Computational Intelligence Based Face Recognition

THESIS

Submitted in partial fulfillment
of the requirements for the degree of
DOCTOR OF PHILOSOPHY

by

Vandana Agarwal

Under the Supervision of

Prof. Surekha Bhanot

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI
2015

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

CERTIFICATE

This is to certify that the thesis entitled "**Computational Intelligence based Face Recognition**" and submitted by **Vandana Agarwal**, ID No 2008PHXF407P, for award of Ph.D. of the Institute embodies original work done by her under my supervision.

Signature of the Supervisor
Name in capital letters : SUREKHA BHANOT
Designation :Professor
Date: December 18, 2015

Acknowledgements

I consider myself fortunate to have with me many people who have been extremely helpful in supporting me in this journey of Ph.D. research. First and foremost, I am thankful to my supervisor Prof. Surekha Bhanot without whose support and guidance I would never have been able to complete my research successfully. She has always been a great advisor and mentor for me over these years. I am thankful to her for her invaluable suggestions and ideas given to me from time to time which shaped up the overall work presented in this thesis. I am indebted to her for her concerns and efforts. She would always go an extra mile to help me professionally, academically or personally. I admire her immensely as a teacher, advisor and as a person as well.

I am thankful to Prof. V.S. Rao, Acting Vice Chancellor, BITS Pilani and Prof. Ashoke Kumar Sarkar, Director, BITS Pilani, Pilani campus for giving me an opportunity to pursue my Ph.D. research with financial support. I am thankful to Prof. Sanjay Kumar Verma, Dean, and Prof. Hemant R. Jadhav, Associate Dean, Academic Research Division (Ph.D. Programme) for providing necessary guidelines and extending full support, which were very important for the successful completion of this thesis. I would like to thank Prof. S.C. Sivasubramanian, Dean, Administration and Prof. J.P. Misra, Unit Chief, Information Processing Centre for their encouragement and support during the entire duration of this research.

I am also thankful to my Doctoral Advisory Committee (DAC) members Prof. Raj Kumar Gupta, Professor of Physics, and Prof. Sundaresan Raman, Professor of Computer Science for evaluating my work and giving me their valuable comments and suggestions. Thanks are due to all unknown reviewers of various conferences and journals for reviewing our research papers and providing their valuable comments and suggestions for improvement.

All colleagues at the Department of Computer Science and Information Systems have been helpful and always provided support. I am thankful to the Head of the Department, Prof. Rahul Banerjee for all his support in granting funds for attending conferences and his constant encouragement throughout. I would like to thank Prof. Sudeept Mohan, Convener, Doctoral Research Committee (DRC), and all members of the DRC for their constant support and

guidance. I have learnt many things from my senior professors Prof. J.P. Misra, Prof. Sundar Balasubramaniam, Prof. Sudeept Mohan, Prof. Rahul Banerjee, Prof. Navneet Goyal and Prof. Poonam Goyal whose valuable inputs given during the departmental seminars or otherwise were very useful and helped me in analyzing the results more intensely. I am also thankful to all faculty colleagues at the Department of Computer Science and Information Systems, in particular, and at BITS, Pilani, especially Dr. Parikshit Singh from the Department of Electrical, Electronics and Instrumentation Engineering, for their valuable inputs and support. I am also thankful to Mr. Sanwarmal for his administrative support throughout.

My friends and colleagues, Mayuri, Avinash and Asma have always been there to help me with all technical discussions and moral support in difficult situations professionally or personally. Mayuri has been there with me since the beginning of my research when we used to work in the lab popularly known as Cisco Lab. It was the time when I was struggling with building the initial framework for my research. She encouraged me in many difficult times and worked out solutions to make them working for me, critically analyzed the situations and helped me always. I would like to thank my friends Vandita Srivastava and Padma Singh who are not my colleagues but have played a great role in my life. Vandita, working as a Scientist, and Padma, a Human Rights Activist, are long time friends and have been with me in every phase of my life. Both the friends are the true well wishers, and have been there to empower me at times of need. Both of them constantly encouraged and motivated me to complete my research. I would also like to thank Dr. Monica Sharma for her help and support.

Last, but not the least, I am thankful to the great mother Nature for blessing me with Shivam and Gauransh, my loving sons. Both of them gave me unconditional love and were always there with me in all ups and downs of my life. For me, life has been beautiful in working hard for Shivam and Gauransh. This research is a reality, only because my sons sacrificed their comforts and they always wished their mother to do the research comfortably.

Vandana

To My Loving Children,

Gauransh and Shivam

ABSTRACT

Computational Intelligence based Face Recognition

By

Vandana Agarwal

Machine based face recognition is a challenging pattern recognition problem and has been researched for over two decades. The challenge lies in efficiently handling variations in pose, illumination, expression and occlusion. Conventional techniques have been found to have limitations in handling above variations, while a set of algorithms known as Computational Intelligence offers better solutions to handle complex real world problems. Computational Intelligence is a set of nature inspired algorithms which consists of Artificial Neural Networks, Evolutionary Algorithms and Fuzzy Computing of which we have explored the potential of the first two categories in the present research. In this thesis, an evolutionary approach inspired by the natural fireflies is used and its potential in handling face recognition problem is explored. Research work is mainly focused on the Firefly Algorithm and the Radial Basis Function Neural Network for improved face recognition.

First, the problem of feature selection is addressed using Firefly Algorithm and a novel algorithm is proposed. The proposed algorithm is analyzed for its convergence and parameter values of the algorithm are obtained for the benchmarked face databases namely ORL, Yale, AR and LFW. The effect of light absorption coefficient γ is investigated on algorithm convergence, average recognition accuracy and dimensionality reduction. The proposed technique is compared with the feature selection methods based on Particle Swarm Optimization and Genetic Algorithm. The proposed technique outperforms some of the existing methods.

Second, an evolutionary firefly inspired algorithm for designing hidden layer of Radial Basis Function Neural Network (RBFNN) for improved face recognition is proposed. The proposed technique uses Firefly Algorithm to obtain natural sub-clusters of training face images formed around variations in pose, illumination, expression and occlusion. Movement of fireflies in a hyper-dimensional input space is controlled by tuning the parameter Gamma (γ) of the Firefly Algorithm which plays an important role in 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 the Radial Basis Function Neural Network (RBFNN) in adaptive evolution of number and centers of hidden neurons. The strength of the proposed technique lies is its fast convergence, improved face recognition performance, reduced feature selection overhead and the algorithm stability.

Third, the effect of the basis function's shape on face recognition performance of Radial Basis Function Neural Network is investigated and a shape estimation technique, based on the overlapping of the neighboring basis functions, is proposed. The shape of basis function for sub-cluster of each class is controlled by an overlapping factor α. The amount of spread is proportional to the radius of the sub-cluster and the distance of the nearest sub-cluster belonging to a different class.

The integrated classifier employing optimal number and centers of hidden layer neurons, and the shape of basis functions thus obtained is then evaluated for its performance using Sensitivity Analysis, Confusion Matrix and Receiver Operating Characteristics (ROC) Curve.

Table of Contents

List of Figures

(Contd.)

List of Tables

List of Abbreviations

DCT	Discrete Cosine Transform
NN	Neural Network
RBFNN	Radial Basis Function Neural Network
MLP	Multi Layer Perceptron
FRBFNN	Firefly Inspired Radial Basis Function Neural Network
OLAF	Overlapping Factor
DWT	Discrete Wavelet Transform
FR	Face Recognition
PCA	Principal Component Analysis
FA	Firefly Algorithm
PSO	Particle Swarm Optimization
GA	Genetic Algorithm
FS	Feature Selection
ORL	Olivetti Research Lab
FIFS	Firefly Inspired Feature Selection
ROC	Receiver Operating Characteristics
CM	Confusion Matrix
K-NN	k-Nearest Neighbor
LDA	Linear Discriminant Analysis
SVM	Support Vector Machine
ANN	Artificial Neural Network
ABC	Artificial Bee Colony
ACO	Ant Colony Optimization
CI	Computational Intelligence
FC	Fuzzy Computing
LFW	Labeled Faces in the Wild
ICA	Independent Component Analysis
EA	Evolutionary Algorithm

EBGM	Elastic Bunch Graph Method
LBP	Local Binary Pattern
p-NN	Polynomial Neural Network
PCNN	Pulse Coupled Neural Network
BP	Back Propagation
FLA	Fisher Linear Discriminant Analysis
SOM	Self Organizing Maps
GSA	Gravitational Search Algorithm
KDCV	Kernel Discriminant Common Vector
FHLA	Fuzzy Hybrid Learning Algorithm
FCM	Fuzzy C-Means
EM	Expectation Maximization
NP	Nondeterministic Polynomial
SORBF	Self Organizing Radial Basis Function
ROLSA	Recursive Orthogonal Least Square Algorithm
PDE	Partial Differential Equations

Chapter 1

Introduction

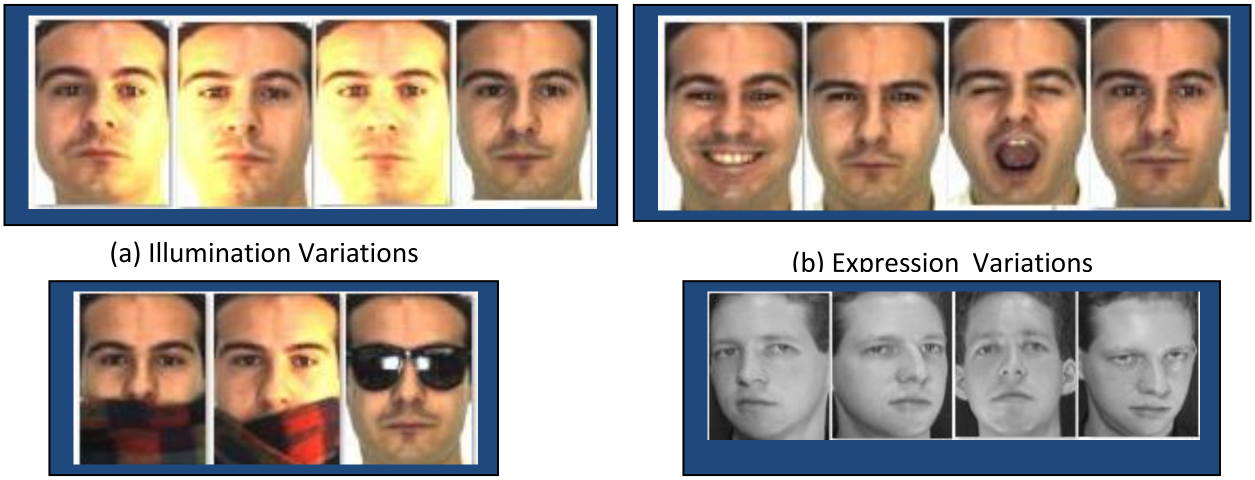
1.1. Introduction

Humans possess a very important trait of recognizing people by their faces. Faces of all humans appear similar in construction with two eyes, one nose, one mouth, two cheeks, a forehead, a chin and two eyebrows. Despite similarities of these geometrical features, all persons have difference in their faces and are recognized by fellow human beings without any difficulty. The faces of persons are different due to a large number of minute unique features specific to their genetic structure, region of their origin and the construction of the bones and muscles of their faces. Thousands of minor differences between two persons' faces are processed by human brain. The neurons connected to human eye transmit signals and information to the brain, which it processes and recognizes people based on past acquaintances. The specific portion of the brain, called as *fusiform gyrus* is responsible for face recognition. An injury in this portion of the brain may bring down the ability of humans to recognize faces resulting in an inability known as *prosopagnosia*.

These days, globally there have been situations arising from terror threats and it has become necessary to authenticate person's identity automatically through surveillance cameras mounted at airports and other public places. Machine based face recognition is needed to relieve humans from repetitive and voluminous task of face recognition. In today's era of smart phones, smart homes and smart offices, machine based face recognition is needed to recognize authorized person's face. Any unauthorized face recognized by the machine, may trigger alarm for necessary actions. Passwords for system authorization are based on face and other biometric traits such as finger print, iris scan etc. The development of face recognition technology has contributed significantly in growth of internet banking. Face recognition research finds challenges in handling variations in illumination, pose, expression or disguise etc. and demands robust algorithms for unconstrained face recognition. The added face recognition step in future could be to add blinking of eyes, frowning of eyebrows or smiling in the face recognition process where it will ensure that no impostor can misuse the technology. The face recognition technology has been adopted by many countries for various purposes such as availing schemes and subsidies etc. In India, every citizen is issued a unique identification card known as AADHAR which is used as a proof of identity. It is a 12 digit unique number generated for an individual based on his or her face, finger prints and iris biometric traits.

Person identification and verification have been automated using various human traits known as biometrics which are of two types - physiological and behavioral. Physiological biometrics are finger print, iris scan, palm scan, face and speech. Behavioral biometrics include gait, handwriting and signature, and speech. The key advantage of the face biometric over the other biometric traits is that the face images and videos of persons can be captured without the cooperation of the participants as in airports or other public places, while other biometric data for surveillance cannot be captured without persons' cooperation.

The major challenges in face recognition have been in handling variations in illumination, pose, expression and occlusion etc. where the test images differ from those of the training images [Fig.1.1]. Despite conventional research in face recognition for almost three decades, handling such variations to the fullest is still a challenge. *Computational intelligence* based face recognition techniques offer better solutions as compared to the traditional techniques. Computational Intelligence is a set of nature inspired computational methodologies and approaches to handle complex real world problems in more efficient ways compared to the conventional approaches. These include Artificial Neural Networks (ANN), Fuzzy logic, evolutionary algorithms such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Artificial Bee Colony(ABC), Firefly Algorithm (FA) etc. These techniques are inspired by human nervous system, human gene selection, flocking behavior of birds, ants, bees and flashing behavior of firefly etc. Computational intelligence techniques are adaptive and can learn from the data itself, if learning parameters are selected properly. The evolutionary algorithms can handle various face recognition tasks such as feature selection, clustering, subset selection, parameter optimization etc. Most of these algorithms use a fitness function and work well for different applications, and face recognition in particular, if problem is represented as a search problem.



(a) Illumination Variations

(b) Expression Variations

(c) Occlusion and Disguise

(d) Pose Variations

Fig. 1.1. Challenges in Face Recognition in handling variations due to (a) Illumination (b) Expression (c) Occlusion and Disguise and (d) Pose *[(a)-(c) Images from AR Face database and (d) images from ORL face database]*

Face recognition is posed as a classification problem where human faces are represented using appropriate feature extraction techniques and the most discriminative features are selected using feature selection techniques. The face recognition system consists of two stages - training and testing. Similar to human brain, where human brain is trained to recognize persons by eyes, nose, mouth, lips, shape of face, jaw line etc., and their relative associations and distances amongst each other. The machine based face recognition system is trained with features extracted using information theoretic approaches such as Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT). The mathematical model thus generated using the training data is used to test a person's unseen image.

The algorithms developed for face recognition rely on the discriminative power of face features so as to uniquely define a person's face. The recognition process uses training features and constructs a framework to be used with the similar set of features of the test images. The most commonly used recognition algorithms are k-Nearest Neighbor classifier (K-NN), Linear Discriminative Analysis (LDA), Elastic Bunch Graph Method (EBGM), Support Vector Machines (SVM) and Artificial Neural Network (ANN) etc.

1.2. Face Representation Techniques

In machine based faced recognition processes, face images are acquired by a digital camera which captures the light reflected from the surface of face. The light falling on the three dimensional face of a person gets reflected in unique directions due to unique structure of facial bones and muscles of each person. The amount of light captured by the detectors at the camera is quantized into gray values. A face image looks like a two dimensional array of quantized intensity values. A machine has a challenge to understand the *pattern* or arrangement of such gray values for each individual. As each individual has a unique combination of shapes and sizes of geometrical features, the corresponding image also reflects structural properties, unique to the person's face. The intuitive features of human face such as eyes, nose, mouth, jaw line, shape of face etc. are well understood and recognized by humans, but are not robust to be handled by a machine. Most of the today's face recognition techniques therefore use features extracted using information theoretic approaches such as DCT and DWT. All features that are extracted may or may not form a unique pattern. There is a need for *selecting features* that are capable of distinguishing people and form unique patterns for an individual person. These selected features form patterns representing the face images. The problem of face recognition thus reduces to a *pattern recognition* problem.

Early face recognition techniques relied on *geometric features* such as center points of the two eyes, distance between two center points of the two eyes, length of the nose, angle between the base line. The *conventional approaches* to face recognition are of two types - *holistic and local*. The *holistic* methods use the complete face information to create the subspace using various feature extraction methods such as Principal Component Analysis (PCA) based Eigen face methods, Linear Discriminant Analysis (LDA) based Fisher Face methods and Independent Component Analysis (ICA) etc. Face images are then projected to low dimensional space to reduce dimensionality. Face representation techniques such as *Eigen Face* and 'Fisher Face' are based on *Principal component analysis (PCA).* The PCA technique attempts to find the directions of maximum variance in the data and data is projected along the axes of maximum variances. The new values of data in these directions are called *principal components*. The representation of the face images uses the features called as Eigen vectors, and the transformed face image is known as *Eigen Face*. The limitation of the Eigen face approach to face recognition is that it is sensitive to illumination differences and performs well in constrained environment with almost nil illumination variations. These approaches attempt to maximize the between-class scatter or maximize the ratio of the between-class scatter and the within-class scatter. *Local* methods compare the local regional statistics of the face images such as Elastic Bunch Graph Method (EBGM) which uses a graph representation of set of facial components or features and the Local Binary Pattern (LBP) which divides the face area into two windows and uses chi-square statistics to compare the LBP histograms for face recognition. These methods are sensitive to illumination variations and also require a complete face database to develop the model and cannot implement incremental or dynamically changing face database.

Transform based feature extraction methods such as *Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT) etc.* represent a face through their coefficients. These transforms possess high information packing ability due to which a face can be represented efficiently in terms of small number of coefficients of the transform thereby reducing the computational complexity of the Face Recognition system. Pixels in the original image are highly correlated and these transforms remove the spatial redundancy between the neighboring pixels. Efficiency of a transformation scheme depends on its ability to pack maximum information content in as few coefficients as possible. The utility of high frequency coefficients is less in face recognition as most of the useful information in a human face lies in cheek region of the face, or the details around eyes and mouth. Most of the transform based techniques use low frequency coefficients for face representation. Usefulness and energy compaction ability of any transform can also be assessed by reconstructing the images from only few most informative coefficients of their transformed counterparts. The DCT method captures the information in frequency domain while the wavelets capture the information in both time and frequency domain. Wavelets can be used to perform multiresolution analysis of the face images so that different details at various levels of resolution - coarse or fine, can be used for face recognition more accurately. Some of commonly used Wavelets are Haar, Daubechies, curvelet, Gabor etc.

1.3. Feature Selection Techniques

Feature selection phase of the face recognition process attempts to obtain the most discriminative features that can discriminate between two or more person's faces and captures variations in illumination, pose, expression or occlusion. *Curse of Dimensionality* is caused by the large number of features; many among them are not so useful features and cause *over fitting* of the face data resulting in reduced performance of the face recognition system. Therefore, it is important to select features to reduce the dimensionality of the feature vector as well as to have features with maximum discriminative powers. The feature selection process enables more accurate face recognition and reduces computational complexity of the face recognition process. *Feature selection* is a combinatorial problem and the exhaustive Brute Force search for the best combination of relevant and non-redundant features is exponential in time. If the total number of extracted features is *n* and if *k* features are to be selected from *n* features using brute force method, where k << n, then the time complexity is of order $O(2^n)$, which is *exponential*. For example, if the number of initially extracted features is 1000, then the time to select some important features will be of order of 10^{301} units of time and computations may go on for millions of years, which is practically impossible. The feature selection problem is viewed as a combinatorial problem and therefore can be considered to be *NP Hard* problem. The optimal solution approximately close to the best combination of features is achievable in *polynomial time* as established by past research *(Hruschka et al, 2009)*. The challenge is in evolving a technique that selects features closest to the best features and the number of selected features is very small.

The feature selection problem has been addressed using two approaches - Filter approach and Wrapper approach. *Filter* approach based methods are computationally cheap and apply statistical analysis for ranking of features based on utility criteria, e.g. energy probability or entropy is used as a criteria for selecting image features for face recognition. *Wrapper* approach based methods rely on the data and evaluate selected features on the basis of their predictive power.

The feature selection problem is also represented as a search problem in hyper dimensional space and a heuristic is used for searching the best combination based on a fitness function. The most preferred approach today is wrapper based while researchers also use evolutionary algorithms such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO) etc. These techniques are also known as meta-heuristic algorithms as they offer variety of search heuristics. Some techniques such as GA, take a large number of iterations to converge to the best features while others such as PSO, ACO converge faster than GA. These algorithms are efficient and reach the near optimal solution in polynomial time, but are also likely to get trapped in local optima and produce a feature set which may not give the best classification accuracy.

1.4. Face Recognition as a Classification Problem

Classification is a process which predicts class labels for a test data based on the training of the classifier. Each class in face recognition process has a unique label corresponding to each person, which can be a unique identifier or person's name. Classification algorithms attempt to learn from data for predicting the class label for an unseen face image. The simplest classifier for any application is the rule based classifier, which based on the validity of the conditions of the rule attempt to classify test data. Other classifier algorithms are k-Nearest Neighbor (k-NN), Decision trees, Naive Bayes Classifier, Support Vector Machines (SVM) and Artificial Neural Networks (ANN) etc.

The *k-NN classifiers* classify the test data based on its distance from the cluster centers. These classifiers may not perform well if data is non-linearly separable as in face recognition problem. If the decision boundary is obtained carefully, the test feature vector can be classified with better accuracy. *Decision Trees* are graph like structures in which the internal nodes represent the test performed on the test data to reach any leaf node which associates class label (decision) to the test data. This technique suffers from a drawback that it cannot predict correctly the class label for the test feature vector, if the test data deviates from the values of training data. Also construction of the optimal decision tree is a complex combinatorial problem. *Support Vector Machines* (SVM) are the classifier models which define the decision boundary as lines or planes as wide as possible to separate two or more class data. If the data is non linearly separable as in face data, the SVM uses kernels to map the data to linearly separable space to obtain the hyperplane decision boundaries.

Artificial Neural Networks (ANN) mimic human brain and have the capability to learn from the training data and are able to predict class labels for data which may not have been seen by the system yet. The neural network is a layered architecture and consists of a number of processing units called neurons in each layer. The performance of the neural network depends on the number of neurons in a layer and the number of layers it possesses. Neural Networks are designed in different ways - feed forward and recurrent neural networks. The flow of information is from input layer to the output layer in *feed forward* architecture while the information also flows back in the *recurrent neural network* architecture. Radial basis function neural network is an example of a feed forward neural network. A well known learning algorithm called as *Back propagation* uses the error information for weight learning. There are one or more hidden layers in each neural network. The training face feature vectors are used to train the neural network by changing the weights using appropriate learning algorithms to minimize the error.

1.5. Challenges in Face Recognition Research

Major challenges observed today in face recognition research revolve around face recognition in unconstrained natural environment. These are listed broadly as follows

- Robust face recognition under illumination, pose, expression, accessories (goggles etc.) and disguise variations.
- Improved recognition accuracy.
- Near real time Fast algorithms capable of handling large volumes of face data.
- Age invariant face recognition etc.
- Single face image based recognition
- Low resolution face images (through mobile camera or surveillance cameras)

1.6. Neural Network as a classifier

Artificial Neural Networks (ANN) learn from the data and have the ability to self organize. The ANN model of computation does not require all inputs together, rather it keeps modifying the system parameters such as learning weights, neuron centers etc. as it gets more input. Due to their high generalizing capability, the ANNs can perform well with incomplete data or faces with disguise or occlusion.

Artificial Neural networks are input output mappings of the form Y = WX, where Y is a vector of class labels for the input test feature vectors X and W is the weight matrix obtained from the training data using a suitable supervised learning algorithm. A neural network classifier learns from the training data and finds the weights in an optimized way such that all classes can be separated from each other using a definite decision boundary. The weights in neural network represent knowledge of the learned environment.

Face recognition problem is non-linearly separable in the input feature space, and it is a challenging task to place the decision boundary in terms of the synaptic weights. A face recognition problem is a many class classification problem and needs more number of decision boundaries to separate the classes. As long as the decision boundaries are straight line (in two dimensional space), plane (in three dimensional space) or hyper plane (in hyper dimensional space), the number of neurons equal to the number of decision boundaries suffices to construct a single layered neural network architecture for linearly separable data. The face data is non-linearly separable and more hidden layers are needed in the neural network. The adaptation of weights to the training data is through feeding the error of one layer to the other in backward way to minimize error. This leads to the computationally intensive algorithm for face recognition. There have been attempts in constructing an architecture of the neural network which uses less computational time and produces more accurate classification outputs.

