databases, while it is 3.25 for AR and 7.05 for LFW due to the huge variations in uncontrolled environment which is still challenging to handle.

6.3 Effect of number of fireflies on algorithm convergence

The convergence of fireflies has been investigated with respect to varying values of number of fireflies (Fig. 5). The number of iterations is kept fixed as 20 in order to ensure fast computation. The number of fireflies is varied from one to 40 in the step of size 2 for ORL and AR face databases, while a longer range of variations in number of fireflies is taken for Yale and LFW face databases which is taken as 1–58 in step of size 3. The error is computed using (10) till the algorithm terminates, and the square of the error thus obtained is plotted against the number of fireflies. The convergence is investigated with respect to varying values of γ for the given range of fireflies variations.

Table 2 Details of face images used in this study

Face database	Cropping	Resizing	Number of persons	Number of training images per person	Number of testing images per person	Selection of training images	Selection of testing images
ORL	No	No	40	5	5	Any 5 selected randomly	Remaining 5 of 1–10
AR	Yes	60 × 83	40	13	6	1–13 fixed	14, 15, 17, 19, 22, 26
Yale	Yes	92 × 112	15	6	4	Any 6 selected randomly	Remaining 4 of 1-10
LFW	Yes	92 × 112	10	8	8	Any 8 selected randomly	Remaining 8 of 1–16

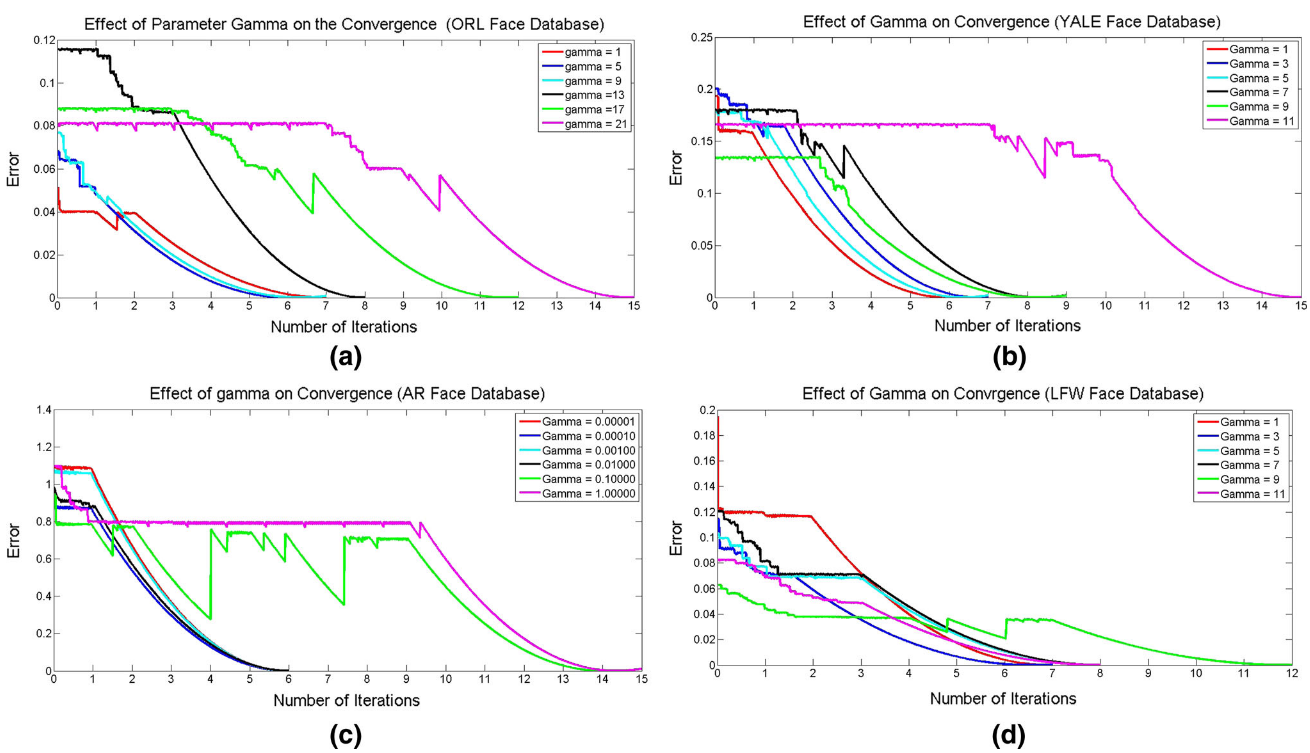

Fig. 4 Effect of parameter γ on the algorithm convergence **a** ORL, **b** Yale, **c** AR and **d** LFW face databases

It is observed that number of fireflies less than 11 is not sufficient to converge in 20 iterations. The choice of number of fireflies also depends on the value of γ. For ORL face database, for γ equal to 1, 5 and 9, if number of fireflies is taken as 11, the error reduces to zero implying algorithm convergence (Fig. 5a). Similarly, for AR face database, 11 fireflies are sufficient for algorithm convergence for all values of γ equal to 0.00001–1.0 in multiples of 10 (Fig. 5c). The convergence trend for Yale and LFW is observed using longer x-axes with large range of fireflies with γ equal to 1, 3 and 5. It is evident from Fig. 5b, d that γ equal to 5 leads to fluctuations in value of squared error and convergence is not ensured even with large number of fireflies. Therefore, it is recommended not to use this value of γ for center selection. Values of γ taken as 1 and 3 for Yale face database result in convergence using 13 and 25 fireflies, respectively (Fig. 5b). For LFW face database, values of γ equal to 1 and 3 lead to convergence using 16 fireflies. The minimum numbers of fireflies required for ORL, Yale, AR and LFW therefore are 11, 13, 11 and 16, respectively, using appropriate values of γ which enable algorithm convergence in 20 iterations.

6.4 Proposed parameters

The behavior of the proposed algorithm in hyper-dimensional face training input space shown in Figs. 4 and 5 is summarized in the parameter selection of gamma (γ), number of fireflies (Q) and the number of iterations in Table 4. The maximum numbers of fireflies and iterations are 20 each.