Among various architectures existing in neural network domain such as Support Vector Machines (SVM), Radial Basis Function Neural Network (RBFNN), Pulse Coupled Neural Network (PCNN), Probabilistic Neural Network (PNN) etc, RBFNN has been found to be better suited for pattern recognition. The reason for preference of RBFNN over other architectures in pattern recognition and specifically in face recognition problems is its extraordinary generalization capabilities. The RBFNN is also computationally efficient and consists of a simple three layered structure which includes input layer, hidden layer and output layer to solve complex face recognition problem. The RBFNN hidden layer design includes obtaining the most appropriate number and centers of the Radial Basis Function (RBF) units and shape of the radial basis functions. Conventional methods use various clustering algorithms such as k-Means, Fuzzy C-Means etc. to obtain centers of RBF units which may not be the optimal ones. The RBFNN structure can be made adaptive to the underlying data and intelligent algorithms are used for obtaining optimal hidden layer structure for improved face recognition accuracy.

1.7. Research Objectives

Based on extensive literature review and observations made by us, the research objectives are listed below

Objective 1. Explore the potential of the evolutionary firefly algorithm in feature selection for improved face recognition and propose a firefly inspired feature selection algorithm.

Objective 2. Propose a new framework of firefly inspired RBFNN hidden layer design for optimal centers and number of RBF units.

Objective 3. Propose an algorithm to investigate the effect of shape of the basis functions used in RBF units on face recognition accuracy.

Objective 4. Evaluate the optimized RBFNN based Integrated Classifier.

1.8. Thesis Structure

An exhaustive literature review is presented in Chapter 2 wherein relevant literature from two broad categories are presented - growth of face recognition research in three different time frames such as 1967-1995, 1996-2005 and 2006 till date, and the face recognition research in computational intelligence domain. The research done in computational intelligence discusses work on face recognition using ANN, PSO and GA etc.

In chapter 3, a novel algorithm for feature selection is proposed. The two feature extraction methods namely Discrete Cosine Transform and Discrete Wavelet Transform are used for feature extraction due to their high information packing ability and multi resolution analysis. The proposed feature selection algorithm, inspired by the natural fireflies, uses Firefly Algorithm for feature selection. Effect of the algorithm parameter γ, which controls the speed of fireflies, is investigated for algorithm convergence, recognition accuracy and dimensionality reduction. The proposed algorithm is analyzed and compared with the feature selection methods based on Genetic Algorithm and Particle Swarm Optimization. Its superiority over the other existing evolutionary methods evaluated on four benchmarked face databases namely ORL, Yale, AR and LFW is demonstrated.

Chapter 4 presents a novel algorithm for center selection of RBF units in Radial Basis Function Neural Network (RBFNN). In this chapter a novel firefly design for RBF center selection is presented and also fitness function formulation for center selection, movement and convergence of fireflies; sub-cluster formation of face images; and role of γ parameter in controlling the speed of fireflies in the hyper-dimensional input space etc. are discussed. The parameter selection is handled carefully using the convergence metrics of the proposed algorithm and finally optimal parameters are recommended for all face databases listed above. Results and discussion on the performance of the proposed algorithm and the comparative analysis of the proposed technique with existing techniques are also presented.

In chapter 5, the work on RBFNN design is extended to the basis function shape and an algorithm on shape estimation is proposed. The algorithm uses Gaussian basis function and obtains spread based on the relative distance between nearest sub-clusters of two different classes using overlapping factor. The integrated classifier for face recognition is developed by combining the proposed center selection and basis function shape algorithms. The performance of the integrated classifier is then investigated in terms of the performance metrics such as Sensitivity Analysis, Confusion Matrix and Receiver Operating Characteristic (ROC) Curves. The performance of integrated classifier on selected face databases is compared with GA, PSO and results of other techniques reported in literature.

The conclusion of the complete research work is presented in Chapter 6.

Chapter 2

Literature Review

Face recognition from face images is a pattern recognition problem, which involves three main phases - feature extraction, feature selection and pattern classification [Fig.2.1]. Major research in face recognition revolves around these three broad topics. Many researchers have been working on different aspects of face recognition system. In this chapter, we review the techniques reported in the literature and later illustrate the research gaps.

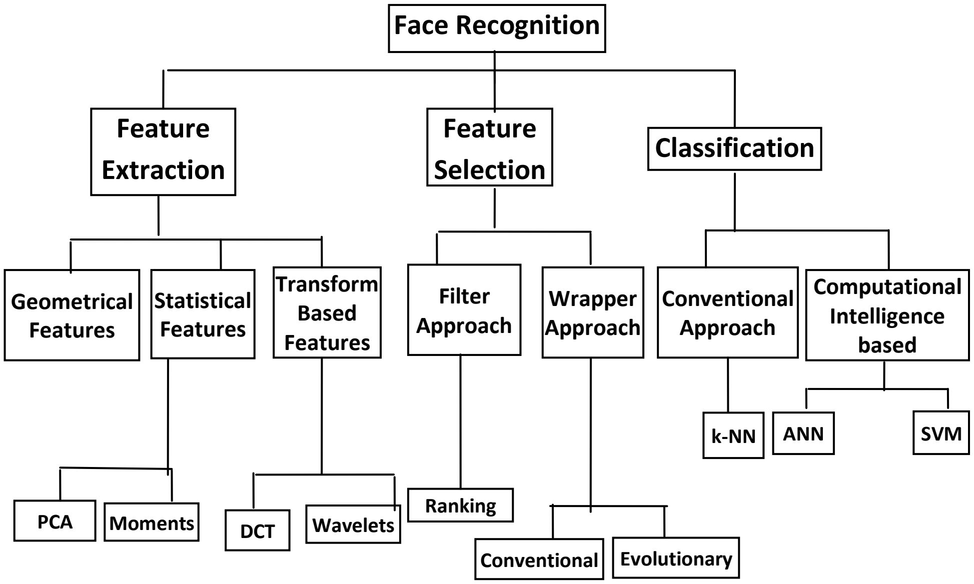


Fig.2.1 : Broad categories and sub-domains in Face Recognition Research

2.1. Overview of Growth of Face Recognition Research

In this section, we present the growth of face recognition research in about last four decades. We place that in three time domains : Early face recognition research i.e. Pre-1995 era, Middle-era i.e. the decade of 1996-2005, and recent time, i.e. post-2005 period. The early face recognition work was constrained with limited computational power of the computers available at that time. The middle era saw the maximum growth of face recognition research and the recent time the focus of face recognition has shifted to unconstrained face recognition handling illumination, pose, expression and accessories variations.

2.1.1. *Early Research in Face Recognition during 1967 - 1995*

This was an era when fast computers were not there and also constraints with respect to processing speed and memory existed. One of the earliest works in machine based face recognition was presented by *Taylor, W.K. (1967). Turk and Pentland (1991)* proposed Eigen faces approach for face recognition which proved to be the most tested approach. Their approach transformed the face images into a small set of characteristic feature images, called as Eigen faces. These were the principal components of the training images. The Eigen face method was tested on a large test bed of about 2500 persons but had limitations with respect to the sensitivity to scale and illumination variations. In this duration, feature extraction techniques were based on either the geometric features or on the Principal Component Analysis based Eigen face methods. As only very few features, mostly of an order of 5 to 10 geometric features or Eigen values were used for classification purposes, the issue of feature selection was not prevalent in this era. Neural Networks were used by *Kerin and Stonham(1990), Poggio and Girosi (1990), Bouattour et al (1992), Allinson and Ellis(1992)* etc. *Brunelli and Poggio (1993)* used geometrical face features such as nose width and length, mouth position and chin shape and used template matching for recognition of faces. The limitation of the geometric feature based approaches was that they were sensitive to scale and dimension differences and performed poorly with these variations in test images. In, first of its kind, survey of face recognition techniques in paper by *Chellappa et al (1995),* the authors consolidated research done during 1970 - 1995. Authors mentioned that the Face Recognition research in 1970s was limited to measuring attributes as was done in pattern classification problems, and the decade of 1980-1990 was dormant as regards face recognition research. Then in 1990s, the face recognition need was felt due to various surveillance requirements. Due to emergence of high speed high power computation available at low cost, neural networks reemerged due to their learning and adaptability capabilities.

2.1.2. ***Face Recognition Research in the decade of 1996-2005***

Wiskott et al (1997) used Elastic Bunch Graph Matching (EBGM) for face recognition where a face was represented as a labeled graph of nodes. The authors used Gabor Wavelet features and used similarity criteria for comparison. It was observed by *Zhang et al* (1997) that Eigen faces and EBGM were sensitive to illumination variations. To overcome these limitations, the authors combined the advantages of both techniques and used them with the back-propagation neural network. Self Organizing Map (SOM) and Convolution Neural Network were used by *Lawrence et al (1997)* for face recognition. *Belhumeur et al (1997)* proposed Fisher Linear Discriminant Analysis (FLA) method to overcome the limitation of the Eigenfaces. Fisherface method was established as the one capable of handling illumination variations.

The decade of 1990s, especially the period after 1995 has seen a steep rise in research in face recognition research. The reason for this increase was the confidence the research community had gained after the invention of systems with high computing power. Neural network being computationally intensive, revived in this decade after having almost an inactive period of the 80s. An exhaustive survey of neural networks for classification was presented by *Zhang (2000)*. The author highlights the significance of neural networks in various classification activities such as handwriting recognition, speech recognition, medical diagnosis, fault detection and bankruptcy prediction. The neural networks gained attention of researchers working in face recognition *(Jamil et al, 2001; Zhang et al, 2004; Amira and Ferrel, 2005; Liu and Wechler, 2003; Zhang et al, 2005; El-Bakry et al, 2000; Haddadnia et al, 2001; Haddadnia et al, 2002; Er et al 2002; Er et al, 2005)*. Neural network based face recognition was experimented with various feature extraction methods such as Wavelet transforms *(Zhang et al, 2004; Amira and Ferrel, 2005),* Gabor Wavelets *(Liu and Wechler, 2003; Zhang et al, 2005)*, Fourier Descriptors and PCA *(El-Bakry et al, 2000)*, Moment Invariants such as Zernike Moments *(Haddadnia et al, 2001; Haddadnia et al, 2002),* PCA *(Er et al 2002)* and Discrete Cosine Transform (DCT) *(Er et* *al, 2005)*. Support Vector Machines (SVM) and Hidden Markov Models (HMM) were also used in Face recognition during this period *(Zhao et al, 2005; Xu et al, 2003). Zhao et al (2003)* presented an exhaustive survey of face recognition technology. *Lu et al (2003)* worked with the Linear Discriminant Analysis (LDA) for face recognition.

The features extracted using above techniques were extremely large in number, mostly of the order of image size. The dimensionality of features was attempted to be reduced with a key idea of improving computational efficiency while also keeping in mind good performance. The feature selection problem in Face Recognition gained attention sometime in late 1990s. Some of the literature available on feature selection for face recognition reported are *Gokberk et al (2002); Guo and Dyer (2003)*. One of the early works in evolutionary feature selection for face recognition using genetic algorithm was proposed by *Harandi et al (2004).*

2.1.3. ***Recent Trend in Face Recognition Research***

The last decade i.e. 2006-2014 has seen a threefold increase in the interests of researchers in face recognition technology as compared to the total work in this area till 2005. The reasons for the interests were manifold- First, due to the need for automated face recognition systems to handle law enforcement and surveillance due to increased crime rate and terrorism all over the world, and second, because of availability of huge computational power at affordable price.

Since the early years of face recognition research using geometric features and Eigen faces to Fisher faces to neural networks, there were always efforts in making face recognition algorithms more robust in handling variations in illumination, pose, expression and accessories etc. The researchers were attempting to make the face recognition more accurate and robust to handle these variations. While in recent times, there is a vast growth in feature selection and classification methods. At earlier times, the most preferred choice of feature selection were Linear Discriminant analysis(LDA) and PCA with an intention of reducing the dimensionality of the input features. In recent times, feature selection methods had another objective of selecting features so as to handle variations in more accurate ways. The problem of feature selection was considered NP hard and various newer trends were introduced for selecting the best face features. This era witnessed a huge amount of research work in feature selection for face recognition. *Ekenel and Stiefelhagen (2006)* studied the effects of feature selection while *Xiao et al (2006)* proposed a feature selection based on joint boosting algorithm.

Chellappa et al (2010) presented an overview of the challenges and issues in face recognition problem in today's scenario. They explained the reason why face recognition is hard. The reasons mentioned are mainly due to acquisition conditions. Pose with respect to the camera, illumination, facial expressions and number of pixels in the face region, human aging etc. cause the human face images to undergo many changes. The authors emphasized on the needs for recognition from unconstrained video sequences and on modeling effects of aging. *Zhifeng et al (2011)* proposed a discriminative model for age invariant face recognition. Recent work focused on handling pose, expression, disguise and illumination invariant techniques so as to cater to the today's needs of unconstrained face recognition. *Wang et al (2011)* worked towards illumination normalization and *Lin et al (2011)* worked towards handling partial occlusion and illumination variations. Pose oriented face features based approach was used by *Lee et al (2012)*. *Ho and Chellappa (2013)* used Markov Random Fields and worked towards pose invariant face recognition, *De Marsico et al (2013)* worked towards uncontrolled pose and illumination variations and expression invariant techniques were presented by *Liu et al (2013)*, *Taffar et al (2013); Srinivasan and Balamurugan (2013).*

Discrete Cosine Transforms have tremendous information packing ability and have been used in handling illumination variations in face recognition. Wavelets were used to decompose the face image using multiresolution analysis. The frequency domain based face representation techniques such as DCT *(Chen et al, 2006; Dabbaghchian et al, 2010)* and Wavelet Transforms have been used in handling illumination variations *(Hu, 2011; Cao et al, 2012)*. The wavelets are said to be efficient in handling illumination variations. The authors used a wavelet based approach to obtain the illumination invariant face representation. A variety of wavelets such as Haar wavelets *(Kanan and Faiz, 2005; Nicholl et al, 2010; Ahmad et al, 2011; Radji et al, 2013)* and Gabor *(Ou et al, 2007; Choi et al, 2008; Du et al, 2009; Zhenhua et al, 2014; Hu, 2014; Ajitha et al, 2014)* wavelets have been used in face recognition research. Face recognition performances of variety of wavelets such as Daubechies, symlet, coiflet, Haar, Gabor wavelets etc. have been compared *(Utsumi et al, 2006; Dawoud and Samir, 2011).*

The features extracted using these approaches were large in number and research on face recognition also focused much at large in feature selection domain. *Yu et al (2006)* used a combined feature selection approach to select DWT and DCT based features for face recognition. *Atta and Ghanbari (2012)* used embedded DCT approach for feature selection. Other conventional feature selection techniques include Similarity feature-based feature selection by *Tran et al (2014)*, Cardinal sparse partial least square based feature selection by *Zhang et al (2014)*, multi-condition relighting with optimal feature selection *Yujie et al (2014)*, common vector approach based feature selection by *Koc and Barkana (2014)* and binary adaptive weight Gravitational Search Algorithm (GSA) based feature selection by *Chakraborty and Chatterjee (2014)*.

The feature selection problem witnessed use of evolutionary approaches such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO) etc. *(Harandi et al, 2004; Liu and Wang, 2008; Vignolo et al, 2012; Bhatt et al, 2013; Ramadan and Abdel-Kader, 2009; Cheng et al, 2011; Lei et al, 2012; Sattiraju et al, 2013; Ajit Krisshna et al 2014; Kanan et al, 2007).* Some feature selection techniques are shown in Table 2.1. In this table, some of the feature selection techniques with conventional or evolutionary approaches starting from the year 2008 till 2014 are mentioned along with the performances evaluated on a variety of face databases. Genetic Algorithms were developed by *Holland and Reitman* in early 1970s *(1977)*. These algorithms converge prematurely i.e. may get trapped in local optimal solution. PSO is another evolutionary algorithm, proposed by *Kennedy and Eberhart (1995)* which is inspired by the collective behavior of swarm of birds or fish etc. Recently a meta heuristic algorithm inspired by the flashing behavior of the natural fireflies and named as Firefly Algorithm (FA) was proposed by *Yang (2008)*. Performance of firefly algorithm in clustering has been reported to be better than PSO and ACO *(Senthilnath et al, 2011).*

As neural network learning based techniques had found place in face recognition research in early 1990s, the trend continued in the recent years. This is visible through a number of research papers published in various conferences and journals through the years 2012-2014 *(Qiakai et al, 2012; Slavkovic et al, 2013; Fatahi et al, 2013; Dong et al, 2013; Oh et al, 2013; Huang and Lin, 2014; Elazhari and Ahmadi, 2014; Lu et al, 2014).*

Table 2.1 Feature Selection approaches in Face Recognition

Year	**Authors**	**Feature Selection Method**	**Face Database (no of persons)**	**Training and testing Images**	**Selected Features**	**Recognition Accuracy**
2008	*Liu and Wang*	Genetic Algorithm	ORL(40)	5,5	35	90.50%
2009	*Li et al*	Multi-channel Dimension Reduction Scheme (MDRS)	Yale (15)	5,6	-	85.33
2011	*Cheng et al*	Binary PSO (BPSO)	ORL(40)	5,5	-	93.25%
2012	*Xiao-Dong and Wei*	Within-class distance and between class distance	Yale (15)	6,5	-	89.12%
2012	*Vignolo et al*	Genetic Algorithm	Essex(100)	5,15	56	98.0%
2013	*Darestani et al*	PSO	ORL(40)	5,5	-	90.00%
2013	*Sattiraju et al*	Adaptive Binary PSO (ABPSO)	Feret CMUPIE	8, 12 9,4	148 549	89.38% 33.58%

2014	*Ajit Krisshna et al*	Threshold-Based Binary Particle Swarm Optimization (ThBPSO)	ORL(40)
			YaleB(28) (subset 5)
				5,5
				10, 9
					187
					305
						98.14%
						100%

2014	*Koc and Barkana*	Discriminative Common Vector Approach (DCVA)	Yale (15) AR (50) AR (50)	5,5 7, 7 non occluded 3,3 occluded	4000 2200 2005	98.7% (best) 73.1% 96.3%
2014	*Tran et al*	Similarity Feature-Based Selection	ORL (40) YaleB (38)	5,5 6,58	- -	98.0% 87.23%

2014	*Chakraborti and Chatterjee*	Binary Adaptive Weight using GSA	ORL(40)
			YaleA (15)
			YaleB (32)
			AR(100)
				5,5
				6,5
				10, remaining
				5,5
						91.7%
						98.1%
						98.7%
						95.5%

Computational Intelligence based techniques emulate human intelligence achievable through computational models designed carefully. Three broad categories of algorithms are - Artificial Neural Networks(ANN), Fuzzy Computing (FC) and Evolutionary Algorithms(EA). *Zhang and Zuo (2007)* presented a paper on Computational Intelligence based biometric technologies. Back Propagation (BP) and Multilayer perceptron (MLP) based neural networks have limitations in handling complex problems like face recognition. The authors illustrate that radial basis function neural network (RBFNN) has the potential to handle complex face recognition problem. RBFNN possesses good approximation capability, has less computational requirements and fast learning speed as compared to other architectures such as back propagation neural network *(Poggio and Girosi, 1990; Zhang and Zuo , 2007; Bishop, 1995; Er et al, 2002).*

2.2. Computational Intelligence Based Face Recognition Techniques

In this thesis work, we use Radial basis function neural networks (RBFNN) as our neural network model for classification. The potential of the Evolutionary Algorithms has been explored for feature selection problem and for optimal design of RBFNN hidden layer. Fuzzy Computing algorithms are not explored currently as we are not aiming at modeling imprecise and incomplete information in face recognition. Therefore, an exhaustive literature review of the Computational Intelligence based techniques in feature selection and RBFNN hidden layer design is presented in this section.

*2.2.1. **Evolutionary Feature Selection for Face Recognition***

Harandi et al (2004), Liu and Wang (2008), Vignolo et al (2012) and Bhatt et al (2013) used Genetic Algorithm (GA) for feature selection for improved face recognition. Feature Selection using Particle Swarm Optimization (PSO) evolutionary algorithm for face recognition has been proposed by *Ramadan and Abdel-Kader (2009), Cheng et al (2011), Lei et al (2012), Sattiraju et al (2013) and Ajit Krisshna et al (2014).* Ant Colony Optimization(ACO) was used by *Kanan et al (2007)* for feature selection for face recognition application. These algorithms have disadvantages in that (i) they converge very slowly to a solution, and (ii) the global optimum solution is not guaranteed. Firefly algorithm is said to be performing well as compared to other algorithms such as GA, PSO or ACO in applications such as function approximation or clustering *(Yang, 2008; Senthilnath et al, 2011)* But its potential has not been explored in face recognition so far to the best of our knowledge. The algorithms such as GA and PSO are effective only for single mode solution, while Firefly Algorithm (FA) proposed by Xin-She Yang is efficient for multi modal optimizations also.

2.2.2. Face Recognition using Radial Basis Functions Neural Networks

Radial basis function neural networks (RBFNN) have been used for various face processing tasks such as face recognition, facial expression recognition, gender, age determination etc. *(Zhang and Zuo, 2007).* RBFNN consists of three layers - input layer, hidden layer and the output layer. The design of hidden layer is of utmost importance as it captures the underlying structure of the training data *(Bishop, 1995)*. The parameters of the hidden layer are: center and spread of the radial basis function (RBF) units, number of RBF units and the choice of the basis function. The performance of RBFNN depends on the structure of the hidden layer and the learning weights. Radial basis functions at the hidden layer of the RBFNN perform the nonlinear mapping of the input space to the linearly separable hyperspace.

RBFNN based face recognition problem was addressed by many researchers who used different feature extraction methods such as Discrete Cosine Transform (DCT), Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) and Kernel Discriminant Common Vector (KDCV) *(Er et al, 2002; Er et al, 2005; Jing et al, 2008).* A fuzzy hybrid learning based RBFNN for face recognition was proposed by *Haddadnia et al (2003).* An incremental learning algorithm for RBFNN for face recognition by *Wong et al (2011). Balasubramanian et al (2009)* used face and mouth information for real time face recognition using RBFNN. A number of key researches in the area of center selection for RBF units in face recognition exist in literature which are described as follows.

2.2.3. Research in RBFNN Center Selection

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 *(Bishop, 1995).* The design of RBFNN has been of research interest in face recognition application. Different approaches such as pruning and growing *(Er et al, 2002, Er et al, 2005),* fuzzy hybrid learning *(Haddadnia et al, 2003),* polynomial based RBF neural networks *(Oh et al, 2013),* point symmetry distance *(Sing et al, 2007)* and self adaptive approach *(Sing et al, 2009)* were used in RBFNN design for face recognition.

Er et al (2002) used PCA for feature extraction and trained the RBFNN centers by splitting of a cluster, iteratively obtained. The authors considered the centers of the RBFNN as the class means using the supervised information about the classes. They initially set the total number of centers to be equal to the total number of output classes. All training face images of a class belonged to the cluster identified by the class mean and the distance of the furthest point of the cluster defined the span of the cluster. The authors split the cluster into two if one class embodied in it another class completely. The iterative process kept splitting the intermediate clusters into two, if needed.

Haddadnia et al (2003) proposed a technique named as FHLA (Fuzzy Hybrid Learning Algorithm) 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 *(Oh et al, 2013).* The FCM technique has limitation of being sensitive to initialization and getting trapped at local optima *(Izakian and Abraham, 2011).* RBFNN training for face recognition has been done using a modified k-Means clustering algorithm using Point Symmetry Distance *(Sing et al, 2007).* 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 *(Sing et al, 2009).*

Clustering is considered as one of the most difficult and challenging problems in machine learning and is NP hard *(Hruschka et al, 2009; Tsekouras and Tsimikas , 2013; Wang et al, 2012).* The clustering algorithms such as k-Means, FCM, Expectation Maximization (EM) etc. get trapped in the local optima while evolutionary algorithms provide near optimal solution in reasonable time *(Hruschka et al, 2009).* Various evolutionary algorithms such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Artificial Bee Colony(ABC) etc. have been used in the design of RBFNN in various applications such as function approximation, web source classification and machine learning tasks etc.

Various studies in applications other than face recognition also exist in literature illustrating the significance and need for better methods in RBFNN center selection. *Mao (2002)* proposed Fisher linear class separability measure based center selection for studies involving machine learning. *Han et al (2010)* used self organizing RBF (SORBF) for function approximation. Recursive orthogonal least square algorithm (ROLSA) was used for the design of RBFNN hidden layer by *Huang and Zhao (2005).* Clustering plays an important role in capturing the structure of the data, especially the centers of the RBF units. *Wang et al (2012)* emphasized use of supervised information in clustering and proposed an output constricted clustering for RBFNN initialization. *Tsekouras and Tsimikas (2013)* also used input-output fuzzy clustering and particle swarm optimization for center determination for applications in machine learning. In almost all studies reported above, the researchers put efforts in designing the structure of the hidden layer and few of them also worked on obtaining optimal weights of the network while many of them did not work towards face recognition. A summary of the research in the RBFNN with respect to the work done towards parameters such as center and number of RBF units, width and weights, and the application used for the research, is presented in Table 2.2 and Table 2.3.

Various evolutionary algorithms such as genetic algorithm *(Gan et al, 2012; Oh et al, 2014),* particle swarm optimization *(Tsekouras and Tsimikas, 2013; Feng et al, 2010; Oh et al 2012; Alexandridis et al, 2013; Feng, 2006),* artificial bee colony *(Yu and Duan, 2013)* and memetic pareto algorithm *(Qasem et al, 2012)* etc. have been reported in design of RBFNN. Recently a meta heuristic optimization algorithm named Firefly Algorithm (FA) was proposed by *Yang (2008)* in which the author proposed the idea of attractiveness between the fireflies for solving optimization problems in which randomly generated fireflies move in the search space to reach the brightest firefly. In object tracking application, it was established by *Gao et al (2013)* that the firefly algorithm outperforms particle swarm optimization. *Yu et al (2013)* use firefly algorithm in multithreshold image segmentation and established the superiority of the firefly algorithm over genetic algorithm. A detailed survey of various other swarm intelligence based algorithms was presented by *Yang (2013).* A comprehensive review of variants of firefly algorithms was studied by *Fister et al (2013).* A clustering algorithm based on firefly algorithm was proposed by *Senthilnath et al (2011).* The authors established that the firefly algorithm is reliable, efficient and robust for clustering and it outperforms PSO and ant colony optimization (ACO) techniques in various machine learning applications.

Table 2.2: Research in RBFNN Design (2002-2010)

Year	**Authors**	**RBFNN Parameters**				**Method**	**Major Application Area**
		Center	**Width (Shape)**	**Number of neurons**	**Weight**		
2002	*Er et al*	Yes	Yes	Yes	Yes	Splitting and Merging sub-clusters	Face Recognition
2002	*Mao*	Yes	No	No	No	Fisher ratio class separability measure	Classification
2003	*Haddadnia et al*	Yes	Yes	Yes	Yes	Fuzzy Clustering	Face Recognition
2005	*Er et al*	Yes	Yes	Yes	Yes	Clustering	Face Recognition
2005	*Huang and Zhao*	Yes	No	Yes	No	Recursive Orthogonal Least Square Algorithms (ROLSA)	Function Approximation
2006	*Feng*	Yes	Yes	Yes	Yes	Particle Swarm Optimization	Nonlinear function approximation
2007	*Sing et al*	Yes	Yes	Yes	Yes	Point Symmetry Distance	Face recognition
2008	*Jing et al*	Yes	No	No	No	kernel discriminative common vectors (KDCV)	Face recognition
2009	*Sing et al*	Yes	Yes	Yes	Yes	Self Adaptive RBFNN	Face recognition
2010	*Han et al*	Yes	Yes	Yes	No	Growing and Pruning	function approximation
2010	*Feng et al*	Yes	No	No	Yes	Swarm Intelligence Clustering	Deep Web Resources

Table 2.3: Research in RBFNN Design (2011-2014)

Year	**Authors**	**RBFNN Parameters**				**Method**	**Major Application Area**
		Center	**Width (Shape)**	**Number of neurons**	**Weight**		
2011	*Wong et al*	Yes	No	No	Yes	Recursive Orthogonal Least Square (ROLS) Algorithms	Face Recognition
2012	*Oh et al*	Yes	Yes	No	Yes	Particle Swarm Optimization and Differential Evolution	Machine Learning
2012	*Gan et al*	Yes	Yes	Yes	Yes	Genetic Algorithms	Nonlinear Time Series Prediction
2013	*Oh et al*	No	No	No	Yes	Particle Swarm Optimization with PNN using fuzzy C-means approach	Face Recognition
2013	*Tsekouras and Tsimikas*	Yes	Yes	Yes	Yes	Fuzzy Clustering and Particle Swarm Optimization	Machine Learning
2013	*Alexandridis et al*	Yes	Yes	Yes	Yes	Non-Symmetric partition of input space and PSO	Machine Learning
2013	*Yu and Duan*	Yes	Yes	Yes	Yes	Artificial Bee Colony and Fuzzy C-Means	Image Fusion
2014	*Oh et al*	Yes	Yes	No	Yes	Parallel Genetic Optimization	Nonlinear Function Approximation

It is observed that the potential of firefly algorithm has not been explored in RBFNN design and therefore we propose to use Firefly Algorithm in the present thesis for RBFNN design.

2.2.4. Research in RBFNN Basis Functions and their Shape

The face recognition research using RBFNN has been focused on issues such as center initialization, spread and weights learning. The most preferred choice of basis functions is the Gaussian Basis Function. However, a variety of radial basis functions such as Gaussian, Multiquadric, Inverse multiquadric, Thin plate spline functions etc. have been used by researchers in solving partial differential equations (PDE) and in function approximation. The facial expression recognition using cloud basis functions is proposed by *De Silva et al (2008)*. An importance of the choice of adequate basis function is described by *Rocha (2009)*. *Cheng (2012)* studied the Multiquadric family of radial basis function and the shape parameters. The effect of different basis functions on a RBFNN for time series prediction was studied by *Harpham and Dawson* (2006). The q-Gaussian radial basis function has been used by *Fernandez-Navarro et al (2012)* for binary classification. It was also suggested in the same paper that the choice of the radial basis function should be the part of the optimization problem instead of a priori choice. Hardy's Multiquadric *(Hardy 1971; Hardy 1990)* was found to be highly efficient for interpolating continuous multivariate functions as well as for the solution of the partial differential equations.

Wertz et al (2006) studied the role of shape parameters and worked on searching for optimal shape parameters. *Sarra (2006)* observed that determining the optimal shape parameter is an open problem and is often chosen by brute force method. Shape parameter has also been studied by *Huang et al (2007a)* where the authors used the differentiation of the error estimate to compute the optimal value of the shape parameter. *Tsai et al (2010)* used the golden search method for finding a good shape parameter for meshless collocation method. Optimal shape parameter for the basis functions were computed and applied in applications mainly in function approximation and interpolation by many researchers. *Fornberg and Piret (2008)* investigated the shape of radial basis functions in solving PDEs. *Bayona et al (2011)* investigated the shape parameter and illustrated that the accuracy strongly depends on the value of shape parameter.

2.3. Gaps in Existing Research

Based on the literature review reported in this chapter, we observe the following

- The major challenge in Face Recognition research today is to develop techniques capable of handling variations in illumination, pose, expression, accessories such as spectacles, goggles or disguise.
- Geometric features are sensitive to the scale and orientation variations. These are not able to handle expression and pose variations.

- Frequency based features such as DCT, WT etc. have been proved to have potential for better face representation than PCA based Eigenfaces which are sensitive to the illumination variations.
- Feature Selection being an NP hard problem can be solved efficiently using evolutionary algorithms, such as GA, PSO, ACO etc. which are capable of computing the near optimal (approximately best) features in polynomial time.
- Firefly Algorithm has been used in applications such as function approximation and clustering where it was established that firefly algorithm is better than GA, PSO or ACO. Its potential in solving feature selection problem was not explored earlier.
- Neural Networks are better classifiers for face recognition than others such as k-NN, Naive Bayes' classifier, Decision Tree, SVM etc. These later techniques do not possess better generalization ability and ability to adapt to the underlying face data as compared to the neural networks.
- Radial Basis Function Neural Network architecture is a better classifier as compared to other architectures of neural networks. An optimal structure of the hidden layer involving the center and number of RBF units (neurons) can produce better performances.
- Various techniques such as merge and split, GA, PSO, fuzzy logic etc., have been used in designing the optimal structure of the RBFNN hidden layer, but the potential of FA has not been explored in the RBFNN design.
- We conclude that the computational Intelligence based techniques such as neural networks and evolutionary algorithms display strength in handling challenges of face recognition research.

Based on exhaustive literature review, following gaps have emerged, resulting in impetus to develop

1. Intelligent algorithms with *improved face recognition performance* to handle *variations* in illumination, pose, expression and accessories
2. *A*lgorithms having more *adaptation* to the structure of the underlying data to identify natural clusters
3. Faster learning of the sample face images so as to enable almost *real time* face recognition

4. *Faster* feature selection algorithms

We therefore propose to work on face recognition problem using computational intelligence techniques namely Radial Basis Function Neural Network (RBFNN) and evolutionary Firefly Algorithm (FA).

Chapter 3

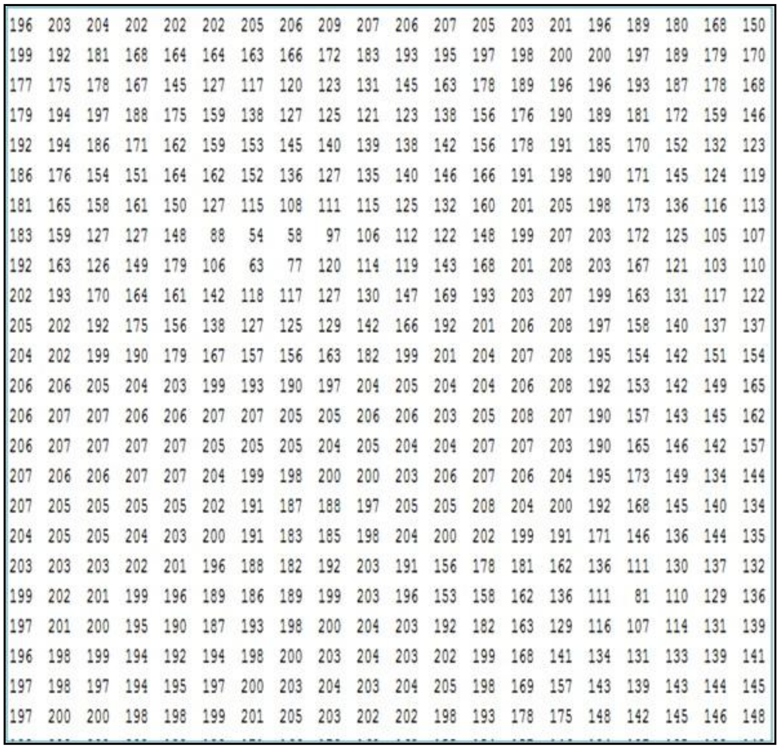
Firefly Inspired Evolutionary Feature Selection

3.1. Introduction

Feature Selection is a significant phase of any pattern recognition problem which contributes not only in selecting non-redundant and relevant features for efficient classification, but it also reduces the dimensionality of the problem space. Feature extraction and selection are the two important phases of any face recognition system. The features normally computed as geometric or mathematical coefficients define a human face. Various techniques of face representation use geometric features, Principal Component Analysis (PCA) based Eigen Faces, statistical moments and transform based techniques such as Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT) etc.

Feature Selection attempts to select the most discriminatory *k* number of features from a total of *d* features (k << d), which leads to the best face recognition accuracy. These features are carefully extracted and selected so as to meet today's challenges in face recognition research such as ability to handle variations in illumination, pose, expression and accessories etc. Feature selection is a combinatorial problem and is viewed as a search problem. Computing the best features using an exhaustive Brute Force method is computationally expensive and is of order $O(2^d)$ where *d* is the total number of initially extracted features. The feature selection problem is said to be NP hard problem and research efforts attempt to achieve an optimal solution in polynomial time.

A human face for a machine or a computer is a collection of numeric values, called as gray values, arranged uniquely to identify the person whose face image it is and should be close to the same arrangement of these values despite any variations in image acquisition conditions such as light source direction, pose and expression of the person etc. The camera acquires the image of human face by recording the light reflected from the surface of human face by an array of sensors at the camera. The light falling on a small unit area of the three dimensional face of a person, gets reflected appropriately in unique direction due to unique three dimensional facial structure of bones and muscles. The amount of light captured by the detector at the camera is quantized to form gray values and a face for a machine is only a two dimensional array of values as shown in Fig 3.1. Camera resolution plays an important role in gaining the fineness of the gray value for each detector capturing corresponding picture element. A face image thus looks like a two dimensional array of quantized intensity values, of which only few are relevant and contain non-redundant information.



196	203	204	202	202	202	205	206	209	207	206	207	205	203	201	196	189	180	168	150
199	192	181	168	164	164	163	166	172	183	193	195	197	198	200	200	197	189	179	170
177	175	178	167	145	127	117	120	123	131	145	163	178	189	196	196	193	187	178	168
179	194	197	188	175	159	138	127	125	121	123	138	156	176	190	189	181	172	159	146
192	194	186	171	162	159	153	145	140	139	138	142	156	178	191	185	170	152	132	123
186	176	154	151	164	162	152	136	127	135	140	146	166	191	198	190	171	145	124	119
181	165	158	161	150	127	115	108	111	115	125	132	160	201	205	198	173	136	116	113
183	159	127	127	148	88	54	58	97	106	112	122	148	199	207	203	172	125	105	107
192	163	126	149	179	106	63	77	120	114	119	143	168	201	208	203	167	121	103	110
202	193	170	164	161	142	118	117	127	130	147	169	193	203	207	199	163	131	117	122
205	202	192	175	156	138	127	125	129	142	166	192	201	206	208	197	158	140	137	137
204	202	199	190	179	167	157	156	163	182	199	201	204	207	208	195	154	142	151	154
206	206	205	204	203	199	193	190	197	204	205	204	204	206	208	192	153	142	149	165
206	207	207	206	206	207	207	205	205	206	206	203	205	208	207	190	157	143	145	162
206	207	207	207	207	205	205	205	204	205	204	204	207	207	203	190	165	146	142	157
207	206	206	207	207	204	199	198	200	200	203	206	207	206	204	195	173	149	134	144
207	205	205	205	205	202	191	187	188	197	205	205	208	204	200	192	168	145	140	134
204	205	205	204	203	200	191	183	185	198	204	200	202	199	191	171	146	136	144	135
203	203	203	202	201	196	188	182	192	203	191	156	178	181	162	136	111	130	137	132
199	202	201	199	196	189	186	189	199	203	196	153	158	162	136	111	81	110	129	136
197	201	200	195	190	187	193	198	200	204	203	192	182	163	129	116	107	114	131	139
196	198	199	194	192	194	198	200	203	204	203	202	199	168	141	134	131	133	139	141
197	198	197	194	195	197	200	203	204	203	204	205	198	169	157	143	139	143	144	145
197	200	200	198	198	199	201	205	203	202	202	198	193	178	175	148	142	145	146	148

Fig. 3.1 Numeric Representation of a Face for Machine based Face Recognition

Each individual has a unique combination of geometrical features and the corresponding image also reflects structural properties unique to the person's face thus representing a unique pattern of these gray values. Gray values are processed to gain more information, useful in face recognition, by using various techniques known as Feature Extraction techniques. All features that are extracted may or may not form a unique pattern and may not match the human perception of facial features such as eyes, nose, mouth etc. There is a need for selecting features forming unique patterns which are capable of distinguishing different persons by their face images. The problem of face recognition therefore reduces to a *pattern recognition problem*. A machine has a challenge to understand the *pattern* or arrangement of such gray values for each individual.

A *face recognition system* is trained with the selected features obtained from training face images of each person [Fig.3.2]. The test face images are also subjected to feature extraction and selection using the same methods as used for the training face images. A suitable classifier then recognizes the person by the test face image by finding its similarity with one of the persons face images in the training database (termed as authorized) or discards the identity of the person being tested (termed as impostor).

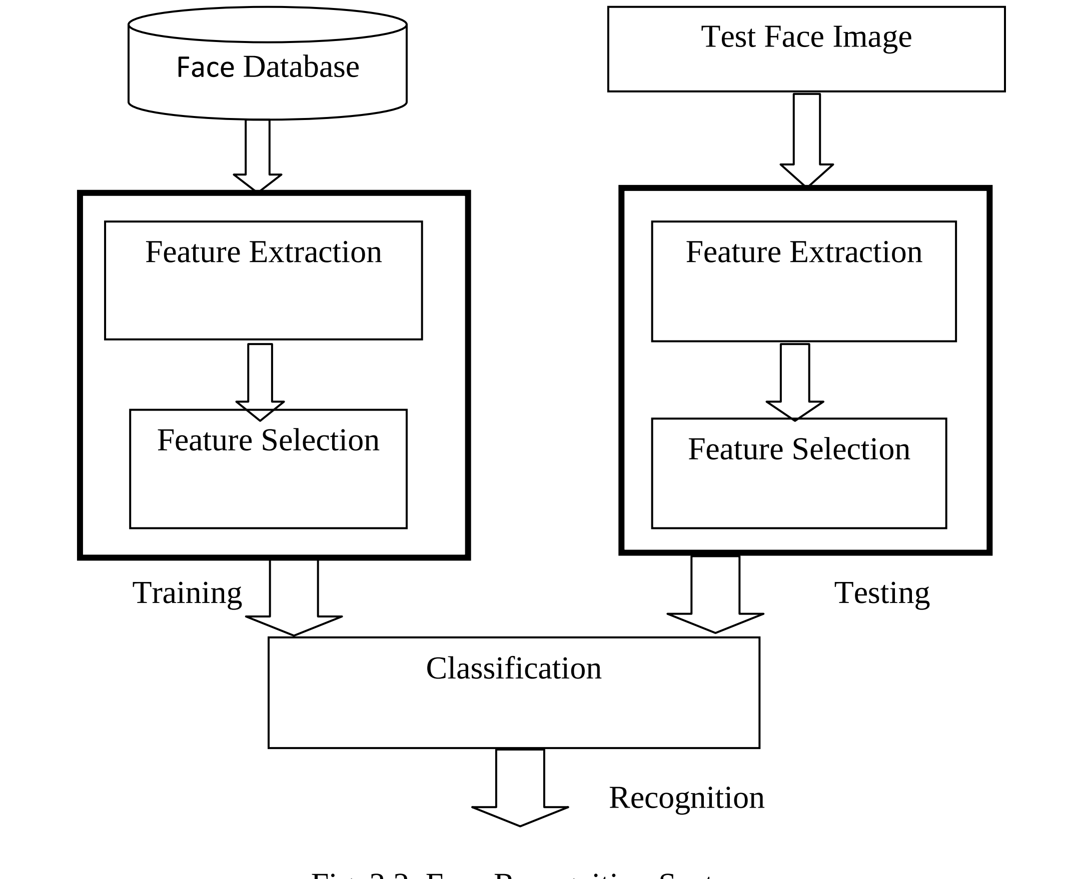


Fig. 3.2. Face Recognition System

The feature selection phase represents a person's face as a collection of relevant, non-redundant and most discriminative features f_1, f_2, f_3, ...f_k where *k* is the total number of selected features. These are represented as a k-dimensional vector <f_1, f_2, f_3, ...f_k> called as *feature vector*. As the face is a complex real world object and uses a large number of features, the face data is said to be *hyper-dimensional*. Based on the fact that a person's face structure remains same, the feature vectors corresponding to the same person's different face images captured at different illumination, pose, expressions and occlusion etc. should resemble closely while being largely different from other person's face image feature vectors. It is observed that the feature vectors of all training face images of one person, if plotted as points on a k-dimensional space, accumulate near each other and form cluster distant apart from other person's cluster of feature vectors as shown in Fig. 3.3. In this figure the axes f_1 and f_2 represent only two features used to present a two dimensional visualization of the input feature space in which each face image is represented as a point.

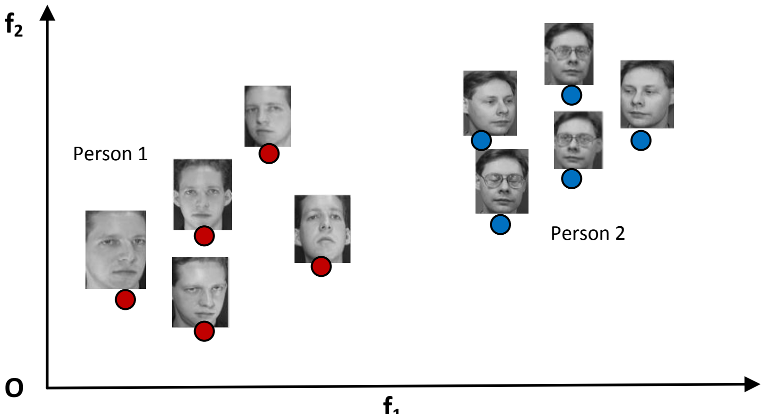


Fig. 3.3. Sample two dimensional feature space representing feature vectors as points

In this chapter, a novel algorithm for feature selection is proposed. The proposed algorithm uses Firefly Algorithm for feature selection to improve face recognition performance. The proposed Firefly Inspired Feature Selection (FIFS) algorithm is evolutionary in nature as it learns from the data and is inspired by the flashing behavior of the natural fireflies. The algorithm parameter selection and convergence are investigated taking fitness function as recognition accuracy. The effect of number of features, number of training images and the feature extraction methods namely DCT and DWT using Haar wavelets on overall recognition accuracy are investigated in this chapter. The proposed algorithm is also compared with some of the existing feature selection techniques using other evolutionary algorithms such as Genetic Algorithm and Particle Swarm Optimization.

3.2. Feature Extraction Techniques

Human face consists of a definite arrangement of visible organs such as eyes, nose and mouth. Early face recognition techniques relied on these geometric features. The features extraction techniques consisted of various edge detection techniques. These techniques use the rate of change of gray values in horizontal, vertical and diagonal directions. Canny edge detector, Hough transform and other edge enhancement methods have also been used to detect the important edges describing the eyes, nose and mouth regions of all persons.

Major features considered for face recognition are the center points of two eyes, distance between center points of two eyes, length of nose, angle between base line, i.e. the line perpendicular to the line joining two eyes, and the line joining nose point to eye center, length of lips etc. Geometric features can be taken as segments, perimeters or areas of definite shapes on the face region. The disadvantage of geometric features is that they are not robust with respect to expressions and are less preferred for face recognition.

3.2.1. ***Principal Component Analysis based Eigen Faces***

Principal component analysis (PCA), also known as Karhunen-Loeve method, is used as a feature extraction method and it transforms the data into uncorrelated data using Orthogonal Transformations. The PCA technique attempts to find the directions of maximum variance in the data and the transformed values of data in these directions are called *principal component*s. The face image consisting of a two dimensional array of numeric values is projected along the axes of maximum variances.

Consider a set of N sample training face images $\{x_1,x_2, .., x_N\}$ belonging to K person classes say $(C_1, C_2, .., C_K)$. Let each image has a total of *n* pixels in it and let *m* (m < n) be the reduced dimension of the feature space obtained by projecting *n* dimensional input space to the new space. The transformed features $y_k \in \Re^m$ are defined as the following linear transformation.

$$y_k = W^T x_k \qquad\qquad k = 1,2, \dots n \tag{3.1}$$

where $W \in \Re^{n\times m}$ is a matrix with orthonormal columns.

Consider the total scatter matrix

$$S_T = \sum_{k=1}^{n} (x_k - \mu)\,(x_k - \mu)^T \tag{3.2}$$

where $\mu \in \Re^n$ is the mean of all sample images. The scatter of the transformed feature vectors $\{y_1, y_2, .., y_N\}$ is $W^T S_T W$. Matrix W_{opt} selected optimally so as to maximize the determinant of $W^T S_T W$ is given as

$$W_{opt} = \arg\max_W |W^T S_T W| = [w_1\ w_2\ \ ..w_m] \tag{3.3}$$

where $\{w_i \mid i = 1, 2, ..., m\}$ is the set of n-dimensional eigen vectors of S_T corresponding to m largest eigen values. Thus each eigen vector is of size similar to the input dimension, i.e. *n*. If these eigen vectors are displayed as a matrix, they appear like a face, and are called as Eigen faces. The limitation of the Eigen vector approach to face recognition is that it is sensitive to illumination differences and performs well only in constrained environment with almost nil illumination variations.