Table 3 Effect of parameter γ on average recognition accuracy using the proposed FRBFNN algorithm

γ	Average recognition accuracy ± standard deviation	Average number of converged fireflies	Average number of iterations	Number of evolved neurons
ORL face database (ORL/40/5/5/20/20/45)				
1	96.05 ± 2.18	16.36	5.65	114
5	**97.35 ± 0.97**	**16.23**	**6.41**	123
9	97.35 ± 1.94	16.51	6.74	120
13	96.30 ± 1.67	15.76	6.87	121
17	96.45 ± 1.95	15.50	7.13	115
21	97.35 ± 1.62	15.61	7.54	116
Yale face database (Yale/15/6/4/20/20/30)				
1	99.33 ± 1.41	14.71	6.44	50
3	**99.83 ± 0.53**	**14.07**	**8.78**	53
5	**99.83 ± 0.53**	13.27	12.51	52
7	**99.83 ± 0.53**	10.73	16.62	49
9	**99.83 ± 0.53**	5.90	19.14	53
11	99.67 ± 0.70	3.14	19.70	50
AR face database (AR/40/13/6/20/20/260)				
0.00001	**93.15 ± 3.25**	**19.10**	**5.95**	295
0.0001	92.35 ± 2.04	19.43	5.90	298
0.001	91.95 ± 2.40	19.27	5.98	297
0.01	91.75 ± 1.06	19.46	6.37	290
0.1	92.80± 2.61	19.42	6.92	284
1.0	91.65 ± 2.24	18.85	9.01	274
LFW face database (LFW/10/8/8/20/20/65)				
1	**52.50 ± 7.05**	**16.90**	**7.08**	51
3	51.38 ± 6.78	16.32	11.20	53
5	50.50 ± 6.16	7.18	1.87	49
7	55.50 ± 8.62	1.05	20	49
9	49.25 ± 9.76	1.01	20	49
11	51.75 ± 8.06	1.10	20	48

Bold values indicate best results

7 Results and discussion

With optimal values of parameter γ chosen for different face databases in Sect. 6, effects of feature dimension, number of training images and sub-clustering on average recognition accuracy, firefly convergence and evolution of neurons, etc., are studied. Maximum numbers of iterations and fireflies are 20 each.

7.1 Effect of number of features on average recognition accuracy

Feature dimension is investigated for number of features varying from 5 to 100 features for ORL, Yale and LFW and 20–300 for AR face databases. It is evident from the curves shown in Fig. 6 that the performance of the proposed algorithm increases with the increase in number of features but does not improve much after the points marked with red-colored arrow. The best accuracies obtained are 97.75% (±2.31), 99.50% (±1.12), 92.40% (±2.17) and 60.5% (±9.65) for ORL, Yale, AR and LFW face databases, respectively. It is observed that the numbers of features to obtain these accuracies for respective face databases are 45, 30, 260 and 65, respectively (Fig. 6).

Figure 7 depicts the effect of number of features on the average number of converged fireflies. It is observed that the number of converged fireflies increases with the increase in number of features despite the fact that fireflies now move in much higher-dimensional space. Table 5 depicts performance evaluation of proposed FRBFNN with respect to number of features.

The size of each training image from ORL, Yale and LFW face databases is 92 × 112, i.e., 10,304. As we have used only 45, 30 and 65 features, respectively, dimensionality reduction achieved is 99.5, 99.7 and 99.4%, respectively. For AR face database, only 260 features from 4980 are taken

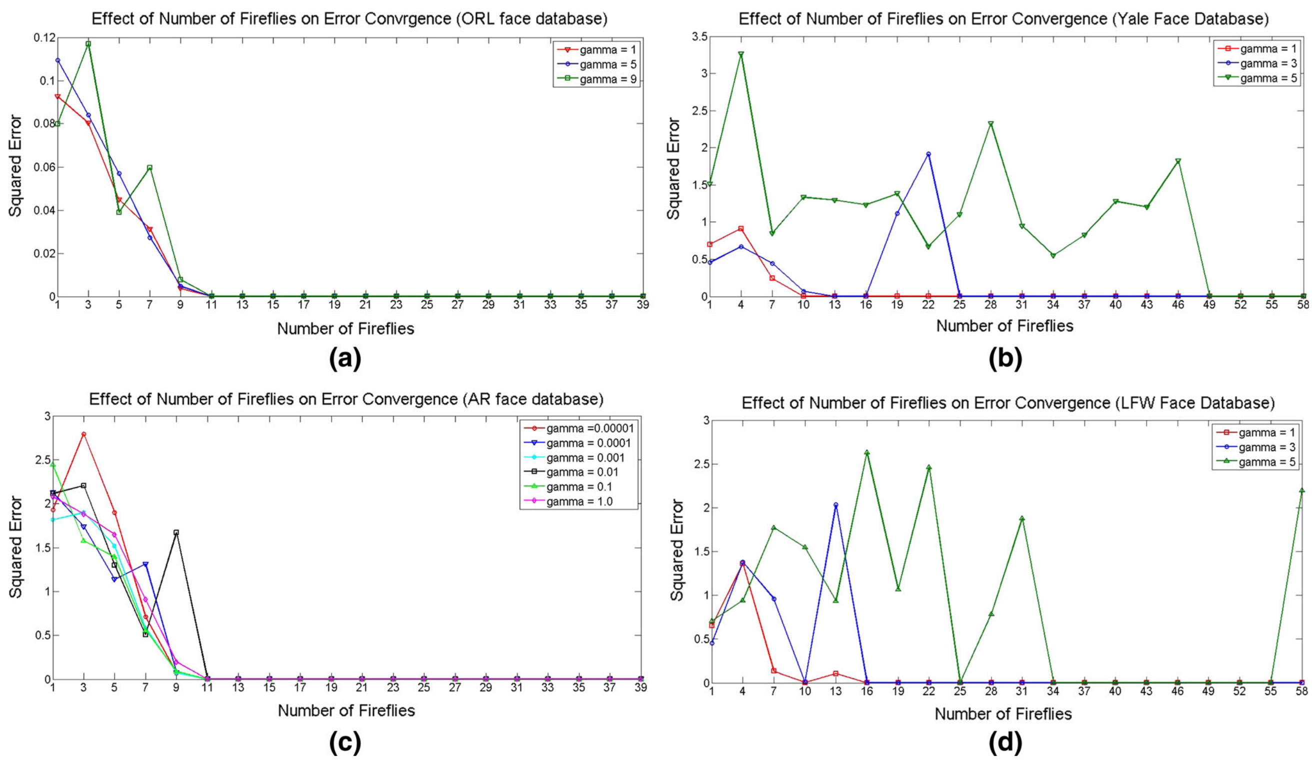

Fig. 5 Effect of number of fireflies on the algorithm convergence **a** ORL, **b** Yale, **c** AR and **d** LFW face databases