An extension of the Eigen Face approach is known as Fisher Face approach. In this approach the matrix W used in (3.1) is selected in such a way that the ratio of the *between-class scatter* and the *within-class scatter* is maximized. If μ_i is the mean of all face images in class C_i, then the between-class scatter matrix is defined as

$$S_B = \sum_{i=1}^{K} N_i(\mu_i - \mu)\,(\mu_i - \mu)^T \tag{3.4}$$

where N_i is the number of sample face images in class i and μ is the overall mean of all images. The within-class scatter matrix is defined as

$$S_W = \sum_{i=1}^{K} \sum_{x_k \epsilon C_k} (x_k - \mu_i)(x_k - \mu_i)^T \tag{3.5}$$

The W_{opt} is chosen as the matrix

$$W_{opt} = \arg\max_W \frac{|W^T S_B W|}{W^T S_W W} = [w_1\ w_2\ \ ..w_m] \tag{3.6}$$

The Fisher face technique is illumination invariant.

3.2.2. ***Discrete Cosine Transform***

Pixels in an image are highly correlated and 'Transforms' such as Discrete Cosine Transform (DCT) remove the redundancy between the neighboring pixels. Efficiency of a transformation scheme is said to be more if it has the ability to pack maximum information content in the given image in as few coefficients as possible. The utility of high frequency coefficients is less in face recognition as most information content in a human face lies in the structure of the face in cheek region, or the details around eyes and mouth. Most of the transform based techniques, therefore, use low frequency coefficients for face representation. Usefulness and compaction ability of any transform can also be assessed by reconstructing the images from only few most informative coefficients of their transformed counterparts while the subjective quality of the reconstructed images should be as good as the original image.

Discrete Cosine Transform (DCT) , originally proposed by *Ahmed et al (1974)*, has a great information packing capability, which means that only a very small number of DCT coefficients are able to represent the complete image. In face recognition domain, a two dimensional DCT is applied to entire face image to obtain transformed image with few coefficients carrying maximum information. Coefficients at the upper left region of the image capture low frequency components of the face image while the high frequency components are captured as lower right coefficients. JPEG format uses only a few coefficients to compress the entire image which on transmission to the receiving end is reconstructed to the image quality acceptable by humans. An image is a two dimensional matrix of pixel intensities and is represented as f(x,y), x = 0,1,..., M and y = 0,1, .., N. We limit the mathematical description of the DCT with respect to the 2D data as we only handle images in this work. The 2D DCT is given by

$$F(u,v) = \frac{1}{\sqrt{MN}}\alpha(u)\alpha(v)\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y) \times \cos\left(\frac{(2x+1)u\pi}{2M}\right) \times \cos\left(\frac{(2y+1)v\pi}{2N}\right) \qquad (3.7)$$

For u = 0,1,..,M and v = 0,1,..N

Where

$$\alpha(w) = \frac{1}{\sqrt{2}} \quad \text{for } w = 0$$

and $\quad \alpha(w) = 1 \quad$ otherwise

F(u,v) represents the transformed image coefficient at u,v position.

3.2.3. ***Discrete Wavelet Transform***

Wavelets are defined by the wavelet function ψ(t), also known as *mother wavelet* and the *scaling function* φ(t) in the time domain. A signal in time and space is approximated by a number of wavelets and the coefficients represent the complete signal. Wavelets have been effectively used in Data Compression application due to their information packing ability and providing information of signals in both frequency and time. A face image, like any two dimensional signal in space and time, can be represented by a number of wavelet coefficients in four different aspects capturing variations in horizontal, vertical and diagonal directions while the low frequency contents remain as approximation coefficients. Wavelets are used for multi-resolution analysis of face images.

Face of a human being has some regions with more number of intensity changes per unit distance (horizontally or vertically) known as *high frequency* components, and some regions with less number of intensity changes per unit distance, known as *low frequency* components. High frequency facial features areas are around eyes, nose and mouth, while forehead and cheek regions represent the low frequency components. As a human face, physically is a three dimensional structure, its frequencies are different in different space regions and time. It is said that most information is contained in the low frequency components of face. Multi-resolution analysis attempts to find wavelet coefficients in different resolutions. A variety of wavelets are used to approximate any signal in space and time such as Daubechies, Gabor, curvelet etc. Haar wavelet is a special case of Daubechies and is the simplest one for representing any signal. The human face captures maximum information in the low frequencies and therefore low frequency coefficients represented by the approximation coefficients are used as features for face recognition. Two functions ϕ(x) and ψ(x), known as Haar scaling function and Haar wavelet function respectively, define a 2-dimensional Haar wavelets. The unit height, unit width Haar scaling function φ(x) and wavelet function ψ(x) are described as given below

$$\varphi(x) = \begin{cases} 1 & 0 \leq x < 1 \\ 0 & \text{otherwise} \end{cases} \tag{3.8}$$

$$\psi(\mathrm{x}) = \begin{cases} 1 & 0 \leq \mathrm{x} < 0.5 \\ -1 & 0.5 \leq \mathrm{x} < 1 \\ 0 & \text{otherwise} \end{cases} \tag{3.9}$$

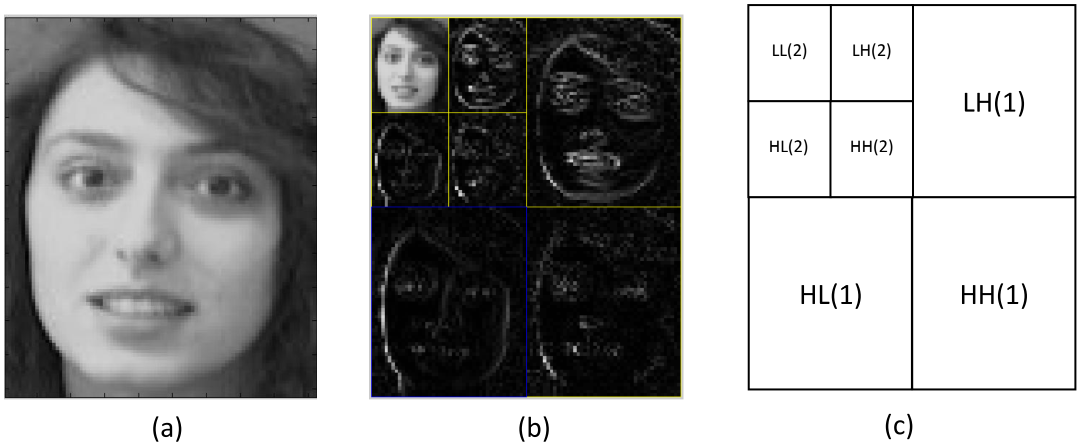


Fig. 3.4. Multiresolution Analysis using DWT (a) Original Face Image from ORL face database (b) Two Level Decomposition (c) Corresponding Resolution Details

The multi-resolution analysis requires two dimensional scaling function φ(x,y) and three two dimensional wavelets $\psi^H(x,y)$, $\psi^V(x,y)$ and $\psi^D(x,y)$. The wavelets $\psi^H(x,y)$, $\psi^V(x,y)$ and $\psi^D(x,y)$ measure the intensity variations along columns (horizontal details), rows (Vertical details) and the variations along the diagonal respectively. A two dimensional scaling function is defined as

$$\varphi(x,y) = \varphi(x)\varphi(y) \tag{3.10}$$

The directionally sensitive two dimensional wavelets $\psi^H(x,y)$, $\psi^V(x,y)$ and $\psi^D(x,y)$ are defined respectively as products ψ(x)φ(y) , φ(x)ψ(y) and ψ(x)ψ(y). The scaled and translated basis functions are defined as

$$\varphi_{j,m,n}(x,y) = 2^{j/2}\varphi\left(2^{j/2}\mathrm{x} - \mathrm{m}, 2^{j/2}\mathrm{y} - \mathrm{n}\right) \tag{3.11}$$

$$\psi^{i}_{j,m,n}(x,y) = 2^{j/2}\psi^{\mathrm{i}}\left(2^{j/2}\mathrm{x} - \mathrm{m}, 2^{j/2}\mathrm{y} - \mathrm{n}\right) \tag{3.12}$$

Where i = {H,V,D}; m and n represent the translation in x and y directions respectively; H, V and D represent the horizontal, vertical and diagonal directions respectively. The discrete wavelet transform of image f(x,y) of size M×N is given by

$$\mathrm{W}_{\varphi}(\mathrm{j}_0,\mathrm{m},\mathrm{n}) \; = \frac{1}{\sqrt{mn}}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\mathrm{f}(\mathrm{x},\mathrm{y})\varphi_{\mathrm{j}_0\mathrm{m},\mathrm{n}}(\mathrm{x},\mathrm{y}) \tag{3.13}$$

$$\mathrm{W}_{\psi}^{\mathrm{i}}\;(\mathrm{j},\mathrm{m},\mathrm{n}) = \frac{1}{\sqrt{mn}}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\mathrm{f}(\mathrm{x},\mathrm{y})\psi_{\mathrm{j,m,n}}^{\mathrm{i}}(\mathrm{x},\mathrm{y}) \tag{3.14}$$

where i = {H,V,D} and parameter j_0 is an arbitrary starting scale. The coefficients $W_\varphi(j_0,m,n)$, for m = 0,1,2,....,M-1 and n = 0,1,2,...,N-1, define approximation of the original face image f(x,y) at scale j_0. The $W^i_\psi(j, m,n)$ coefficients define the details in three directions for higher scales $j \geq j_0$. The 2D discrete Fast Wavelet Transform (FWT) is implemented using digital filters and down-samplers. The FWT algorithm exploits the relationship between the coefficients of adjacent scales and the highest scale coefficients are assumed to be samples of the function itself. That is $W_\varphi(J, m,n)$ is considered to be the input image f(x,y), where J is the highest scale such that $W_\varphi(J, m,n)$ is initialized as original face image. The value of J is taken as the maximum of the values of $\log_2(M)$ and $\log_2(N)$ while j_0 is initialized as 0.

The rows of $W_\varphi(j,m,n)$ are convolved with the analysis filter functions $h_\varphi(-n)$ and $h_\psi(-n)$, where $h_\varphi(n)$ and $h_\psi(n)$ are the scaling and wavelet vectors. These are defined as follows

$$h_\varphi(n) = \begin{cases} \dfrac{1}{\sqrt{2}} & n = 0,1 \\ 0 & otherwise \end{cases} \tag{3.15}$$

$$h_\psi(n) = \begin{cases} \frac{1}{\sqrt{2}} & n = 0 \\ \frac{-1}{\sqrt{2}} & n = 1 \\ 0 & otherwise \end{cases} \tag{3.16}$$

The function $h_\varphi(n)$ represents the low pass filter and $h_\psi(n)$ represents the high pass filter. At each iteration, the rows of the two dimensional approximation coefficients $W_\varphi(j+1,m,n)$ (visualized as a 2- dimensional matrix) are convolved with the functions $h_\varphi(-n)$ and $h_\psi(-n)$ and downsampled along the columns. This generates two subimages; say W_L and W_H respectively, of reduced horizontal dimensions (by a factor of 2). The generated subimage W_L contains low frequency vertical information, while W_H contains high frequency vertical information. Each column of both subimages is then convolved with $h_\varphi(-m)$ and $h_\psi(-m)$ and downsampled along the rows to give four subimages of quarter size with respect to the original matrix $W_\varphi\;(j+1,m,n)$.

The newly generated images are $W_\psi^D(j,m,n)$, $W_\psi^V(j,m,n)$, $W_\psi^H(j,m,n)$ and $W_\varphi\,(j,m,n)$ of resolution HH, HL, LH and LL respectively based on the low(L) and high(H) frequency contents. The sizes of all four matrices thus computed are same and each is one fourth of the size of $W_\varphi\,(j+1,m,n),$ used for the next iteration. The process starts at scale J and computes intermediate coefficients for every scale j ≥ j_0.

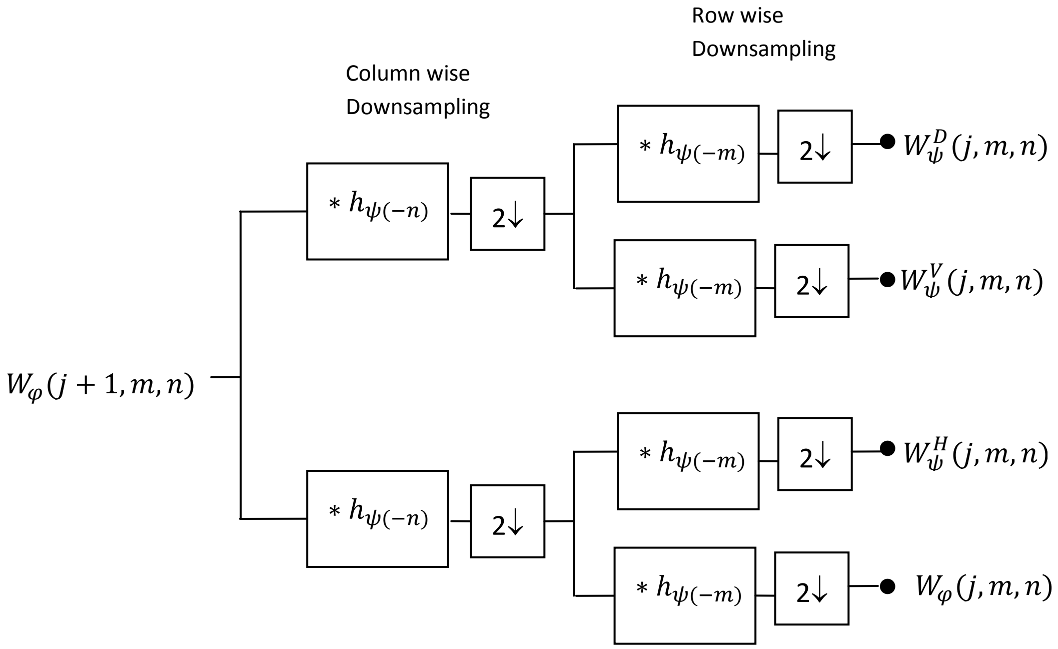


Fig. 3.5. Analysis Filter Bank

It is observed that the approximation coefficients, representing low frequency at level 3 i.e. LL(3) perform better than those at level 1 or 2, i.e. LL(2) or LL(1) coefficients, in face recognition domain. Also the detailed coefficients are less useful as compared to the approximation coefficients in face recognition.

3.3.Feature Selection Techniques

Feature Selection algorithms attempt to select the most informative and discriminative features from the given set of features. Few of the irrelevant features which contribute to the poor performance of various face recognition algorithms are discarded. The problem of feature selection refers to finding an optimal subset of features which produce maximum recognition accuracy. Finding the best subset of features is possible using exhaustive search (Brute Force Method) which is computationally very expensive and is exponential in time. Therefore, in Machine Learning systems, heuristics such as GA, PSO, FA etc. are used, which are computationally fast and provide approximately best subsets of features.

The Feature Selection algorithms are categorized in two classes based on their evaluation methods. The algorithms which are independent of the learning algorithms fall under *Filter approach*. Filter approaches usually rank the features based on their relevance and are computationally efficient. Some ranking methods are based on finding correlation of the features and the target. The limitation of filter approaches is that they do not produce optimal set of features. The algorithms that select features on the basis of their overall contribution in accuracy of the recognition process, fall under *Wrapper approach.*

Conventional feature selection methods use ranking, similarity measures, and various search methods such as gradient descent, hill climbing etc. while evolutionary approaches use methods such as Genetic Algorithm, Particle Swarm Optimization, Ant Colony Optimization etc. to find optimal solutions in polynomial time.

3.3.1. Conventional Feature Selection Approaches

Some of the wrapper approach based techniques are : Greedy search based Sequential Forward Selection (SFS), Sequential Backward Selection (SBS) and "plus-l-take-away-r" method, all of which attempt to select a subset on the basis of classification performance of the features selected or removed one by one to reach the near optimal combination. Other conventional feature selection techniques also use feature similarity and Cardinal sparse partial least square based feature selection techniques *(Zhang et al, 2014; Tran et al, 2014)*. The major disadvantage of conventional feature selection techniques is that if a feature is rejected once, it has no chance of inclusion in the remaining trials, while the left out feature could have contributed significantly in combination with other features.

3.3.2. Evolutionary Feature Selection Approaches

Evolutionary Approaches rely on finding optimal solution using mechanisms inspired by biological phenomena such as reproduction, mutation, selection, recombination, synchronization, movements etc. Producing genetically best offspring (child) from a pool of available chromosomes, synchronizing flights using swarm intelligence of cooperating or catching prey or finding another firefly based on the flashing behavior of fireflies rely on a natural evolution process. Evolutionary Algorithms (EA) are Computational Intelligence techniques mainly used for solving optimization problems. The most popular evolutionary algorithms used in feature selection are Genetic Algorithms(GA) and Particle Swarm Optimization (PSO) techniques. The feature selection process using GA and PSO techniques is presented in sections 3.3.2.1 and 3.3.2.2.

3.3.2.1. Genetic Algorithm based Feature Selection Technique

Genetic Algorithm(GA) is a metaheuristic method to guide a heuristic search and is inspired by the natural evolution such as inheritance, selection, crossover and mutation. Genetic Algorithm is a population based approach and is based on the survival of the fittest. Genetic algorithms are used for optimization problems where the optimizing function's fitness value increases over generations and the fittest solution is achieved by applying crossover and mutation operators to fittest parents.

The problem domain is known in terms of evaluating the fitness function for each possible solution represented as Chromosome or Gene. A chromosome is a string of numeric values representing possible parameter values and their overall effect is evaluated as its fitness. If a chromosome has large fitness value, then it is considered for evolution of next generation and if it has very less fitness value , then the chromosome is discarded.

In feature selection problem, use of Genetic algorithm has been reported by *Liu and Wang (2008); Ming Li (2010) and Vignolo et al (2012).* The features from the initial pool of features are selected randomly to form a chromosome. The approach of feature selection is Wrapper based and uses the classifier performance in terms of recognition accuracy for fitness function evaluation. Two operators known as crossover and mutation are applied to the chromosomes to obtain new generation chromosome representing better subset of features resulting in better classification or recognition accuracy. If the number of chromosomes with fitness value greater than a specified threshold in generation *t* is *p* and if the crossover probability is p_c , then $p_c \times p$ chromosomes are selected as parents and these participate pair wise in crossover to produce pair of offsprings of the new generation. Also, if the mutation probability is p_m then $p_m \times p$ chromosomes undergo the mutation process. If new chromosome is fit to be included, it is included in the new generation. In short, crossover can be understood as exploration process while mutation is exploitation of the neighborhood.

Let us consider a 4×4 window of the image features consisting of initial pool of 16 features [Fig.3.6(a)]. These features form a feature vector [Fig.3.6(d)] using layout represented in Fig.3.6(c) using the encoding scheme given in Fig.3.6(b). The feature vector size is 16 in the given example, therefore 16 random real valued numbers β_i : i = 1,2,..16 in the interval [0 1] are generated to fill the entries 0 or 1 in the chromosome [Fig.3.6(e)] depending on whether $\beta_i < T$ or $\beta_i \geq T$, where T is a threshold and is taken as 0.5.

The feature subset represented by chromosomes shown in Figures 3.6 (e) and (f) are {23, 29, 5, 30, 12, 52, 2, 15, 11, 13} and {29, 2, 5, 30, 12, 25, 52, 2, 15} respectively. The classification accuracy is normally used as fitness function of each chromosome and the features selected as above are used to compute the face recognition accuracy which serves as fitness value of the corresponding chromosome. The chromosomes which represent features contributing to better recognition accuracy are selected as candidates for crossover to produce new offspring. Roulette wheel is one of the commonly used methods to select the chromosomes to be treated as fittest parents. The fitness values of all eligible chromosomes (with fitness greater than a threshold) are added and a random number decides the fittest parent because of its high favorable chances due to its individual large fitness value. A single point *crossover* between two chromosomes is shown in Figure 3.7(a,b). The new generation represents different subsets of features and may produce solutions better than the previous generation chromosomes. The challenge lies in selecting the number and locations of crossover points so as to get optimal subset of features producing best classification accuracy. Crossover simulates a jump of the feasible solution from one point to another in the hyperdimensional space where the size of chromosome can be taken as the dimension of the solution space. This helps in exploring the entire search space heuristically while a neighborhood is exploited by applying a mutation operator on the fittest chromosome.

23	29	10	2
5	30	12	25
52	16	9	2
15	11	12	13

(a)

c1	c2	c3	c4
c5	c6	c7	c8
c9	c10	c11	c12
c13	c14	c15	c16

(b)

c1	**c2**	**c3**	**c4**	**c5**	**c6**	**c7**	**c8**	**c9**	**c10**	**c11**	**c12**	**c13**	**c14**	**c15**	**c16**

(c)

23	**29**	**10**	**2**	**5**	**30**	**12**	**25**	**52**	**16**	**9**	**2**	**15**	**11**	**12**	**13**

(d)

1	**1**	**0**	**0**	**1**	**1**	**1**	**0**	**1**	**0**	**0**	**1**	**1**	**1**	**0**	**1**

(e)

0	**1**	**0**	**1**	**1**	**1**	**1**	**1**	**1**	**0**	**0**	**1**	**1**	**0**	**0**	**0**

(f)

Fig. 3.6 : Genetic Encoding of Face Image (a) 4×4 window of 16 features as the initial pool of features (b) coding of features (c) codes as a vector (d) Feature vector (e)and (f) Two different Chromosomes with 1's representing inclusion of the corresponding features

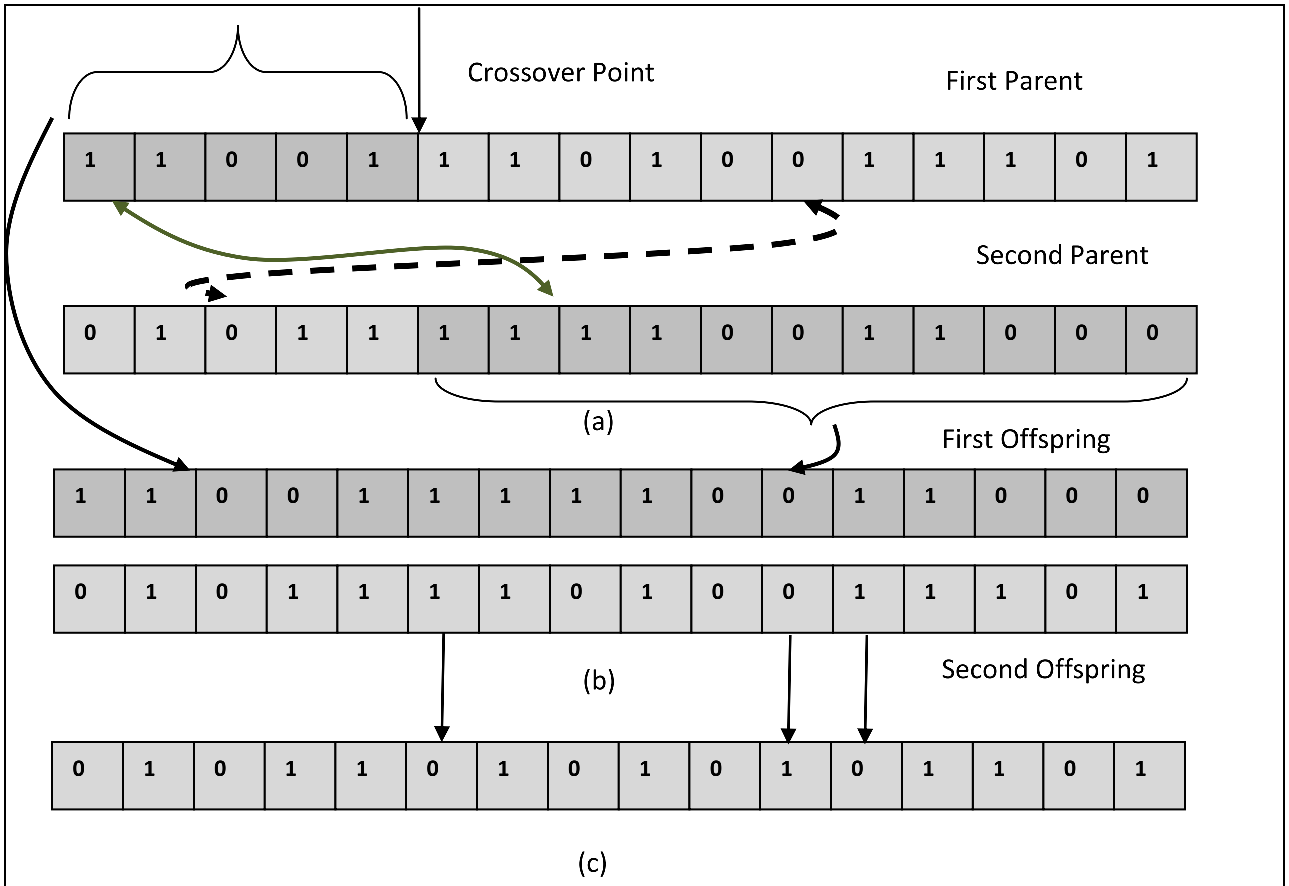


Fig. 3.7 :Crossover and Mutation (a) Previous Generation Parent Chromosomes selected for Crossover (b) Two new off springs generated by crossover using one crossover point. (c) Mutation applied at three places in second offspring

Mutation probability decides how many values will be mutated for one chromosome and the binary value 1 is deliberately made 0 indicating rejection of the corresponding feature while the binary value 0 is deliberately made 1 indicating inclusion of the feature accordingly [Fig.3.7(c)]. Which values will be mutated in a chromosome is randomized and are selected according to the proportionate number guided by the mutation probability.