Table 4 Parameter selection

Database	Gamma (γ)	Minimum number of fireflies to converge (Q)	Minimum number of iterations for convergence (I)
ORL	5	11	7
Yale	3	13	9
AR	0.00001	11	6
LFW	1	16	8

resulting in dimensionality reduction of 94.8%. The low standard deviation for the face databases ORL, Yale and AR displays the strength of the proposed algorithm in terms of its stability across different independent runs.

7.2 Effect of number of training samples on recognition accuracy

The effect of number of training images on the average recognition accuracy and standard deviation was investigated for all four face databases, and the results are shown in Fig. 8. It is observed for ORL face database that the accuracy is maximum 98.50% (±2.11) using nine training images (Fig. 8a). The standard deviation with seven images is minimum among all observations for ORL face database. The maximum accuracy obtained for Yale face database is 100% (±0.00) with eight training images, and similar observation is achieved with nine training images (Fig. 8b). The trend visible in Fig. 8c for AR face database is a continuous increase in average recognition accuracy and a continuous decrease in the standard deviation. The most suitable number of training images for AR face database is 13 having average recognition accuracy as 92.15% (±1.23). LFW has huge variations and is still a challenging face database (Fig. 8d). The best accuracy obtained is 67.0% (±18.36) with 13 training images. The trend shown for all face databases is that the average recognition accuracy increases with the increase in number of training images used.

7.3 Effect of number of training images and sub-clustering on evolution of hidden neurons

The neurons in the hidden layer evolve based on the structure of the input feature space. The proposed algorithm captures variations available in the training face images and attempts to group subjectively similar training images of each person (Fig. 9). Figure 10a shows the trend of evolved hidden neurons averaged over the total number

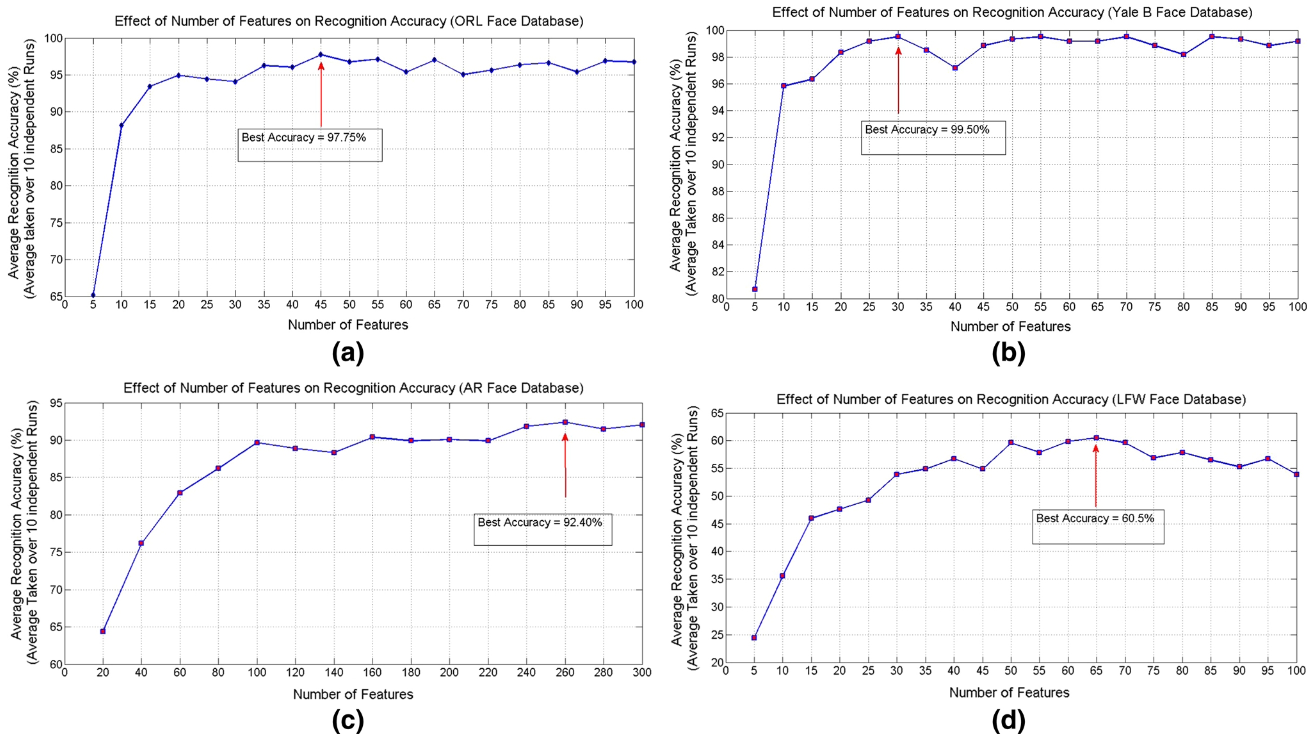

Fig. 6 Effect of feature dimension on average recognition accuracy using the proposed FRBFNN **a** ORL, **b** Yale, **c** AR and **d** LFW face databases

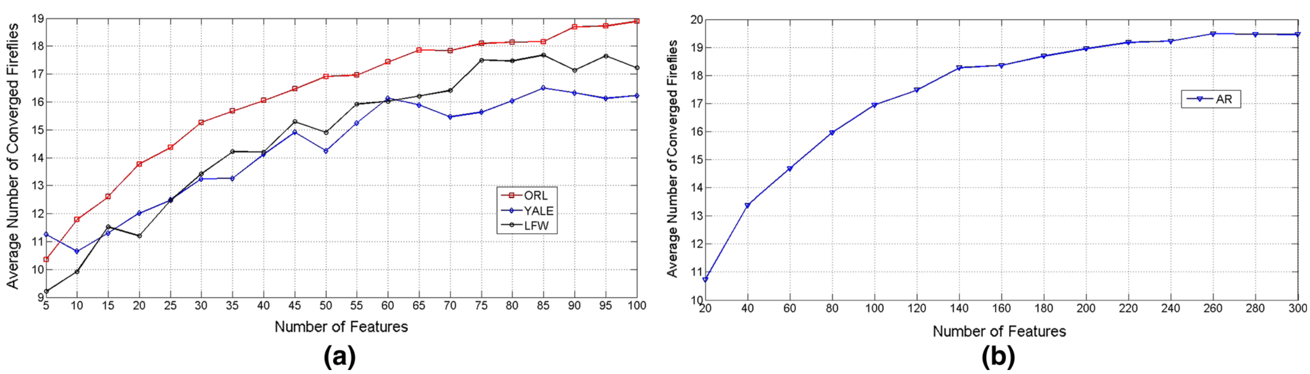

Fig. 7 Effect of feature dimension on average number of converged fireflies using proposed FRBFNN **a** ORL, Yale and LFW face databases, **b** AR face database

Table 5 Performance evaluation of the proposed FRBFNN algorithm with respect to the number of features

Face database	Number of features	Dimensionality reduction (%)	Average recognition accuracy (%) ± standard deviation	Average number of converged fireflies	Average number of iterations	Number of evolved neurons
ORL	45	99.5	97.75 ± 2.31	16.47	6.42	127
Yale	30	99.7	99.50 ± 1.12	13.24	12.37	53
AR	260	94.8	92.40 ± 2.17	19.49	6.48	294
LFW	65	99.4	60.50 ± 9.65	16.21	7.74	47

of persons in the databases, with respect to the number of training images. This is evident from Fig. 10a that more hidden neurons are generated for AR and LFW face databases having more variations, as compared to those for ORL and Yale, having less variations. If less number of sub-clusters (K) are taken, the proposed algorithm groups together face training images of a person involving two or more types of variations, for example, left and right

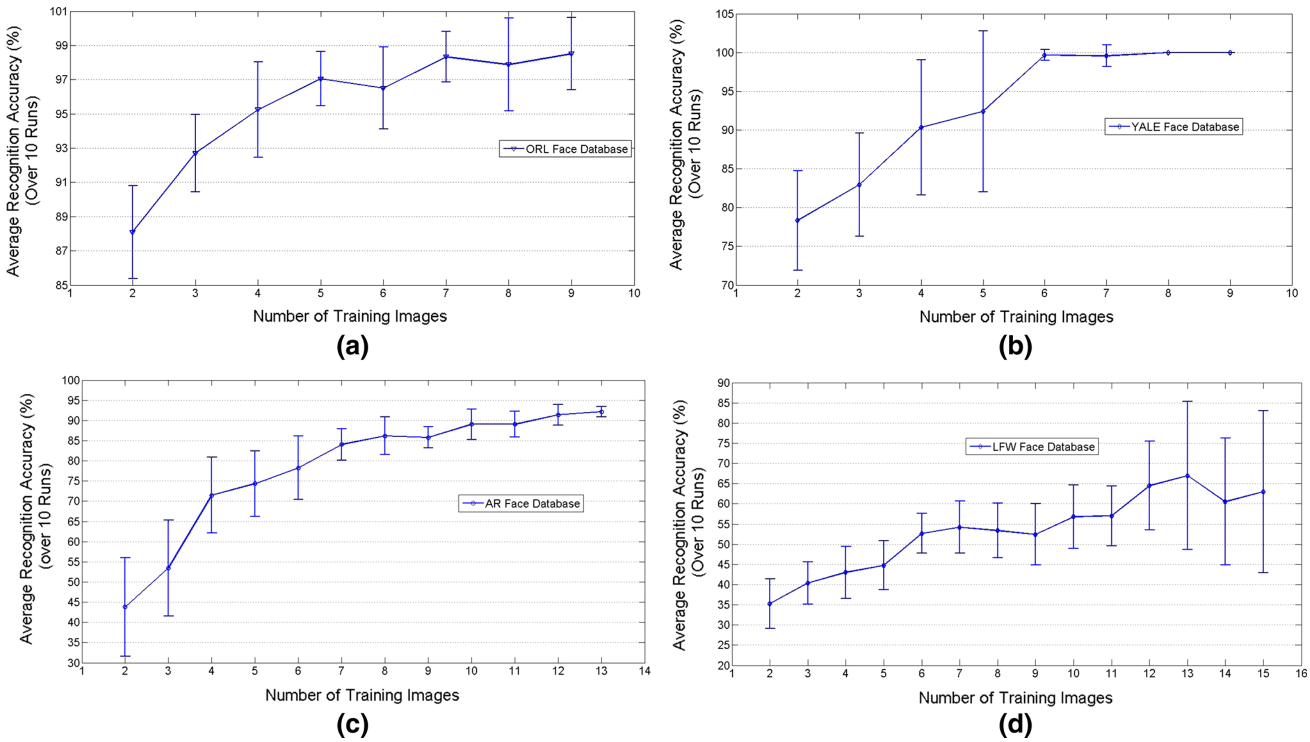

Fig. 8 Effect of number of training images on average recognition accuracy using the proposed FRBFNN (with standard deviation marked) **a** ORL, **b** Yale, **c** AR and **d** LFW face databases

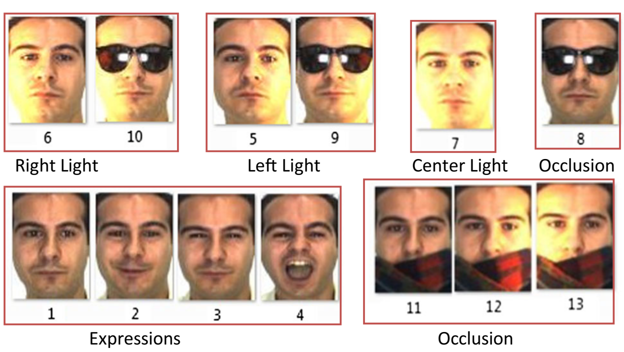

Fig. 9 Sub-clusters of a person class from AR face database forming different hidden neurons

illumination-based faces with goggles (AR face database) leading to misclassification. A trend of evolution of neurons with respect to number of sub-clusters is presented in Fig. 10b–d. The dotted line is drawn to highlight that the evolution of neurons is not linear with respect to the number of sub-clusters and slows down once the variations are captured by the algorithm.