3.3.2.2. Particle Swarm Optimization based Feature Selection Technique

Feature Selection problem in face recognition domain using Particle Swarm Optimization (PSO) has been addressed by *Cheng et al (2011)* and *Ajit Krisshna et al (2014)*. PSO technique is a meta heuristic population based stochastic optimization technique which is inspired by the flocking behavior of birds and fish. Their collective behavior in moving together in the sky has inspired optimization and search solutions in the field of Evolutionary Computing. The feasible solutions for the given optimization problem are represented as particles which keep moving in the search space. The collective behavior of these particles is used to find the global best (*gbest*) solution, i.e. the solution with maximum fitness value among the moving particles. The particle corresponding to maximum fitness at each iteration is used as guiding heuristic for *exploration* of the search space. Also each particle sees local solutions while it moves from its original position to the current position, and maintains locally obtained best solution (*lbest*). Local best solution becomes the guiding heuristic for *exploitation* of the neighborhood. In essence, both global and local best solutions are used for movement of the particles using appropriate weights given to each. The search for an optimal solution is guided by two parameters - the global best (knowledge gained by the whole population) and local best (particle's own memory).

Each particle (say i^{th} particle) is considered as a point x_i described by position vector <x_{i1} , x_{i2}, ..., x_{id}> in a d-dimensional space where *d* is initial number of features from which relevant features are to be selected. Each particle (say i^{th} particle) moves with a velocity V_i which is computed at each iteration as given by.

$$V_i^{(new)} = w \times V_i^{(old)} + c_1 \times rand_1 \times (lbest_i - x_i) + c_2 \times rand_2 \times (gbest_i - x_i) \quad (3.17)$$

where w, c_1 and c_2 are the acceleration constants, $rand_1$ and $rand_2$ are the random numbers between 0 and 1. If c_1 and c_2 are very low, then the particles wander in the space away from the optimal solution for long, and if these constants are very large, then the particles movement is very fast and may lead to premature convergence to a sub-optimal solution. The new position of the particle is given by:

$$x_i^{(new)} = x_i^{(old)} + V_i^{(new)} \qquad (3.18)$$

Where $V_i^{(new)}$ is the new velocity obtained using (3.17). The particles move in the hyperspace in search of better solutions guided by the global best and their local best seen so far. Algorithm converges when the swarms cannot move or have reached at position of global best, representing the selected features.

3.4. Firefly Algorithm

The firefly algorithm was originally proposed by Xin-She Yang and was inspired by the natural fireflies seen around in the tropical summer *(Yang, 2008)*. The natural fireflies have a bioluminescence substance called luciferin that causes a flashing light coming out of the fireflies. The firefly uses this light to communicate with other fireflies, or to attract another firefly for mating or to catch prey firefly of other species. Brightness of the flashing light seen by the other firefly is inversely proportional to the distance at which it is being seen.

The firefly algorithm uses the formulation of the feasible solution as a firefly. The firefly, which is simply a representation of the solution is unisex as gender does not get any mathematical formulation in this algorithm. The fireflies attract each other on the basis of the amount of attractiveness which is proportional to their brightness. The brightness of a firefly is associated with the value of the objective function corresponding to the firefly position in the d-dimensional landscape. A firefly moves towards the brighter firefly which represents a better solution in the search landscape. The amount of movement is proportional to the attractiveness and the attractiveness between the fireflies is inversely proportional to the distance between them. The parameter β, defining the attractiveness between the two fireflies at distance *r* apart, is expressed as

$$\beta = \beta_0 \mathrm{e}^{-\gamma \mathrm{r}^2} \qquad (3.19)$$

where β_0 is the 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. When a firefly is seen from a distance, it is assumed that some light emitted by it has been absorbed in the medium, causing the overall attractiveness of the firefly to reduce by some extent. The firefly F_i moves to a brighter firefly F_j by a distance and acquires a new position x_i defined by

$$x_i^{(new)} = x_i + \beta_0 e^{-\gamma r^2} \|x_j - x_i\| + \alpha \in_i \qquad (3.20)$$

The first term (x_i) on the right hand side is the previous position of the firefly. The second term is due to the attractiveness between the two fireflies F_i and F_j while the third term ($\alpha \in_i$) causes randomness in the movements of firefly. As firefly F_i moves to a new position, the brightness of it changes. The parameter α is known as the randomization parameter. Its value is chosen randomly from an interval [0,1]. The parameter $\in_i$ is the i^{th} component value of the vector of random values drawn from a uniform distribution or Gaussian distribution. The attractiveness at zero distance is considered maximum which is taken as unity i.e. $\beta_0 = 1$.

Fister et al (2013) illustrated that Firefly Algorithm (FA) is efficient due to various reasons such as (i) Fireflies have strong local attraction due to which they can organise themselves in subgroups easily and therefore can efficiently solve highly non-linear problems with multiple optimal solutions (ii) Unlike PSO, fireflies do not remember any local or global best and are capable of avoiding premature convergence. (iii) Fireflies can control their speed while moving in the hyper dimensional search space. In fact, FA can be considered as a generalization of PSO when the parameter $\gamma \to 0$ *(Yang, 2010).* It is observed that theoretical development of rate of convergence of metaheuristic algorithms such as PSO, ACO, FA etc. is difficult and is still at early stage and that FA converges for values of $\beta < 2$ (*Yang, 2011).* In the present work, emphasis has been on establishing the proposed technique for face recognition by experimentally demonstrating the convergence. A comparative description of the three evolutionary techniques Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Firefly Algorithm (FA) is presented in Table 3.1.

Table 3.1. Comparison of the Evolutionary Algorithms

Characteristic	GA	PSO	FA
Population	Chromosomes	Particles	Fireflies
Search Heuristic	Survival of Fittest	Swarm Behavior	Attractiveness
Exploration Process	Crossover	Global Best	Attractiveness
Exploitation Process	Mutation	Local Best	Randomization
Modality	Single	Single	Multiple
Convergence	Slowest	Slow	Fast
Overall merit in finding optimal solution	Weak: May get trapped to Local Optima	Moderate: Only single optimal solution can be obtained	Strong: Reaches Global, Multimodal Optimal Solutions and is very fast

3.5. Proposed Firefly Inspired Feature Selection (FIFS) Algorithm

In this section, a conceptual framework for firefly inspired feature selection is presented. Sections 3.5.1 and 3.5.2 present the firefly design and its movements in the search space. Section 3.5.3 defines the fitness function for the proposed fireflies. Sections 3.5.4 and 3.5.5 present the parameters and the pseudo code of the proposed FIFS algorithm.

3.5.1. *Firefly Design*

A firefly is mathematically defined as a d-dimensional point moving in the d-dimensional hyperspace, where d is the total number of features considered for dimensionality reduction. A face image is processed with feature extractors such as DCT or DWT. Logically more significant *d* features are initially placed in the feature vector <f_1, f_2, f_3, ...f_d> for all training face images. Of all such features, some may not contribute to the overall performance of the face recognition system and require to be removed. Inclusion of a feature f_k in the set of selected features depends on the position of firefly in the search space, where $1 \leq k \leq d$. To illustrate graphically, a three dimensional grid is shown in Fig. 3.8 which is visualized as a cube of size d×d×d with each side of size *d* and *d* = 3.

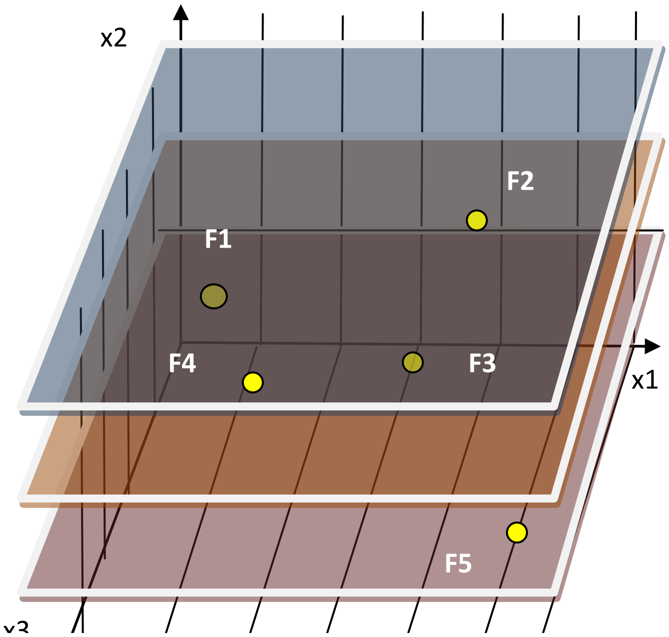


Fig. 3.8 : Three dimensional visualization of the search space and existence of fireflies

Five fireflies F_i : i = 1, 2, 3,4,5 are represented as three dimensional tuples, e.g. F_1: <1,1,1> represents a firefly F_1 with the selection of only one feature f_1, Firefly F_2: <4,2,4> represents the selection of two features f_2 and f_4, and so on. These fireflies represent selected features as above based on their position in the search space.

A number of fireflies (say Q) is generated randomly in the search space initially. A firefly (F) is mathematically represented as a vector $<p_1, p_2, p_3, ...p_d>$ such that the value of p_i is a real number in the interval [1, d] . A firefly is generated randomly by generating p_i $\forall$ i = 1,2,..,d randomly. The firefly keeps moving in the search space and acquires new positions using equation (3.20). Each value p_i $\forall$ i = 1,2,..,d is initially generated as

$$p_i = R(i,1) + [R(i,2) - R(i,1)] * \lambda \tag{3.21}$$

where λ is a random number drawn from the standard uniform distribution on the open interval (0, 1). The values of R(i,1) and R(i,2) are minimum and maximum feature values of i^{th} feature across all training feature vectors. Search space range R, a d×2 matrix, defines the boundary of the search space for the fireflies to move in. Let L be the total number of training images of all authorized persons in the given face database, and set T is a collection of all such training images T_u : u = 1,2,3,..,L. Each training image T_u is represented by the initial pool of d features $< f_1^u, f_2^u, f_3^u, \ldots, f_d^u >$, where the superscript u is indicative of the training image and the subscript (1 to d) is the index representing feature.

$$R = \begin{bmatrix} \min\limits_{\substack{1 \le u \le L \\ T_u \in T}} f_1^u & \max\limits_{\substack{1 \le u \le L \\ T_u \in T}} f_1^u \\ \min\limits_{\substack{1 \le u \le L \\ T_u \in T}} f_2^u & \max\limits_{\substack{1 \le u \le L \\ T_u \in T}} f_2^u \\ \vdots & \vdots \\ \min\limits_{\substack{1 \le u \le L \\ T_u \in T}} f_d^u & \max\limits_{\substack{1 \le u \le L \\ T_u \in T}} f_d^u \end{bmatrix} \tag{3.22}$$

The firefly F acquires new position $< p_1^{(new)}, p_2^{(new)}, p_3^{(new)}, \ldots., p_d^{(new)} >$ on getting attracted to another firefly G represented as vector $< q_1, q_2, q_3, ...q_{d>}$ and $p_i^{(new)}$ is computed as

$$p_i^{(new)} = p_i + \beta_0 e^{-\gamma r^2} \| p_i - q_i \| + \alpha \in_i \tag{3.23}$$

where the parameters β_0, γ, α and r are same as described in Section 3.4. Each firefly has an associated fitness value that indicates the significance of the firefly in given problem domain. If the fitness value is more, the firefly in the context of Feature Selection is said to represent the features which contribute to produce better recognition accuracy. A set of features, considered to be selected by the proposed design, represented by the firefly F = <p_1, p_2, p_3, ...p_d> are computed as

$$S_F = \{ f_{q_k} | \, q_k = round\left(p_k^{(new)}\right), p_k^{(new)} \neq \, p_m^{(new)} \, , k \neq m \, , \forall \, p_k^{(new)}, p_m^{(new)} \, \in \, F\} \qquad (3.24)$$

S_F represents a set of features selected using firefly F, which may or may not represent the most discriminative features. Equation (3.24) collects indices q_k which are obtained by first rounding real valued numbers $p_k^{(new)}$ and then by removing the duplicate indices. As an illustration, let a firefly Q represents its current position in a 15-dimensional space <1.2, 2.6, 4.1, 3.3, 2.4, 1.8, 2.9, 3.2, 1.1, 1.5, 2.2, 1.1, 4.9, 2.9, 0.8> . After rounding the real values as <1, 3, 4, 3, 2, 2, 3, 3, 1, 2, 2, 1, 5, 3, 1> and after removing the duplicates, set of features selected is <1, 3, 4, 2, 5>. The set of features S_Q thus represented by the firefly Q has the features selected as f_1, $f_{3,}$ $f_{4,}$ $f_{2,}$ $f_{5.}$ Same features are selected from all training images T_u: u = 1, 2, 3, .., L.

3.5.2. *Firefly Movements*

The firefly with less fitness value moves towards the brighter firefly with larger fitness value. The attractiveness between two fireflies is the motivation for the less brighter firefly to move to acquire new position. The entire process of movement of fireflies in the search space is based on the fact that the search space is visited using a heuristic so as to avoid exhaustive search. Fireflies explore the search space very fast and have the potential to obtain multiple optimal solutions. Fireflies follow two types of movements- one based on attractiveness (*heuristic Shift*) and the other random (*Random Shift*). The movement due to attractiveness pushes the firefly in the direction of the brighter firefly so that firefly explores the path towards the brighter firefly. The random movement of the firefly in its neighborhood provides an opportunity to explore the nearby positions for better solutions. While moving in the d-dimensional search space, firefly acquires different positions and its brightness changes. It is important to note that the fireflies are bound to move within the boundaries of the search space.

If any firefly tries to move by a distance that may take it beyond the boundaries of the search space, then the new position of the firefly in that move is made on the boundary itself.

3.5.3. *Fitness Function of a Firefly*

The fitness function of a firefly F is computed as the recognition accuracy of the test face images when the face recognition system is trained using the features represented by set S_F. The fitness value G^F of a firefly is defined by

$$G^F = \left(\frac{N_c}{N_t}\right) \times 100 \tag{3.25}$$

Where N_c is the total number of correctly recognized face images and N_t is the total number of test images. When the recognition accuracy is more with the selected features in S_F, the fitness value of the firefly F is large. It is therefore said to be a brighter firefly compared to the one that produces less recognition accuracy with its features. Each firefly, moving in the 'd'-dimensional space, acquires a new fitness value, based on its position in the space. A firefly position represents the selected features based on which the classifier works to obtain face recognition accuracy.

3.5.4. *Parameters of the proposed FIFS Algorithm*

The main parameters controlling the convergence of the proposed FIFS algorithm are as:

(a) Gamma (γ) : It is the Light Absorption Coefficient and controls the movement of the fireflies in the search space. The value of γ contributes to the size of the step moved by a firefly towards the brighter firefly. If the value of γ is very large, the speed of movement of the fireflies is slowed down and therefore the fireflies may not converge in given number of iterations. Also, a very small value of γ speeds up the movements of fireflies and may skip the global optimal solution.

(b) Number of Fireflies (Q): The fireflies must uniformly be distributed in the entire search space. All fireflies constitute a population and behave collectively. A very small number of fireflies may not reach the global optima in given number of iterations.

(c) Number of Iterations (N): A population of fireflies looks at each other pair wise. One firefly attracted towards the other moves towards it, and one complete round of such interactions among the whole population is termed as one iteration. For real time face recognition, the number of maximum iterations is kept low.

3.5.5. *Proposed Algorithm*

The algorithm takes as input the input feature vector from which the most discriminative features are to be selected. It also takes the input as total number of fireflies, algorithm parameters γ and β_0 , search space range R. A number of 'Q' fireflies F_1, F_2, ..., F_Q are generated randomly using (3.21). Each firefly (F_i) represents the subset of features S_{F_i}, computed using equation (3.24) and its fitness function G^{Fi} is computed using (3.25). The detailed algorithm is given in Fig.3.9 which obtains the set of optimal features $S_{F_{Opt}}$. Once the optimal features are selected, the classifier is trained with the selected features of all training images used. To demonstrate the capability of the proposed feature selection algorithm, a simple k-NN (k-Nearest Neighbor) classifier with k=1 is used for recognition of test images. The search space range R is computed using minimum and maximum values of features in each dimension from the training images of all authorized persons using equation (3.22). The vector $\alpha = <\alpha_1, \alpha_2,\alpha_3,..., \alpha_d>$ represents the randomization parameters α_i :i =1,2,3,...,d where each α_i is computed as follows

$$\alpha_i = \frac{abs(R(i,1) - R(i,2)}{10 * \sqrt{N \times d}} \tag{3.26}$$

The time complexity of the proposed algorithm is $O(NQ^2d)$, where *N* is the number of iterations, *Q* is the number of fireflies and *d* is the feature vector dimension.

INPUT: Search Space Range (R), parameters (γ), Randomization Parameters (α), Number of Fireflies (Q) and Maximum Number of Iterations (N).

```
FIFS( R, $\alpha$, Q, N, $\gamma$)
{
    Generate_Random_Fireflies(R,Q, F);
    for iter = 1: N
        for u = 1:Q
            Compute Fitness $G^{Fu}$ using (3.25)
            for v = 1:Q
                Compute Fitness $G^{Fv}$ using (3.25)
                if ($G^{Fu} < G^{Fv}$)
                    $posVec = |F_v - F_u|$;
                    $r = \|F_u - F_v\|$
                    $\beta = \beta_0 e^{-\gamma * r^2}$
                    D = zeros(d);
                    for i = 1:d
                        Random_Shift = $\alpha_i \times$ (rand-0.5);
                        Heuristic_Shift = $\beta$*posVec(i);
                        D(i) = Random_Shift+ Heuristic_Shift;
                    endfor
                    newPos = $F_u$ + D; //Vector Addition
                    for k = 1:d
                        if(newPos(k) > R(k,2))
                            newPos(p,q) = R(k,2);
                        elseif(newPos(k) < R(k,1))
                            newPos(k) = R(k,1);
                    endfor
                    $F_u$ =newPos;
                endif
            endfor
        endfor
    endfor
}
$F_{Opt}$ = Max ($G^{Fi}$)
```

OUTPUT: Set of features $S_{F_{Opt}}$using equation (3.24).

Fig.3.9. Proposed Firefly Inspired Feature Selection (FIFS) Algorithm

3.6. Results and Discussions

3.6.1. Face Databases Used

Four benchmarked face databases namely ORL (Olivetti Research Lab, also known as AT&T) *(www.cl.cam.ac.uk/Research/DTG/attarchive/pub/data/att_faces.zip)*, AR (*Martinez and Benavente, 1998; Martinez and Kak, 2001*), Yale *(http://vision.ucsd.edu/datasets/ yale_face_dataset_original/yalefaces.zip)* and LFW (Labeled Faces in the Wild) (*Huang et al, 2007b)* are used for evaluating the performance of the proposed algorithms *(http://vis-www.cs.umass.edu/lfw/).*

ORL face database consists of frontal images of 40 persons, capturing variations in pose, illumination and accessories like spectacles. There are 10 sample images of each person. The images have been taken against dark homogeneous background with accessories and facial expressions including open/closed eye, smiling/not smiling etc.

The *AR face database* is a large database with 26 face sample images of 126 persons. The database captures face images originally of sizes 768×576 with 24-bit grey level depth. The images are frontal and taken in two sessions separated by two weeks. The AR face database captures variations in illumination with light falling on human face from different directions (left lights on, right lights on and all side lights on). Four different expressions captured are: neutral face, smile, anger and scream. Occlusions are captured in two different ways as occlusion around eyes using goggles and occlusion below the nose covering mainly mouth and chin using scarf. Only 40 persons' face data is used in this study.

The *Yale face database* contains a total of 165 face images belonging to 15 individual persons. The images capture variations in illumination with lights from three different directions - left, right and center. Six different expression variations are also captured in this face database - normal, happy, sad, surprised, winking eyes and sleepy. Images with glasses and without glasses image samples are also provided in the database for each person. All 15 persons face data is used with 6 faces randomly selected for training. Figure 3.10 shows the sample face images of ORL, Yale and AR face databases.

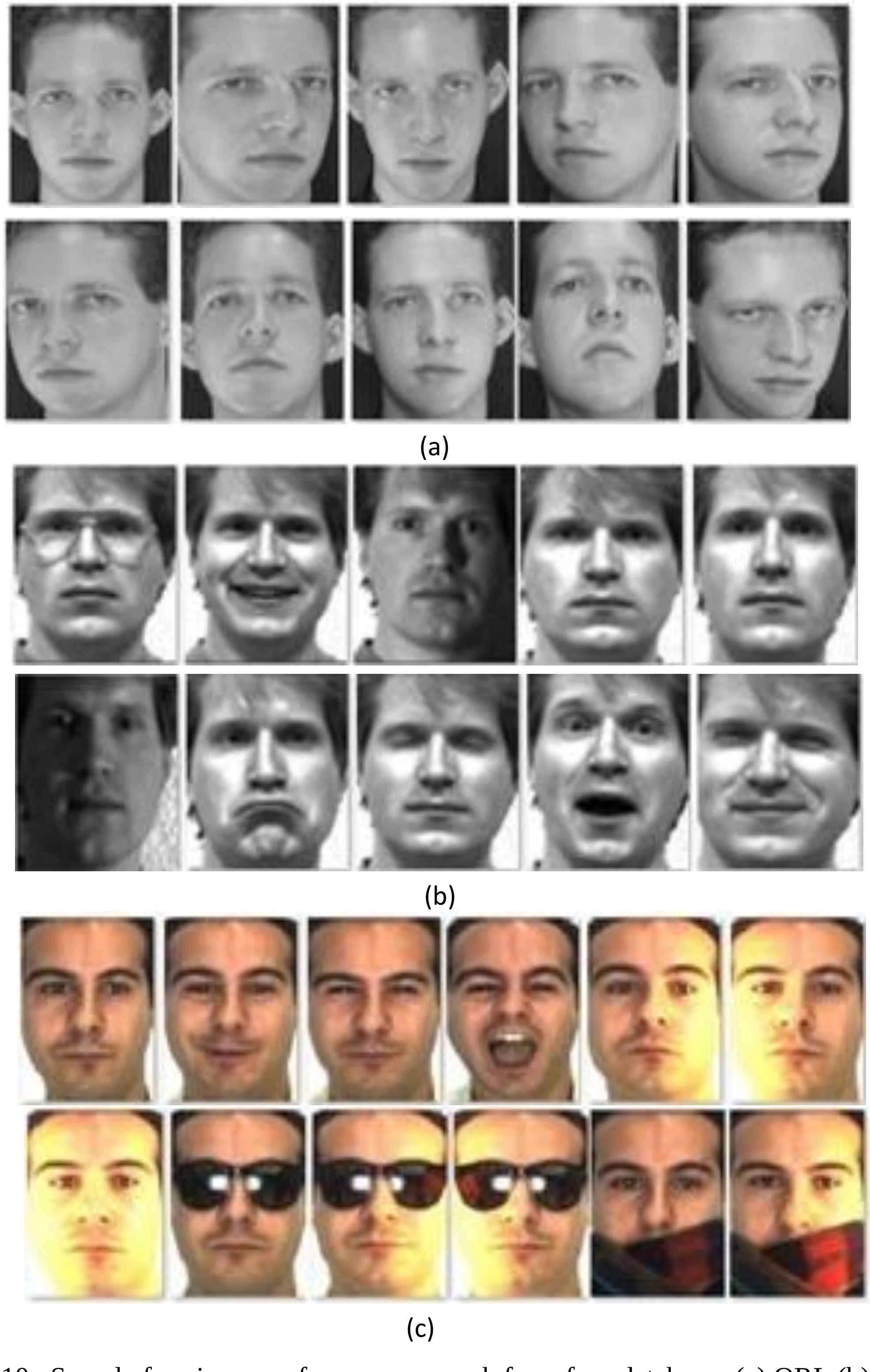
The *LFW face database* is the most recent face database and has about 15000 face images of the celebrities available on the world wide web. It has a collection of varying number of images in different poses, illumination conditions, expressions, occlusion etc. These variations are not systematic as they are in other face databases, therefore the LFW face database is considered suitable for algorithmic validations for unconstrained face images. Sample face images of LFW face database are shown in Figure 3.11. The details of these face databases and details of images used in all experiments in this work are mentioned in Table 3.2 and Table 3.3 respectively. Appendix contains all images from all four face databases used in this study.