7.4 Comparison with other face recognition methods

In this section, we compare the performance of the proposed FRBFNN technique with the results reported in the literature. The comparison is done with the existing techniques on RBFNN design for face recognition.

7.4.1 Performance comparison on ORL face database

The performance of the proposed FRBFNN technique is compared with the *k*-means clustering-based RBFNN [13] in terms of the standard deviation of recognition accuracies, obtained in 10 independent runs (Fig. 11a). The observations were made for number of features varying from 5 to 100. It is observed that the proposed FRBFNN technique deviates very less from the average recognition accuracy than that of *k*-means-based method. Also, the performance of the proposed FRBFNN in terms of average recognition accuracy is compared with the other non-evolutionary classifiers such as Euclidean (nearest neighbor) classifier, *k*-means clustering-based RBFNN and non-evolutionary RBFNN with 50 fixed centers (Fig. 11b). It is observed that the proposed FRBFNN technique outperforms the other three methods.

The performance of the proposed FRBFNN in terms of average recognition accuracy is also compared with five techniques, namely FHLA-based RBFNN [20], DCT + FLD-based RBFNN [17], point symmetry distance-based RBFNN [22], self-adaptive RBFNN [23] and polynomial RBFNN [21], and is shown in Table 6. The techniques are selected for comparison due to common

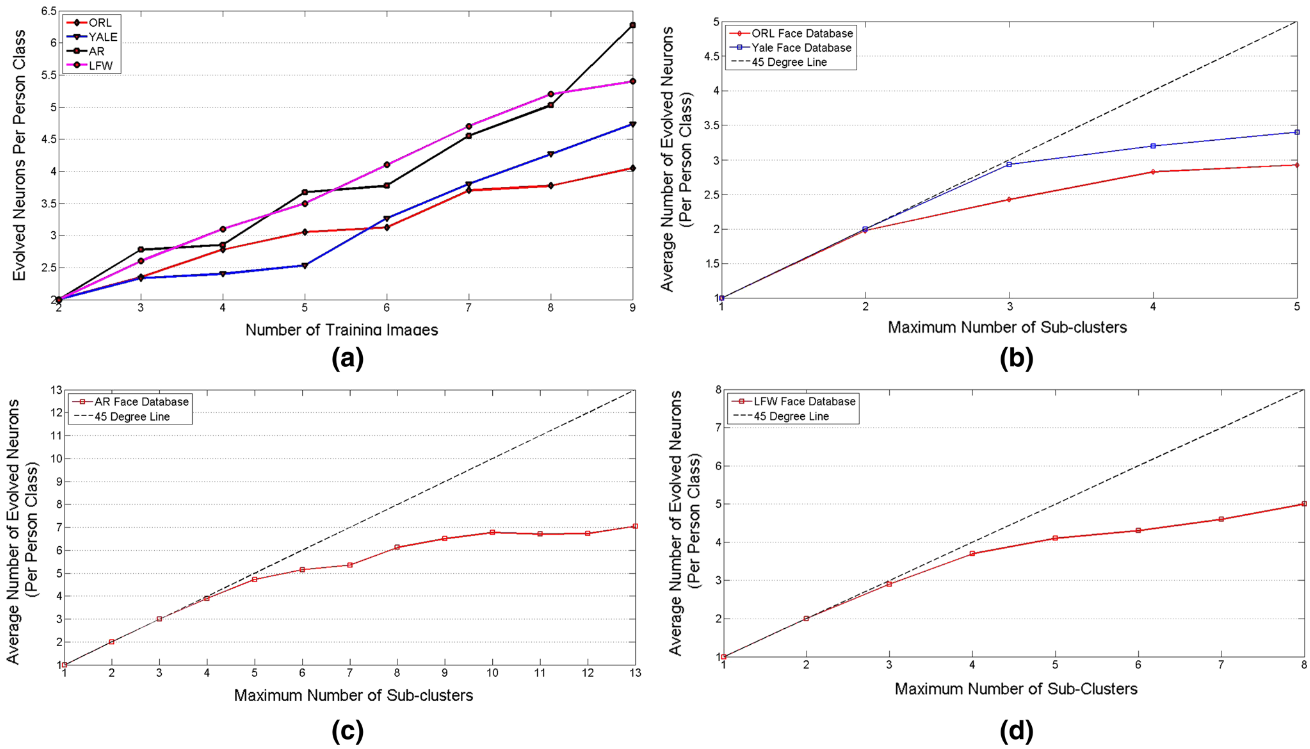

Fig. 10 Average number of evolved neurons **a** with respect to the increased number of training images, **b–d** with respect to the maximum number of sub-clusters

attributes available for comparison. The number of training images used in all studies is five, and the number of runs is ten except in polynomial RBFNN-based study where number of runs is five and in FHLA-based RBFNN where the number of runs is 40. The proposed FRBFNN uses only 20 iterations for algorithm convergence resulting in 97.75% accuracy in ORL face database as against 6500 and 15,000 iterations used in point symmetry distance-based RBFNN and self-adaptive RBFNN, respectively [22, 23]. The number of learning iterations is 100, while another 20 generations are used for differential evolution used for optimizing design parameters in polynomial RBFNN [21].

The number of neurons evolved using the proposed FRBFNN is less, as compared to 152 neurons evolved using self-adaptive RBFNN [23]. Point Symmetry-based study [22] uses three clusters per individual resulting in 120 fixed neurons, while the proposed FRBFNN technique does not fix the number of clusters. DCT + FLD-based RBFNN [17] used 55 features and reduced the dimensionality using FLD to obtain 30 features, while the proposed FRBFNN does not have similar overhead of dimensionality reduction. The number of iterations used by FHLA-based RBFNN [20] is 81, and its minimum average error rate is reported to be 0.45% resulting in average

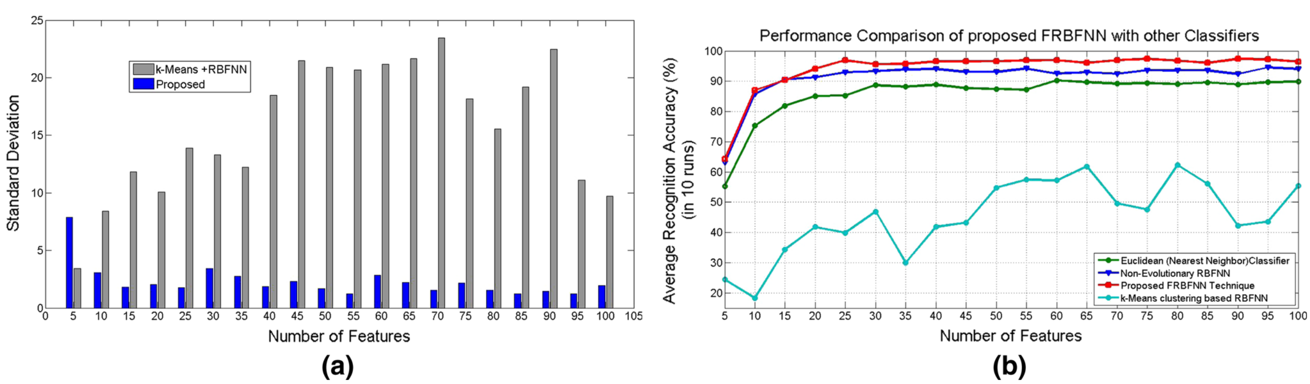

Fig. 11 Comparison of **a** standard deviations of the proposed FRBFNN and *k*-means based RBFNN with respect to varying number of features and **b** average recognition accuracy of FRBFNN with non-evolutionary classifiers

Table 6 Comparison of performance of the proposed FRBFNN algorithm with the existing techniques on ORL face database

Method	Year	Number of samples per person	Number of runs	Number of learning iterations/epochs	Number of neurons evolved/taken fixed	Number of features	Maximum average recognition accuracy (%) (ORL)
FHLA-based RBFNN [20]	2003	5	40	81	48–51	55	99.55
DCT + FLD-based RBFNN [17]	2005	5	10	–	Varied using parameter $1 \leq \alpha \leq 2$	30	97.55
Point symmetry distance-based RBFNN [22]	2007	5	10	6500	120-fixed	128	97.20
Self-adaptive RBFNN [23]	2009	5	10	15,000	152-evolved	64	97.30
Polynomial RBFNN [21]	2013	5	5	100/20	–	–	95.25
Proposed FRBNN technique	–	5	10	20	127-evolved	45	97.75 ± 2.31

recognition accuracy of 99.55% [20]. The proposed FRBFNN algorithm is fast as compared to other techniques and uses a very small number of features. The number of features used in this study is only 45 as against 55, 64 and 128 in [20, 22, 23], respectively.

The performance in terms of average number of iterations is also compared with the fuzzy hybrid learning algorithm (FHLA) [20] for varying number of features and is shown in Table 7. The proposed FRBFNN algorithm is run 40 times with randomly selected five training and five testing images at each run to compare average number of iterations (epochs). To simulate the conditions of FHLA, the number of features was taken as 10, 20, 30, 40, 50, 60, 70 and 80. It is observed that the proposed technique is faster than FHLA by 73.47 and 94.15% for 10 and 80 features, respectively.

7.4.2 *Performance comparison on Yale face database*

The performance of the proposed FRBFNN on Yale face database is compared with four techniques, namely FHLA [20], DCT + FLD-based RBFNN [17], IROLS-based RBFNN [19] and polynomial RBFNN [21], and is shown in Table 8. The proposed FRBFNN technique outperforms these techniques with maximum average recognition accuracy of 99.83% with a small standard deviation of 0.53 for Yale face database.

The performance of the proposed FRBFNN algorithm is also compared with FHLA technique [20] with respect to the average number of iterations required for convergence shown in Table 9. The number of fireflies is taken as 20, and the value of parameter γ is taken as 3. The number of features is varied from 10 to 80 to compare the speed of convergence with the FHLA technique. It is observed that the proposed FRBFNN algorithm outperforms FHLA in terms of speed of convergence. The average number of iterations in which the proposed algorithm converges does not increase with the increase in number of features. Therefore, the proposed FRBFNN algorithm is well suited to perform equally well in hyper-dimensional input spaces with large number of features at no extra cost.

Table 7 Comparison of average number of iterations with FHLA technique with the proposed FRBFNN technique on ORL face database

Number of features	Average number of iterations using FHLA [20]	Average number of iterations (proposed)	Speed increase (%)
10	23	6.10	73.47
20	34	6.29	81.50
30	46	6.38	86.13
40	59	6.40	89.15
50	67	6.47	90.34
60	92	6.43	93.01
70	101	6.52	93.54
80	112	6.55	94.15

Table 8 Comparison of performance of the proposed FRBFNN algorithm with the existing techniques on Yale face database

Method	Year	Number of samples per person	Number of runs	Number of learning iterations/epochs	Number of neurons evolved/taken fixed	Number of features	Maximum average recognition accuracy (%) (Yale)
FHLA-based RBFNN [20]	2003	6	40	67	20–23	55	99.75
DCT + FLD-based RBFNN [17]	2005	–	10	–	–	–	98.20
IROLS-based RBFNN [19]	2011	3	–	–	–	–	95.0
Polynomial RBFNN [21]	2013	6	5	100/20	–	–	95.60
Proposed FRBNN technique	–	6	10	20	53-evolved	30	99.83 ± 0.53

7.4.3 Comparison on AR face database

The proposed FRBFNN technique is compared with IROLS-based RBFNN [19] on AR face database. Using six training images per person, the average recognition accuracy of IROLS-based method is 75.5% (±5.7), while the average recognition accuracy for proposed FRBFNN technique is 78.25% (±7.84). Using all 13 training images, the proposed algorithm produces average recognition accuracy of 93.15% (±3.25) demonstrating better recognition accuracy and stability of the proposed algorithm. The proposed FRBFNN technique eliminates the limitation of the IROLS-based techniques in selecting the training images based on visual perception. The proposed FRBFNN can handle any variations in images used for training by grouping them based on the subjective similarity.