Table 3.2. Details of Face Databases

Face Database	**Developed at**	**year**	**Number of Persons**	**Total face images per person**	**Size (in pixels)**	**Image Format**	**Grey / Color depth**	**background**	**Variations Captured during Face Image acquisition**				
									Pose	**Illumination**	**Emotions/ Expression**	**Accessories/ occlusion**	**Temporal**
ORL	AT&T laboratories, Cambridge, UK	1992 to 1994	40	10	92x 112	PNG	8-bit grey level	Dark, homogeneous	Upright, frontal (minor side poses)	minor	Eyes open/ closed, smiling/ not smiling	glasses	yes
Yale	Yale University	Information not available	15	11	243x 320	GIF	gray	Information not available	frontal	Center, left and right light	Normal, sad, happy, sleepy, surprised and wink	Glasses, no glasses	No
AR	Computer Vision Center at U.A.B.	1998	126	26	120x 165	BMP	24-bit depth	Information not available	frontal	Left, right and all side light	Neutral, smile, anger, scream	Sun glasses, scarf	yes
LFW	MIT	2007	5749	varying	250 x250	JPG	color	Natural	Unconstrained				

Table 3.3. 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 generated randomly**	**Remaining 5 of 1-10**
Yale	**Yes**	**92×112**	**15**	**6**	**4**	**Any 6 generated randomly**	**Remaining 4 of 1-10**
AR	**Yes**	**60×83**	**40**	**13**	**6**	**1-13 fixed**	**14, 15, 17, 19, 22, 26**
LFW	**Yes**	**92×112**	**10**	**8**	**8**	**Any 8 generated randomly**	**Remaining 8 of 1-16**



(a)

(b)

(c)

Fig. 3.10. Sample face images of one person each from face databases (a) ORL (b) Yale (c) AR Face Databases

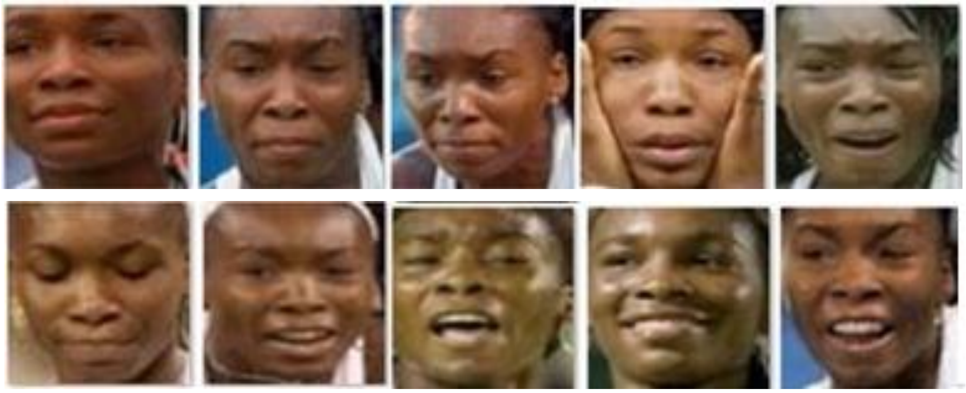
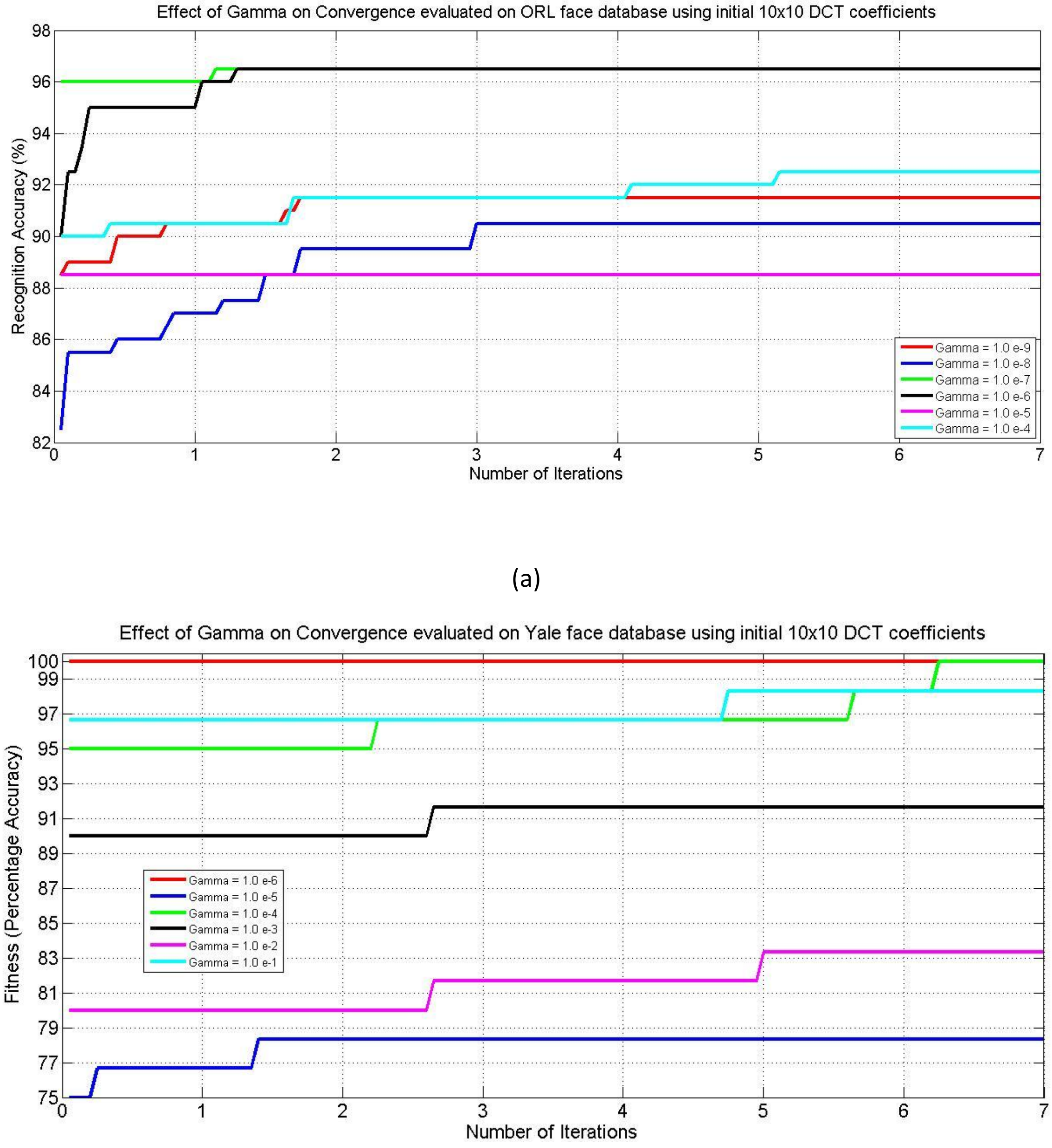


Fig. 3.11. Unconstrained Face Database LFW(Labeled Faces in the Wild)

3.6.2. Effect of Parameter Gamma (γ) on Algorithm Convergence

The effect of the algorithm parameter γ (gamma) on convergence is studied and number of iterations for convergence is achieved by running the program on Intel I5-2430M CPU (clock speed of 2.40 GHz and RAM of 8GB). Initial feature set is taken as the upper left corner square window of size 10×10 and firefly fitness is obtained using proposed FIFS technique. Fireflies move and the fitness of the brightest firefly changes with each round of iterations. In one round of iteration, one firefly interacts with all other fireflies in the population (say Q) and the fitness of the brightest firefly is plotted. When all Q fireflies complete their interactions with all fireflies in Q such rounds, one iteration is completed.

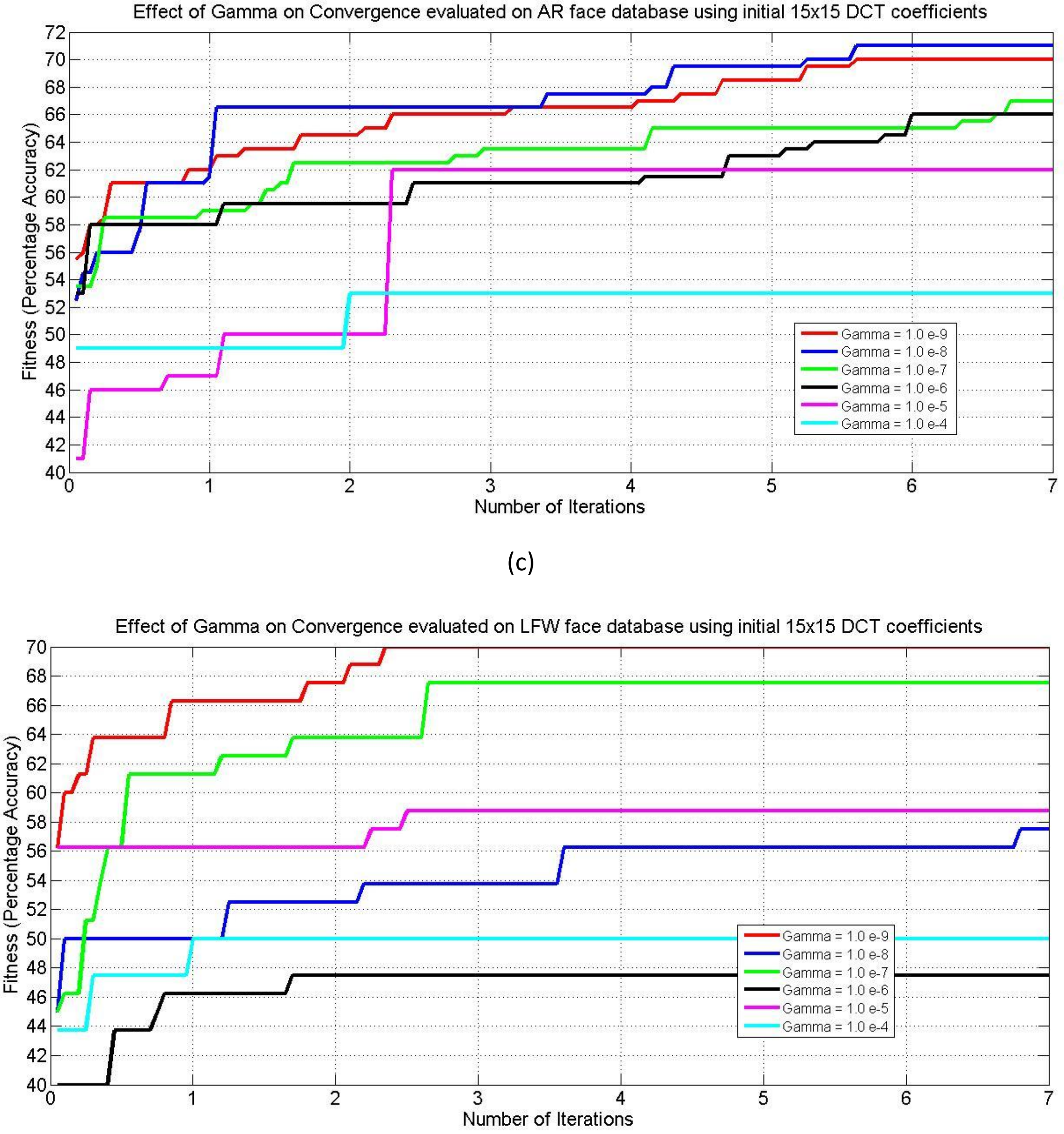
Figures 3.12(a) to 3.12(d) for the four benchmarked face databases namely ORL, Yale, AR and LFW respectively mainly demonstrate the algorithm convergence using different values of the parameter γ. It is observed that in only 7 iterations algorithm converges. The lowest value of γ displays slow convergence and it will not be sufficient to use only 7 iterations for best results. Therefore we chose values of γ in range which is neither too low nor too high. While the effect of γ is less visible in Figures 3.12 (a) and (b) due to less variations in the ORL and Yale Face Databases, the effect is visible in AR and LFW face databases in Fig. 3.12(c) and (d).



(a)

(b)

Fig.3.12 Effect of parameter γ on Recognition Accuracy on (a) ORL (b) Yale Face databases



(c)

(d)

Fig.3.12 (Continued) Effect of parameter γ on Recognition Accuracy on (c) AR (d) LFW Face databases

3.6.3. Effect of Parameter Gamma (γ) on Average Recognition Accuracy and Dimensionality Reduction

In this section, an effect of values of parameter γ on recognition accuracy and dimensionality reduction is investigated using four independent runs with the numbers of fireflies and iterations taken as 20 each. The initial features are taken as combination of 10×10 DCT features and LL(3) coefficients of Haar wavelets to showcase the effect of γ on average recognition accuracy and dimensionality reduction. Dimensionality reduction is computed as the percentage ratio of the difference of number of original features *d* and reduced number of features (equal to the size of the set S_F , i.e. $|S_F|$, denoted as s) and original number of features *d* given as

$$Dimensionality\ Reduction = \left(\frac{d-s}{d}\right) \times 100 \qquad (3.27)$$

Low values of γ such as 1×10^{-6} produce an average recognition accuracy of 94.375% and 43.10% reduction in dimensionality when the proposed FIFS technique was evaluated on ORL face database. It is observed that all 20 fireflies converged to the same fitness value in 20 iterations using this value of γ. The standard deviations from the average accuracy for all the three observations are very less i.e. 3.07, 2.63 and 1.47 over 4 runs of the same experiment with same set of parameters.

The performance of the proposed algorithm on Yale Face database has been obtained using γ as 1×10^{-7} . The Average recognition accuracy for Yale face database is 99.16% with a standard deviation of only 0.96. The dimensionality reduction obtained is 36.66% using γ as 1×10^{-7}. All 20 fireflies converge at the end of 20 iterations for both ORL and Yale face databases using γ as 1×10^{-6} and 1×10^{-7} respectively. Also, values of γ, greater than 10×10^{-6}, have limitation with respect to firefly convergence as this value of gamma slows down the speed of the fireflies.

The proposed FIFS technique when evaluated with the more complex face database AR, using γ value of 1×10^{-7}, produces an average recognition accuracy of 52.375% with a standard deviation of 2.98 and obtains 42.15% dimensionality reduction. Average recognition accuracy of 67.188%, standard deviation of 3.73 and the dimensionality reduction of 41.98% are achieved when FIFS was evaluated using the LFW face database. The results are tabulated in Table 3.4. It is observed that the proposed algorithm's performance on AR and LFW face databases is not at par with the performances of the ORL and Yale face databases. The reason for this is that AR and LFW have images with large variations and therefore require more number of features to represent the images as will be shown in section 3.6.5.

Table 3.4. Performance Evaluation of the Proposed FIFS Algorithm with respect to γ [Initial Features taken as DCT coefficients from upper left square window of size 10×10 and All coefficients of LL(3) of Haar Wavelets]

Face Database	**Gamma (× 10^{-6})**	**Average Recognition Accuracy(%) (over 4 runs)**	**Standard Deviation (Accuracy)**	**Average Number of Selected Features *s* (Max features *d*) ± std**	**Dimensionality Reduction (%)** $\left(\frac{d-s}{d}\right) \times 100$
ORL	0.1	93.75	3.07	158.25 (268) ± 6.8	40.95
	1	**94.375**	**2.63**	**152.5**(268) ± 12.61	**43.10**
	10	94.00	1.47	172.0 (268) ± 8.3	35.82
YALE	**0.1**	**99.16**	**0.96**	**169.75** (268) ± 7.68	**36.66**
	1	97.5	2.15	158.75 (268) ± 12.28	40.76
	10	98.75	0.83	172.5 (268) ± 4.43	35.63
AR	**0.1**	**52.375**	**2.98**	**108.75 (188)** ± 1.89	**42.15**
	1	50.875	2.36	93.00 (188) ± 11.28	50.53
	10	42.125	1.31	120.00 (188) ± 3.56	36.17
LFW	**0.1**	**67.188**	**3.73**	**155.50 (268)** ± 9.68	**41.98**
	1	58.125	9.16	139.00 (268) ± 37.26	48.13
	10	50.313	7.80	169.00 (268) ±6.68	36.94

In the above experiment the number of iterations is taken as 20 so as to give ample time for the fireflies to move in the search space to obtain optimal features. In next section, values of gamma (γ) resulting in best performance in terms of average recognition accuracy are used for individual face database to find the minimum number of iterations needed for convergence.

3.6.4. Effect of Number of Iterations on Algorithm Convergence and Recognition Accuracy

In this experiment, the effect of number of iterations on the convergence is investigated and the fitness function values are plotted against x-axis coordinates taken as points after one round of firefly interactions. One iteration comprises of number of such interactions equal to the number of fireflies.

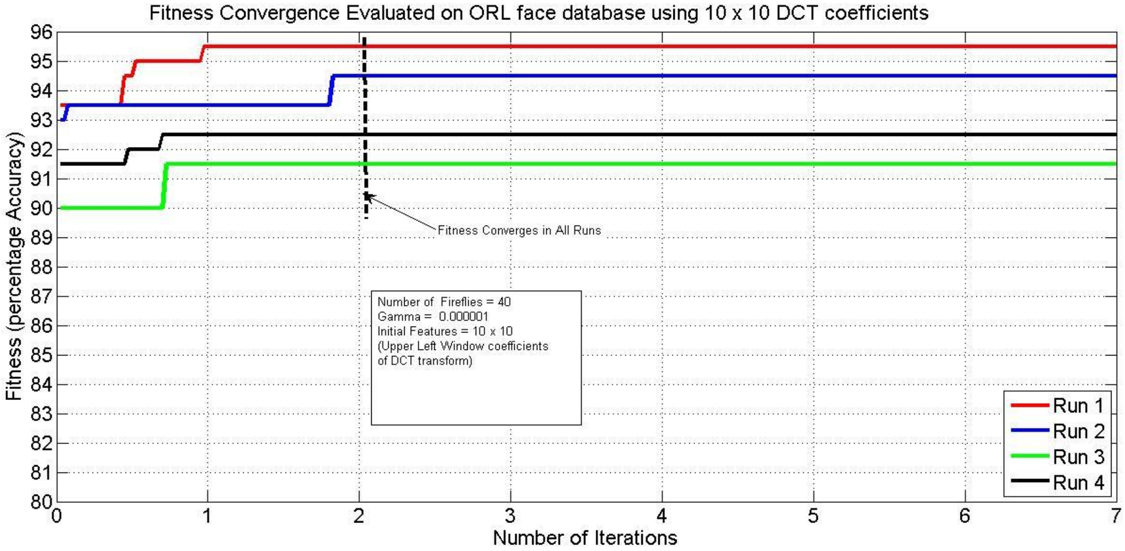


Fig. 3.13 Fitness Convergence trend in four independent Runs using ORL Face Database.

The fireflies kept move according to the proposed FIFS algorithm in search of a better solution and in this process, one of the fireflies obtained a position with better fitness as compared to the maximum fitness obtained in previous round of interactions. The position of the brightest firefly is used to get the set of selected features using (3.24) in each round of interaction of a firefly with all other fireflies in the population (say epoch). These features were used to compute the fitness values using (3.25) corresponding to the brightest firefly which is plotted in each epoch among all 40 fireflies in the population. One ***iteration*** therefore has 40 such ***epochs*** in this experiment. It is observed that the fitness value reaches its maximum in about 1-2 iterations and becomes stable. Different sets of randomly generated training images and test images were used in four independent runs in which the fitness values range in 91.5% to 95.5%. The parameter γ used for algorithm convergence is taken as 1×10^{-6}. All fireflies converge to their optimal fitness in maximum 2-3 iterations. The selected features after running the proposed FIFS algorithm are shown in Table 3.5. White pixels (represented as 1) are only 62 out of 100 initial features (a 10x10 window of DCT coefficients), thereby resulting in dimensionality reduction of 38% (represented as black pixels with value 0) (where dimensionality reduction is computed using equation 3.27).

Table 3.5 Selected Features from a 10×10 upper left window of the DCT coefficients experimented with ORL face database

1	**0**	**0**	**1**	**1**	**0**	**0**	**0**	**0**	**0**
1	**0**	**1**	**1**	**1**	**0**	**1**	**1**	**1**	**1**
1	**1**	**1**	**1**	**0**	**0**	**1**	**1**	**0**	**0**
1	**0**	**1**	**1**	**0**	**0**	**1**	**0**	**1**	**0**
1	**1**	**1**	**1**	**0**	**1**	**1**	**0**	**1**	**1**
0	**1**	**0**	**1**	**1**	**1**	**0**	**0**	**1**	**1**
1	**0**	**1**	**1**	**1**	**1**	**0**	**0**	**1**	**0**
0	**1**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**1**
1	**1**	**1**	**1**	**1**	**1**	**0**	**1**	**1**	**1**
0	**1**	**1**	**0**	**1**	**0**	**0**	**0**	**1**	**1**

Similarly, the convergence is investigated on the Yale Face database in Fig.3.14. The fitness values in all four independent runs start with a low fitness value as 90.0%, 88.3%, 68.0% and 90.0% which converged to optimal fitness as 96.6%, 95.5%, 81.6% and 100% respectively resulting in iteration count of 1, 1, 1, and 4 respectively.

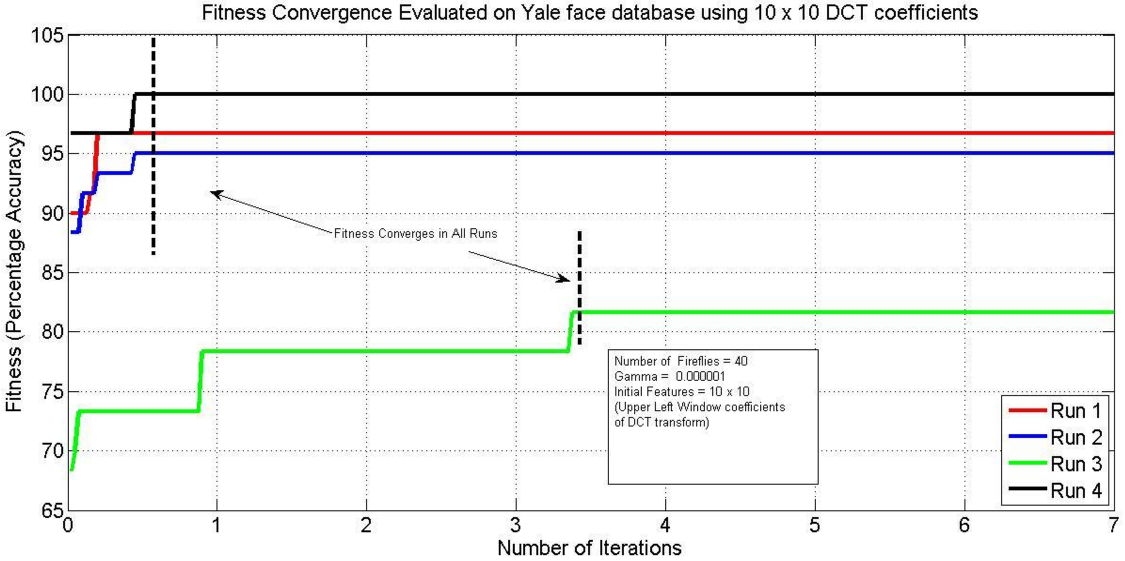


Fig. 3.14 Fitness Convergence trend in four independent Runs using Yale Face Database.

The proposed FIFS algorithm displays the similar behaviour when tested with other two complex face databases namely AR and LFW. The algorithm convergence tested on AR face databases is displayed in Fig. 3.15 where the initial feature window was taken as 10 by 10 to maintain uniformity across all face databases to display the convergence trend. In actual, the number of features for AR database to perform well is quite large (more than 200) when selected. The fitness values in all four independent runs start with a low fitness respectively as 55.0%, 52.5%, 52.5% and 47.5% which converged to optimal fitness as 63.5%, 58.0%, 63.0% and 59.5% respectively in maximum 7 iterations. With LFW database, the algorithm converges in maximum 5 iterations while the fitness values in all four independent runs start with fitness as low as 57.5%, 55.0%, 60.0% and 52.5% converging to optimal fitness as 58.75%, 67.50%, 77.50% and 66.25% respectively. The observation of the above experiment is that the proposed algorithm converges in maximum 7 iterations for all face databases used.

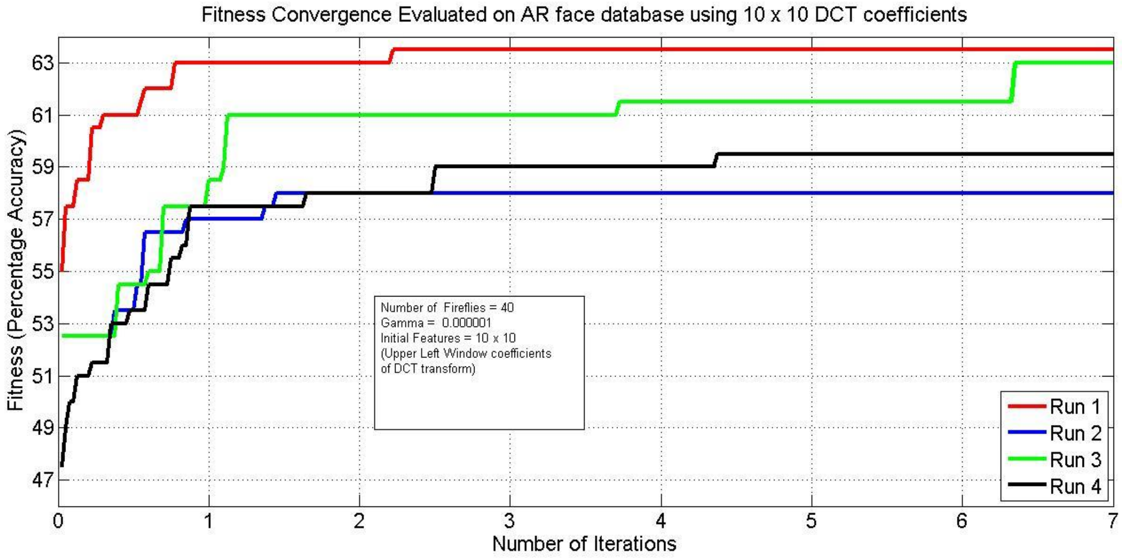


Fig. 3.15 Fitness Convergence trend in four independent Runs using AR Face Database.

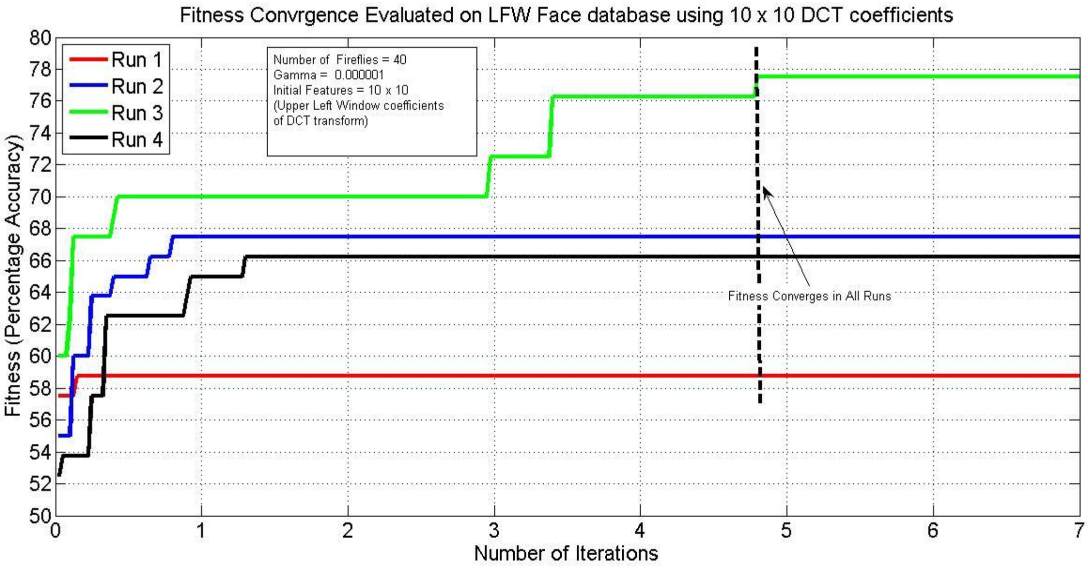


Fig. 3.16 Fitness Convergence trend in four independent Runs using LFW Face Database.

3.6.5. Effect of Feature Extraction Method on Recognition accuracy

The number of fireflies used in this experiment is 20 and number of iterations is 7. Experiments were conducted to assess the significance of feature extraction methods over four independent runs using value of γ as 1×10^{-6} for ORL face database and γ as 1×10^{-7} for all the other three face databases i.e. Yale, AR and LFW in separate experiments. Five methods were tested and performance in terms of average number of features selected, average recognition accuracy and standard deviation is computed for all four face databases shown in Table 3.6.

Table 3.6 Performance of Feature Extraction Methods evaluated on ORL face database

Method	Initial Features (Total Number of Initial Features)	Average Number of Selected Features ± std	Average Recognition Accuracy (%) ± std
DCT	10x10 Upper left window (100)	**63.50± 3.32**	**93.75 ± 1.94**
LL(1) [Haar wavelets]	10x10 Upper left window (100)	61.50 ± 1.00	59.00 ± 6.52
LL(3) [Haar wavelets]	10x10 Upper left window (100)	59.00 ± 4.24	89.625 ± 1.89
LL(3) [Haar wavelets]	all features of LL(3) DWT (168)	99.00 ± 7.62	91.75 ± 4.52
DCT + LL(3)	10x10 Upper left window (100) of DCT and all features of LL(3) DWT (168)	170 ± 2.16	93.50 ± 0.82

It is observed that DCT alone and DCT with LL(3) features of the Haar Wavelet produce recognition accuracies as 93.75% ± 1.94 and 93.50% ± 0.82 respectively when evaluated with the ORL face database. These features outperform LL(1) features of the Haar Wavelets whose performance is only 59.00% ± 6.52, while LL(3) features of Haar wavelet perform better giving average recognition accuracy of 89.625% ± 1.89 and 91.75% ± 4.52 (taken 10 ×10 window and all features respectively) as compared to LL(1), but worse than DCT and DCT+LL(3). On one hand when recognition accuracies of DCT alone and DCT+LL(3) combination of features are comparable (93.75% and 93.50% respectively), the average number of features used by DCT+LL(3) is very high (170 ± 2.16) compared to DCT (63.50 ± 3.32). Hence, the performance of DCT features taken alone is the best among all combinations above.

The performance of the proposed FIFS method using DCT is compared with different other feature extraction methods reported in literature [Table 3.7], and it is seen to give best performance.

Table 3.7. Comparison of Recognition Accuracy of the Proposed FIFS Algorithm using DCT with some of the existing literature using variety of other feature extraction methods on ORL Face Database

Authors (Year)	Year	Feature Extraction Method	Reported Recognition Accuracy
Li et al	2007	Wavelet Transform	93.0%
Chen and Zhang	2007	Wavelet Energy Entropy	90.5%
Chang and Hsu	2008	Gabor Wavelet Transform	82.0%
Ayyavoo and Jayasudha	2013	Discrete Wavelet Transform	87.0%
Darestani et al	2013	Curvelet Transform and PCA	90.0%
Chelali and Djeradi	2014	Haar wavelets	91.0%
Proposed FIFS Algorithm	**-**	**FIFS using DCT**	**93.75%**

Tables 3.8 and 3.9 show the performances of different feature extraction techniques used with the proposed FIFS algorithm for Yale and AR face databases respectively. In Table 3.8, the

DCT feature window of size 10x10 produces the best accuracy and the proposed FIFS algorithm selects only 61.25 features on average (in 4 runs).

Table 3.8 Performance of Feature Extraction Methods evaluated on Yale face database

Method	Initial Features (Total Number of Initial Features)	Average Number of Selected Features ± Std	Average Recognition Accuracy (%) ± std
DCT	10x10 Upper left window (100)	**61.50 ± 1.91**	**95.42 ± 4.59**
LL(1) [Haar wavelets]	10x10 Upper left window (100)	62.50 ± 3.51	50.42 ± 15.48
LL(3) [Haar wavelets]	10x10 Upper left window (100)	64.75 ± 2.22	74.17 ± 7.39
LL(3) [Haar wavelets]	all features of LL(3) DWT (168)	106.25 ± 1.89	94.17 ± 5.00
DCT + LL(3)	10x10 Upper left window (100) of DCT and all features of LL(3) DWT (168)	163.00 ± 3.37	92.91 ± 4.17

Table 3.9 also displays the potential of DCT features producing the best recognition accuracies for AR face database. The 10x10 window of LL(3) is not possible as the size of the LL(3) transformed image of an AR face image is 11x8 each. AR face database has many variations and needs a large number of features for a comparable performance, while here the number of features used has been kept low to demonstrate the potential of the feature extraction techniques relative to each other.

Table 3.9 Performance of Feature Extraction Methods evaluated on AR face database

Method	Initial Features (Total Number of Initial Features)	Average Number of Selected Features ± Std	Average Recognition Accuracy (%) ± std
DCT	10x10 Upper left window (100)	**61.50 ± 2.52**	**60.75 ± 4.21**
LL(1) [Haar wavelets]	10x10 Upper left window (100)	62.75 ± 2.22	18.13 ±2.46
LL(3) [Haar wavelets]	all features of LL(3) DWT (88)	52.00 ± 3.37	37.37 ± 1.93
DCT + LL(3)	10x10 Upper left window (100) of DCT and all features of LL(3) DWT (88)	112.00 ± 1.15	48.00 ± 2.80

The effect of number of features on the performance has also been investigated using AR face database. The number of initial DCT features were varied from window sizes of upper left DCT coefficients as 10×10, 15×15, 20×20, 25×25, 30×30, while keeping the number of Haar Wavelet level three approximation coefficients i.e. LL(3) features fixed as 88. The investigation of effect of size of initial feature set on the average recognition accuracy evaluated on AR face database is shown in Fig. 3.17. The experiments were performed using number of fireflies as 20 and number of iterations as 20. The number of selected features along with the number of features taken initially (mentioned within the parentheses pair) are mentioned using an arrow for each set of features in Fig. 3.17. It is observed that the best performance on the AR face database is obtained using DCT features of size 25×25 along with 88 features of LL(3) coefficients of Haar Wavelets, making a total number of initial features as 713 of which only 380 features were selected by the proposed FIFS algorithm resulting in average recognition accuracy as 74.75% ± 1.32 and dimensionality reduction of 46.7% computed as $\left(\frac{713-380}{713}\right) \times 100$.

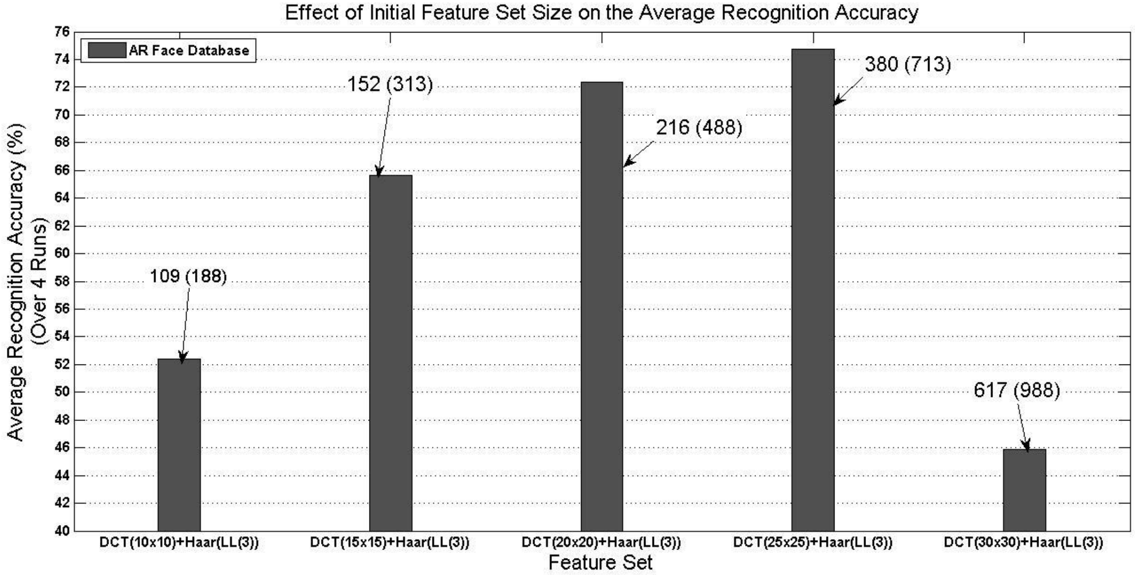


Fig. 3.17. Effect of size of Initial Feature Set on Average Recognition Accuracy evaluated on AR Face Database using FIFS

When evaluated on LFW face database, the proposed FIFS algorithm demonstrates results as shown in Table 3.10

Table 3.10 Performance of Feature Extraction Methods evaluated on LFW face database

Method	Initial Features (Total Number of Initial Features)	Average Number of Selected Features ± Std	Average Recognition Accuracy (%) ± std
DCT	10x10 Upper left window (100)	**62.25 ± 3.10**	**59.38 ± 6.33**
LL(1) [Haar wavelets]	10x10 Upper left window (100)	61.5 ± 1.29	28.43 ± 6.16
LL(3) [Haar wavelets]	10x10 Upper left window (100	59.25 ± 3.95	53.75 ± 2.70
LL(3) [Haar wavelets]	all features of LL(3) DWT (168)	104.25 ± 4.35	55.31 ± 8.86
DCT + LL(3)	10x10 Upper left window (100) of DCT and all features of LL(3) DWT (88)	145.75 ± 9.54	55.63 ± 6.50

It is observed through Tables 3.6 to 3.10 that DCT features outperform the other feature extraction methods such as Haar Wavelet based DWT features taken from resolution levels 1 and 3 in terms of the average recognition accuracy.

3.6.6. Comparative Performance of GA and PSO based Feature Selection with the Proposed FIFS Algorithm

The performance of Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) based feature Selection methods were compared with that of the proposed Firefly Inspired Feature Selection (FIFS) technique. Feature selection techniques using GA, PSO and FIFS were tested on the same four benchmarked face databases ORL, Yale, AR and LFW. The average recognition accuracy was obtained over 4 independent runs. It is established through various experiments discussed in section 3.6.1 that DCT coefficients contribute to better performance as compared to other Haar Wavelets based features evaluated on the four face databases used. The DCT coefficients from the upper left square window of size 10×10 were taken as the initial pool of features. A weak classifier namely k-Nearest Neighbor (k=1) was used to demonstrate the strength of the proposed FIFS in comparison to the other two evolutionary methods. It is observed that PSO and Genetic Algorithms converge in much larger number of iterations as has also been illustrated by Xin She Yang in his paper *(Yang, 2013).* To compare the three algorithms, the number of iterations is kept same for all techniques and is 7. It is recalled that one iteration amounts to one round of interaction in the whole population of evolutionary objects such as particles, chromosomes and fireflies in PSO, Genetic algorithm and the proposed FIFS based feature selection methods respectively. Table 3.11 compares the performance of feature selection methods based on PSO , Genetic algorithm and proposed FIFS experimented in this study on all four face databases. The parameters used are mentioned in first row of the table.

The major achievements are in the form of algorithm convergence and average recognition accuracy. The proposed FIFS algorithm outperforms the feature selection method based on PSO and Genetic Algorithms.

PSO and GA based feature selection methods were experimented with 140 epochs or generations $\left(number\ of\ iterations = \frac{number\ of\ epochs}{swarm\ size}\right)$ while the proposed FIFS technique was experimented with 7 iterations, which is equal to 140 rounds (7 × number of fireflies) of interaction between the fireflies. Table 3.11 presents performance of GA, PSO and the proposed FIFS based feature selection in terms of average recognition accuracy and the standard deviation. It is observed that the overall average time for the PSO, GA and FIFS based feature selection is in the ratio **7.8 : 4.8 : 1** approximately.

Table 3.11 Comparative Performance of Three Evolutionary Algorithms for feature selection

		PSO based Feature Selection	Genetic Algorithm based Feature Selection	Proposed Firefly Inspired Feature Selection (FIFS)
Parameters		C1 = 2.0	$p_c = 0.7$	$\beta_0 = 1.0$
		C2 = 2.0	$p_m = 0.001$	$\gamma = 0.000001$ (ORL)
		W = 1.0	number of crossover Points = 2	$\gamma = 0.0000001$ (Yale, AR, LFW)
		Number of particles = 20	Number of Chromosomes = 20	Number of Fireflies = 20
		Iterations = 7 (140 Epoch)	Iterations = 7 (140 Epochs)	Iterations = 7 (140 rounds)
Average Recognition Accuracy ± std	ORL	90.75 ± 1.89	79.25 ± 4.29	**96.12 ± 1.93**
	Yale	96.25 ± 2.50	70.42 ± 21.62	**96.25 ± 3.44**
	AR	57.13 ± 1.49	19.00 ± 1.47	**58.13 ± 5.95**
	LFW	56.56 ± 1.88	22.81 ± 5.34	**63.12 ± 1.61**

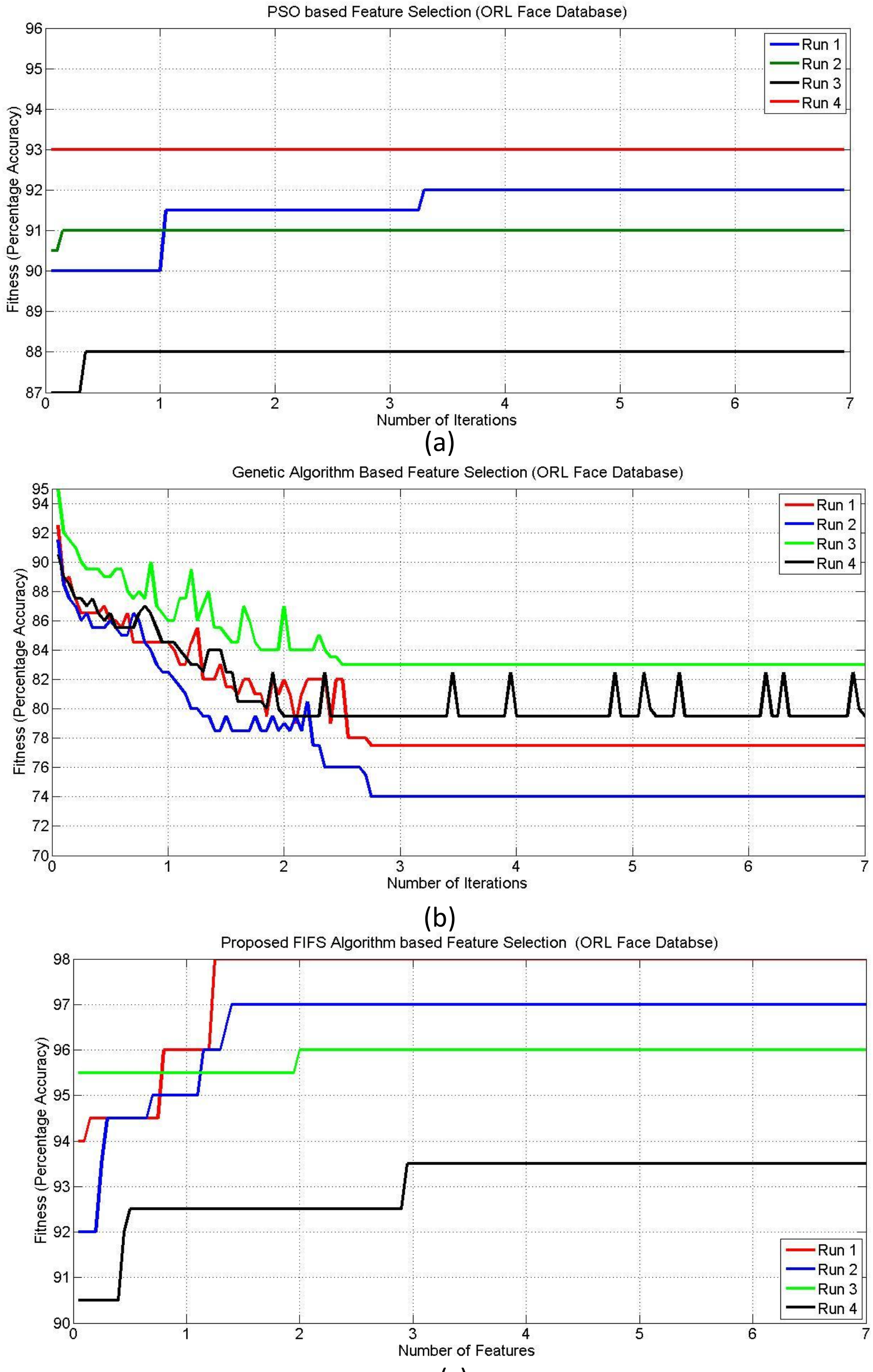
Table 3.11 clearly demonstrates the superiority of the proposed FIFS algorithm in terms of the average recognition accuracy. Convergence behavior of the three algorithms for the four

face databases used in the experiments are shown in Figures 3.18 to 3.21. The maximum number of iterations shown is 7 in all experiments demonstrating the *convergence* trend on ORL, Yale, AR and LFW Face Databases. *Convergence* in the present context of Feature Selection is defined as stable maximum fitness value.

A trend of premature convergence is visible in Figures 3.18(a), 3.19(a), 3.20(a) and 3.21(a) where PSO based Feature Selection method is used for the four face databases respectively, while Figures 3.18(b), 3.19(b), 3.20(b) and 3.21(b) using GA based Feature Selection method display a trend of deteriorating performance. Figures 3.18(c), 3.19(c), 3.20(c) and 3.21(c) display a trend of slow and steady convergence in all runs resulting in better Average Recognition accuracy as reported in Table 3.11.

The three algorithms for feature selection, were tested with five sets of features taken initially from the upper left corner of the DCT transform images of the training face images. The initial features were taken as 20, 40, 60, 80 and 100 respectively to demonstrate the strength of the proposed FIFS algorithm in high dimensional feature space ($d \geq 60$) in comparison to the other two evolutionary techniques of GA and PSO. The average recognition accuracy was obtained in four independent runs of the same experiment. The relative performance in terms of the average recognition accuracy of the three techniques is represented in Figure 3.22(a)-(d) for the four face databases ORL, Yale, AR and LFW respectively.

The face databases AR and LFW capture huge variations in illumination, pose, expression and occlusion, therefore require a large number of features for more accurate face recognition. In Figures 3.22(a) through (d), is is observed that the proposed FIFS outperforms the GA based feature selection and demonstrates its superiority over PSO based feature selection method for larger sets of features, particularly for features 60, 80 and 100. The individual average recognition accuracies corresponding to the Fig.3.22 are shown in Table 3.12.with values in bold where FIFS displays superiority.