A common basis for comparison could not be found in the available literature on RBFNN design using LFW face database. The distinguishing strength of proposed FRBFNN algorithm thus is that it converges fast using small number of features to produce average recognition accuracy better than some of the existing techniques as is depicted in Tables 6, 7, 8 and 9.

7.5 Training and testing time

The training time for images of the face databases ORL, Yale, AR and LFW with 200, 90, 520 and 80 training images is 10.92, 9.23, 59.85 and 4.59 s, respectively, as shown in Table 10. The testing time to test one face image for the above databases is 0.0048, 0.0027, 0.0265 and 0.0022 s, respectively, which are considerably small and hence can be considered for real-time face recognition.

8 Conclusion

A new approach to center selection of the RBF units using firefly algorithm is proposed for face recognition. In the present study, the potential of firefly algorithm is investigated in RBFNN design for deciding number and centers of neurons. The feature selection overhead is negligible as only the upper left triangular DCT coefficients of the transformed training face image are used. The single face image-based training is not the focus of this study. A number of training images used for handling variations are effectively clustered in polynomial time using firefly

Table 9 Comparison of average number of iterations with FHLA technique with the proposed FRBFNN technique on Yale face database

Number of features	Average number of iterations using FHLA [20]	Average number of iterations (proposed)	Speed increase (%)
10	18	9.19	48.94
20	23	9.03	60.74
30	31	8.64	72.13
40	43	8.75	79.65
50	54	8.77	83.76
60	74	8.76	88.16
70	79	8.39	89.38
80	98	8.98	90.84

Table 10 Training and testing time

Face databases	Number of training images	Number of testing images	RBFNN hidden layer training time (s)	Testing time for each image (s)	Average recognition accuracy (%)± standard deviation (10 runs)
ORL	200	200	10.92	0.0048	97.75 ± 2.31
Yale	90	60	9.23	0.0027	99.83 ± 0.53
AR	520	200	59.85	0.0265	93.15 ± 3.25
LFW	80	80	4.59	0.0022	60.50 ± 9.65

algorithm, and the hidden layer neurons evolve automatically based on the structure of the data. A detailed discussion on parameters selection has been presented. The algorithm convergence with respect to the value of γ and number of fireflies has been investigated. The effect of γ on the recognition accuracy has also been investigated. The proposed algorithm outperforms various existing techniques reported in the literature. The strength of the proposed FRBFNN algorithm lies in its capability to perform well with a very small number of features along with fast training and testing. We conclude that the proposed FRBFNN technique is efficiently capable of handling variations in face images in real time. The limitation of the proposed technique lies in its complexity of obtaining algorithm parameters for a face database different from those used in this study. The automatic tuning of parameters is a difficult and computationally intensive task as it has been reported in the literature and remains a challenge for real-time face recognition. Also, it is recommended to extend the proposed work for more complex face databases including incomplete or partial faces.

Acknowledgements This work is part of the Ph.D. research carried out at Birla Institute of Technology and Science, Pilani. We are grateful to the reviewers of this article for their constructive and valuable review comments which helped us to improve this article in many different ways.

Compliance with ethical standards

Conflict of interest Birla Institute of Technology and Science, Pilani, is the employer for both the authors and has provided all necessary support to carry out the research work presented in this paper. There is no external funding received. The authors declare that they have no conflict of interest.

Informed consent There is no direct involvement of the authors with human participants whose face images were used in the present study. The face databases ORL, Yale, AR and LFW are the benchmarked face databases used by the researchers all across the world. The permission to download AR face database was obtained from Prof. Aleix M. Martinez [44, 45]. The databases ORL, Yale and LFW are available free online [42, 43, 46, 47].

References

1. Zhang D, Zuo W (2007) Computational intelligence based biometric technologies. IEEE Comput Intell Mag 2(2):26–36
2. Chellappa R, Sinha P, Phillips PJ (2010) Face recognition by computers and humans. Computer 43(2):46–55
3. Er MJ, Wu S, Lu J, Toh HL (2002) Face recognition with radial basis function (RBF) neural networks. IEEE Trans Neural Netw 13(3):697–710
4. Li Z, Park U, Jain AK (2011) A discriminative model for age invariant face recognition. IEEE Trans Inf Forensics Secur 6(3):1028–1037
5. Ho HT, Chellappa R (2013) Pose-invariant face recognition using Markov random fields. IEEE Trans Image Process 22(4):1573–1584
6. De Marsico M, Nappi M, Riccio D, Wechsler H (2013) Robust face recognition for uncontrolled pose and illumination changes. IEEE Trans Syst Man Cybern Syst 43(1):149–163
7. Turk MA, Pentland AP (1991) Face recognition using eigenfaces. In: Proceedings CVPR '91, IEEE computer society conference on computer vision and pattern recognition, pp 586–591
8. Brunelli R, Poggio T (1993) Face recognition: features versus templates. IEEE Trans Pattern Anal Mach Intell 15(10):1042–1052
9. Wiskott L, Fellous J-M, Kuiger N, von der Malsburg C (1997) Face recognition by elastic bunch graph matching. IEEE Trans Pattern Anal Mach Intell 19(7):775–779
10. Zhang J, Yan Y, Lades M (1997) Face recognition: eigenface, elastic matching, and neural nets. Proc IEEE 85(9):1423–1435
11. Belhumeur PN, Hespanha JP, Kriegman D (1997) Eigenfaces vs. fisherfaces: recognition using class specific linear projection. IEEE Trans Pattern Anal Mach Intell 19(7):711–720
12. Poggio T, Girosi F (1990) Networks for approximation and learning. Proc IEEE 78(9):1481–1497
13. Bishop C (1995) Neural networks for pattern recognition, Chapter 5. Oxford University Press, Oxford, pp 164–193
14. Zhang GP (2000) Neural networks for classification: a survey. IEEE Trans Syst Man Cybern Part C Appl Rev 30(4):451–462
15. Lawrence S, Giles CL, Tsoi AC, Back AD (1997) Face recognition: a convolutional neural-network approach. IEEE Trans Neural Netw 8(1):98–113
16. Boughrara H, Chtourou M, Benmar C, Chen L (2014) Face recognition based on perceived facial images and multilayer perceptron neural network using constructive training algorithm. Comput Vis IET 8(6):729–739
17. Er MJ, Chen W, Wu S (2005) High-speed face recognition based on discrete cosine transform and RBF neural networks. IEEE Trans Neural Netw 16(3):679–691
18. Jing X-Y, Yao Y-F, Yang J-Y, Zhang D (2008) A novel face recognition approach based on kernel discriminative common vectors (KDCV) feature extraction and RBF neural network. Neurocomputing 71(13–15):3044–3048
19. Wong YW, Seng KP, Ang L-M (2011) Radial basis function neural network with incremental learning for face recognition. IEEE Trans Syst Man Cybern B Cybern 41(4):940–949
20. Haddadnia J, Faez K, Ahmadi M (2003) A fuzzy hybrid learning algorithm for radial basis function neural network with application in human face recognition. Pattern Recognit 36(5): 1187–1202

21. Oh S-K, Yoo S-H, Pedrycz W (2013) Design of face recognition algorithm using PCA-LDA combined for hybrid data pre-processing and polynomial-based RBF neural networks: design and its application. Expert Syst Appl 40(5):1451–1466
22. Sing JK, Basu DK, Nasipuri M, Kundu M (2007) Face recognition using point symmetry distance-based RBF network. Appl Soft Comput 7(1):58–70
23. Sing JK, Thakur S, Basu DK, Nasipuri M, Kundu M (2009) High speed face recognition using self adaptive radial basis function neural networks. Neural Comput Appl 18:979–990
24. Izakian H, Abraham A (2011) Fuzzy C-means and fuzzy swarm for fuzzy clustering problem. Expert Syst Appl 38(3):1835–1838
25. Wang D, Zeng X-J, Keane JA (2012) A clustering algorithm for radial basis function neural network initialization. Neurocomputing 77(1):144–155
26. Tsekouras GE, Tsimikas J (2013) On training RBF neural networks using input–output fuzzy clustering and particle swarm optimization. Fuzzy Sets Syst 221:65–89
27. Hruschka ER, Campello RJGB, Freitas AA, de Carvalho ACPLF (2009) A survey of evolutionary algorithms for clustering. IEEE Trans Syst Man Cybern Part C Appl Rev 39(2):133–155
28. Oh S-K, Kim W-D, Pedrycz W, Seo K (2014) Fuzzy radial basis function neural networks with information granulation and its parallel genetic optimization. Fuzzy Sets Syst 237:96–117
29. Feng Y, Wu Z, Zhong J, Ye C, Wu K (2010) An enhanced swarm intelligence clustering-based RBFNN classifier and its application in deep Web sources classification. Front Comput Sci China 4(4):560–570
30. Oh S-K, Kim W-D, Pedrycz W, Joo S-C (2012) Design of K-means clustering-based polynomial radial basis function neural networks (pRBF NNs) realized with the aid of particle swarm optimization and differential evolution. Neurocomputing 78(1):121–132
31. Feng H-M (2006) Self-generation RBFNs using evolutional PSO learning. Neurocomputing 70(1–3):241–251
32. 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
33. Senthilnath J, Omkar SN, Mani V (2011) Clustering using firefly algorithm: performance study. Swarm Evolut Comput 1(3):164–171
34. Yang X-S (2008) Nature-inspired metaheuristic algorithms, Chapter 10. Luniver Press
35. Senapati MR, Dash PK (2013) Local linear wavelet neural network based breast tumor classification using firefly algorithm. Neural Comput Appl 22(7–8):1591–1598
36. Gao M-L, He X-H, Luo D-S, Jiang J, Teng Q-Z (2013) Object tracking using firefly algorithm. IET Comput Vis 7(4):227–237
37. Yu C, Jin B, Lu Y, Chen X, Yi Z, Zhang K, Wang S (2013) Multi-threshold image segmentation based on firefly algorithm. In: Ninth international conference on intelligent information hiding and multimedia signal processing, pp 415–419
38. Yang XS (2014) Swarm intelligence based algorithms: a critical analysis. Evolut Intell 7(1):17–28
39. Fister I, Fister I Jr, Yang X-S, Brest J (2013) A comprehensive review of firefly algorithms. Swarm Evolut Comput 13:34–46
40. Yang X-S, Deb S, Loomes M, Karamanoglu M (2013) A framework for self-tuning optimization algorithm. Neural Comput Appl 23(7–8):2051–2057
41. Dabbaghchian S, Ghaemmaghami MP, Aghagolzadeh A (2010) Feature extraction using discrete cosine transform and discrimination power analysis with a face recognition technology. Pattern Recognit 43(4):1431–1440
42. ORL Face Database, http://www.cl.cam.ac.uk/Research/DTG/attarchive/pub/data/att_faces.zip. AT&T Laboratories Cambridge, UK
43. Yale Face Database, http://vision.ucsd.edu/datasets/yale_face_dataset_original/yalefaces.zip
44. Martinez AM, Benavente R (1998) The AR face database. CVC Technical Report #24
45. Martinez AM, Kak AC (2001) PCA versus LDA. IEEE Trans Pattern Anal Mach Intell 23(2):228–233
46. LFW Face Database, http://vis-www.cs.umass.edu/lfw/
47. Huang GB, Ramesh M, Berg T, Learned-Miller E (2007) Labeled faces in the wild: a database for studying face recognition in unconstrained environments. University of Massachusetts, Amherst, Technical Report 07-49