(a)

(b)

(c)

Fig. 3.18. Algorithm Convergence of Feature Selection Methods evaluated on ORL Face Database using (a) Particle Swarm Optimization (b) Genetic Algorithm (c) Proposed Firefly Inspired Feature Selection (FIFS) Algorithm

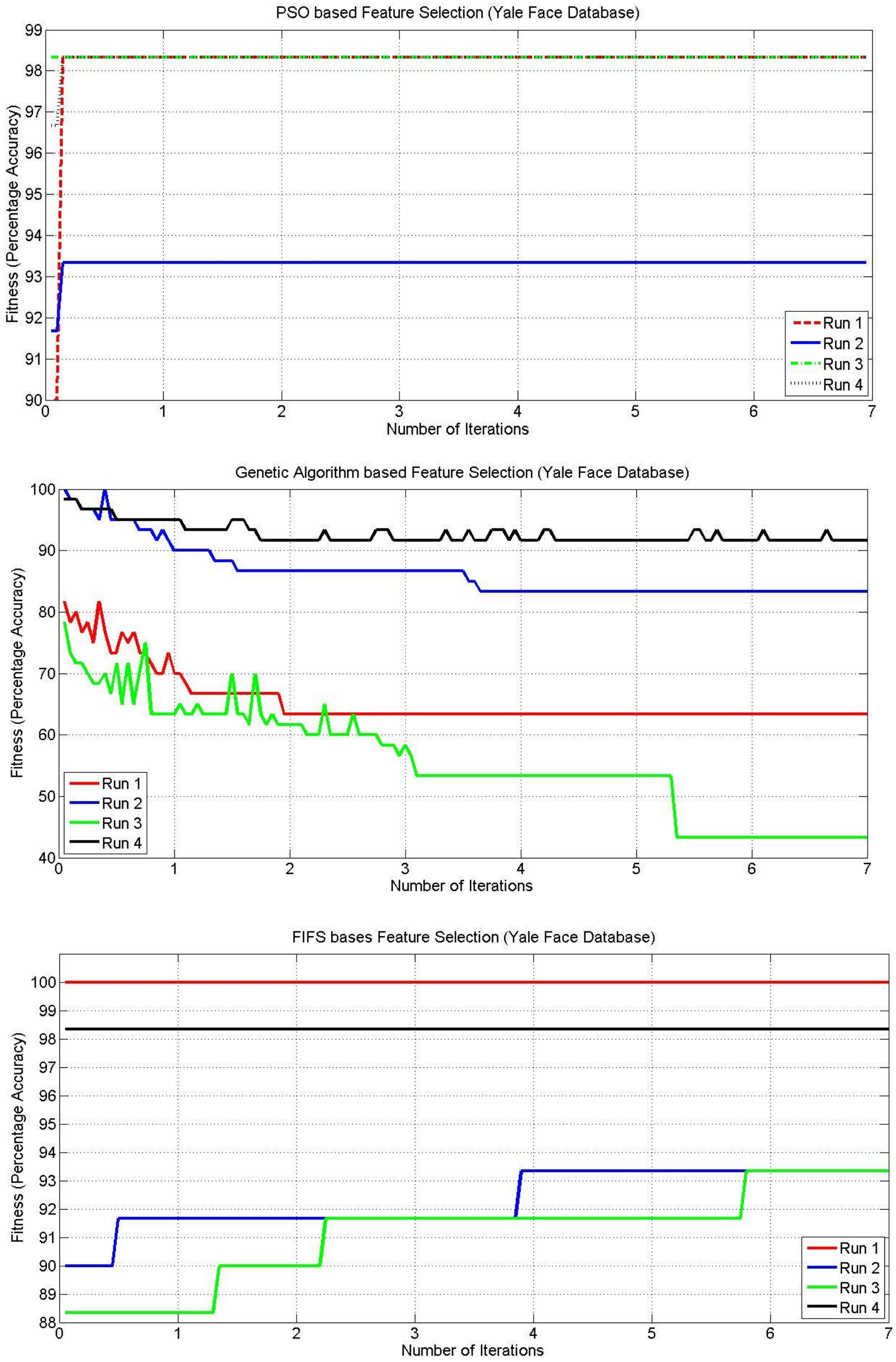


Fig. 3.19. Algorithm Convergence of Feature Selection Methods evaluated on Yale Face Database using (a) Particle Swarm Optimization (b) Genetic Algorithm (c) Proposed Firefly Inspired Feature Selection (FIFS) Algorithm

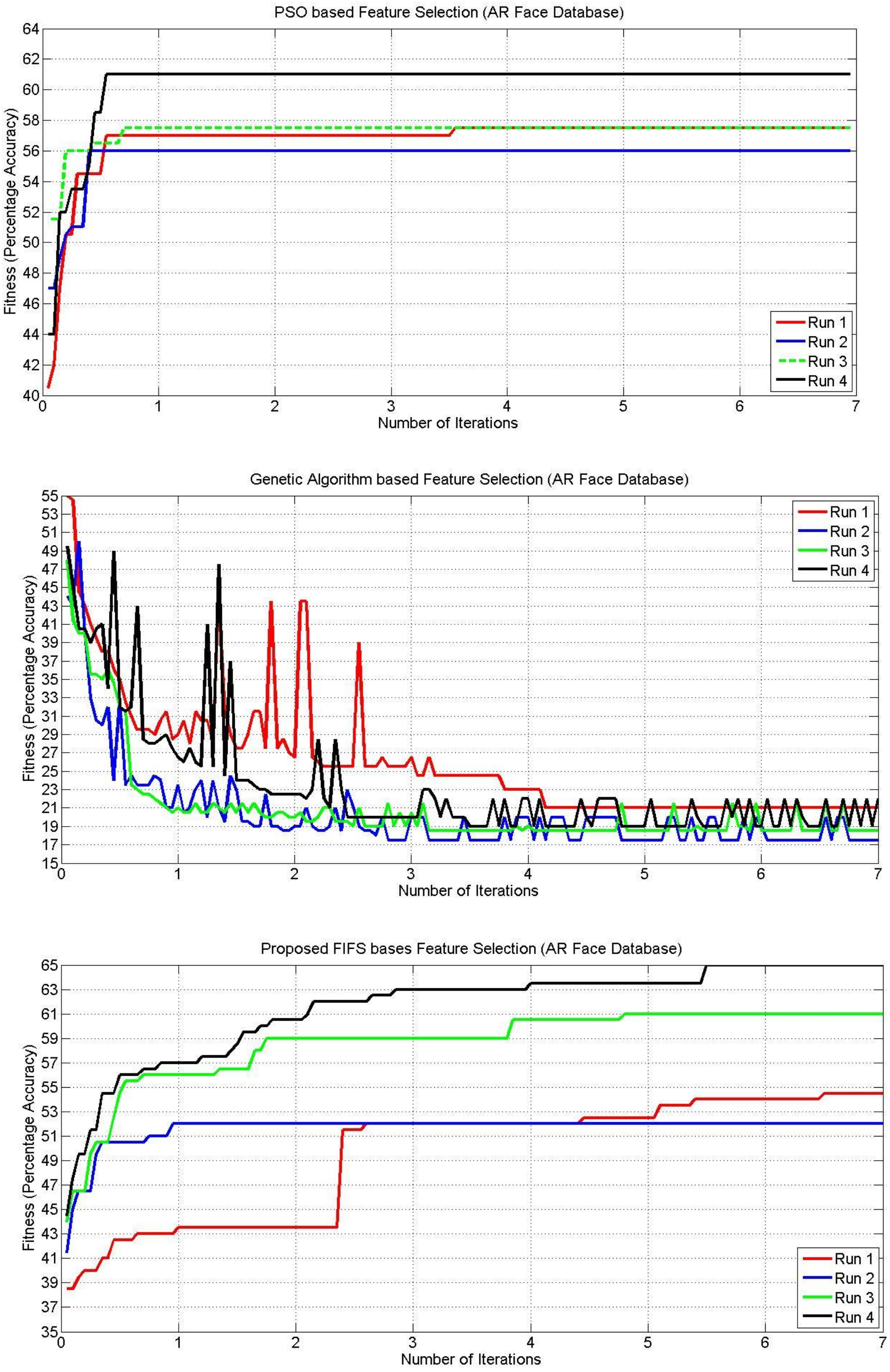


Fig. 3.20. Algorithm Convergence of Feature Selection Methods evaluated on AR Face Database using (a) Particle Swarm Optimization (b) Genetic Algorithm (c) Proposed Firefly Inspired Feature Selection (FIFS) Algorithm

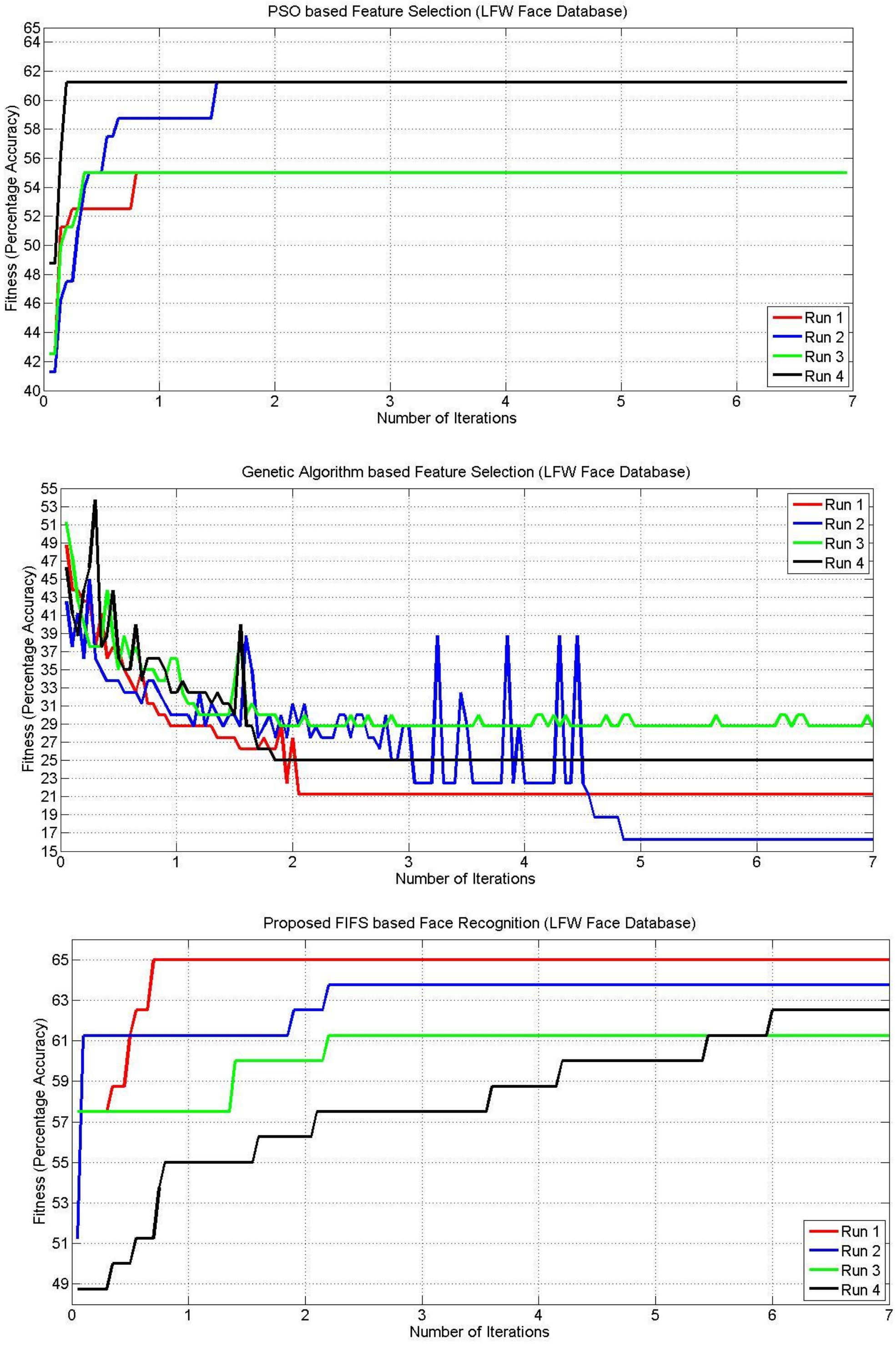
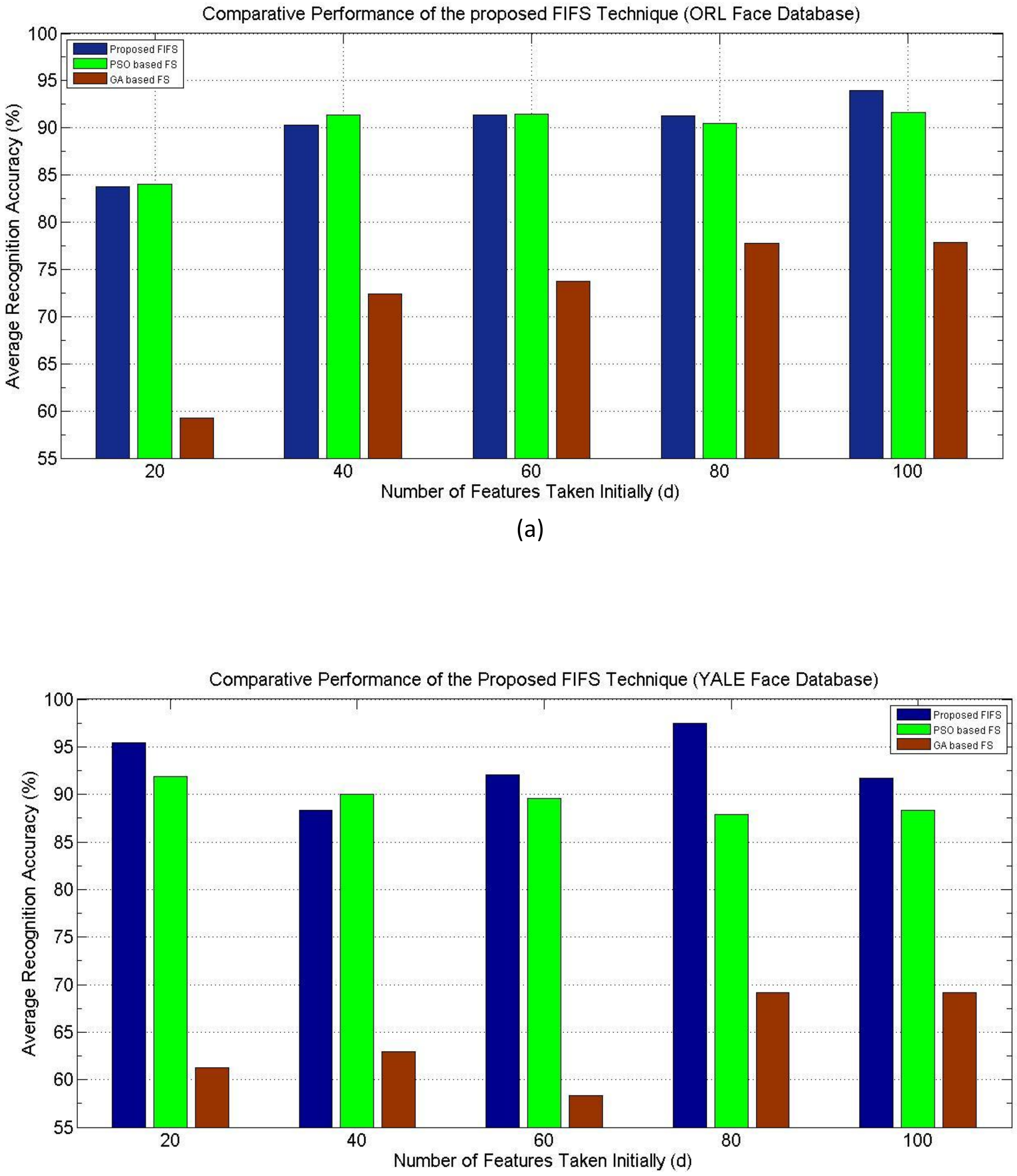


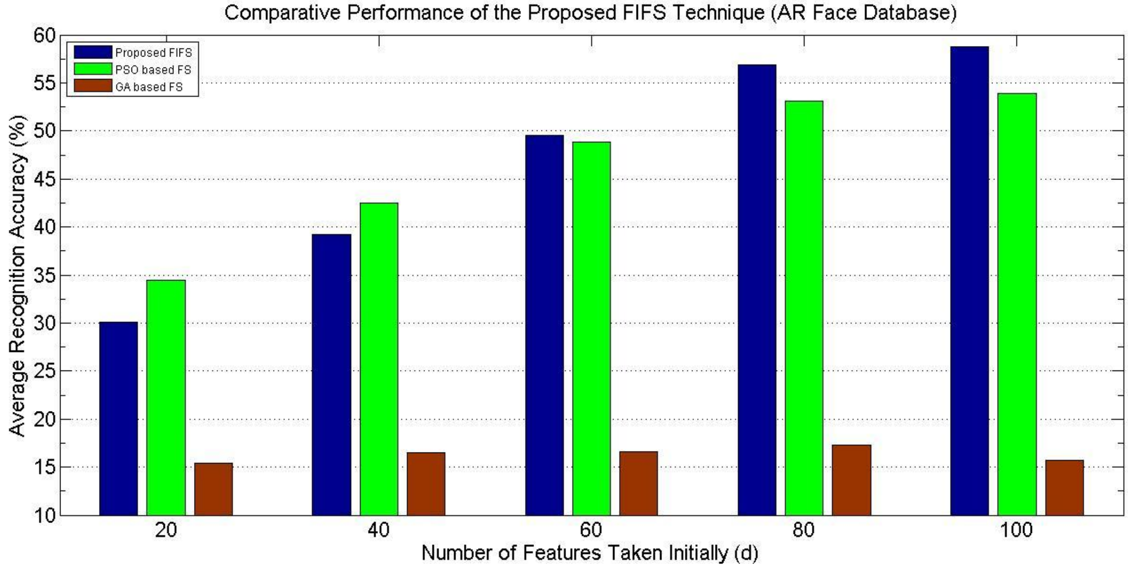
Fig. 3.21. Algorithm Convergence of Feature Selection Methods evaluated on LFW Face Database using (a) Particle Swarm Optimization (b) Genetic Algorithm (c) Proposed Firefly Inspired Feature Selection (FIFS) Algorithm



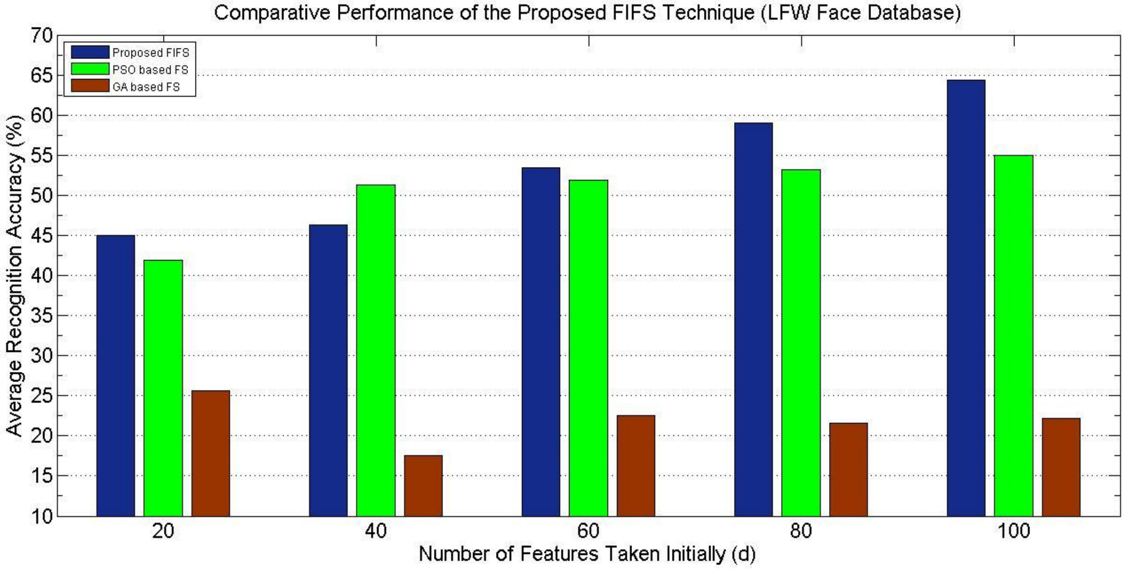
(a)

(b)

Fig. 3.22 Comparative Performance of the Proposed FIFS Technique compared with the feature Selection using Genetic Algorithm and Particle Swarm Optimization techniques evaluated on (a) ORL (b) Yale Face Database



(a)



(b)

Fig. 3.22 (Continued).Comparative Performance of the Proposed FIFS Technique compared with the feature Selection using Genetic Algorithm and Particle Swarm Optimization techniques evaluated on (c)AR (d) LFW Face Database

Table 3.12 Comparative Performance of the proposed FIFS algorithm with GA and PSO based feature selection methods

Face Database	Number of Features Taken Initially	Average Recognition Accuracy (%) using PSO based Feature Selection (in 4 runs)	Average Recognition Accuracy (%) using GA based Feature Selection (in 4 runs)	Average Recognition Accuracy (%) using FIFS based Feature Selection (in 4 runs)
ORL	20	84.00	59.25	83.75
	40	91.38	72.38	90.25
	60	91.50	73.75	91.38
	80	90.50	77.75	**91.25**
	100	91.63	77.88	**94.00**
Yale	20	91.92	61.25	**95.42**
	40	90.0	62.92	88.33
	60	89.58	58.33	**92.08**
	80	87.92	69.17	**97.50**
	100	88.33	69.17	**91.67**
AR	20	34.50	15.38	30.13
	40	42.50	16.50	39.25
	60	48.88	16.63	**49.50**
	80	53.13	17.25	**56.88**
	100	53.88	15.75	**58.75**
LFW	20	41.88	25.60	**45.00**
	40	51.25	17.50	46.25
	60	51.88	22.50	**53.44**
	80	53.13	21.56	**59.06**
	100	55.00	22.19	**64.38**

3.6.7. Comparison of the Proposed FIFS algorithm with some of the Existing Work

The performance of the proposed FIFS algorithm is compared with Genetic Algorithm(GA), Particle Swarm Optimization (PSO) and Binary Particle Swarm Algorithm (BPSO) based feature selection techniques as reported in literature on ORL face database [Table 3.13]. Table 3.14 gives the parameters used by the methods listed in Table 3.13. The proposed algorithm displays superiority over all three techniques in terms of recognition accuracy obtained using only 5 training images while the existing studies mentioned in Table 3.13 use 5 and 7 training images. The number of runs is 10 in this experiment to simulate the similar number as is used in other studies. GA and BPSO methods use 100 iterations, PSO uses 30 iterations while the proposed FIFS uses only 7 iterations to achieve the best performance in terms of average recognition accuracy. Also the swarm size used by the proposed FIFS algorithm is only 20, the minimum of 50, 30 and 100 used by GA, BPSO and PSO based feature selection methods respectively. It is therefore established that the proposed Firefly Inspired approach is superior as compared to feature selection methods based on Genetic Algorithm, Binary PSO or PSO.

Table 3.13. Comparison of various evolutionary algorithms based feature selection with the proposed FIFS technique for ORL Face Database

Authors	**Year**	**Approach**	**Classifier Used**	**Number of Training Images used**	**Swarm Size**	**Iterations**	**Runs**	**Recognition Accuracy**
Liu and Wang	2008	GA	k-NN	5	50	100	10	90.50%
Cheng et al	2011	BPSO	Nearest Neighbor	5	30	100	10	93.25%
Darestani et al	2013	PSO	MLP	7	100	30	-	90.00%
Proposed (FIFS)	**-**	**FA**	**Nearest Neighbor**	**5**	**20**	**7**	**10**	**93.85% (std = 1.84)**

Table 3.14: Parameters specific to the methods used for comparison in Table 3.13

Genetic Algorithm based Feature Selection *(Liu and Wang, 2008)*	Binary Particle Swarm Optimization (BPSO) based Feature Selection*(Cheng et al, 2011)*	PSO based Feature Selection *(Darestani et al, 2013)*	Proposed Firefly Inspired Algorithm (FIFS)

$P_c = 0.65$
$P_m = 0.004$
	c1 = 2
	c2 = 2
	W = 0.6
		c1 = 2
		c2 = 2
			γ = 0.000001
			β_0 = 1.0

It is observed that on Yale Face database also, the proposed FIFS algorithm outperforms two techniques of feature selection reported by *Xiao-Dong and Wei (2012) and Li et al (2009)* as is shown in Table 3.15.

Table 3.15. Comparison of some of the existing algorithms based feature selection with the proposed FIFS technique for Yale Face Database

Authors	**Year**	**Approach**	**Classifier Used**	**Recognition Accuracy**	**Number of Training Images used**
Xiao-Dong and Wei	2012	Within-class distance and between class distance	Nearest Neighbor	89.12	6
Li et al	2009	Multi-channel Dimension Reduction Scheme (MDRS)	Minimal Distance Classifier	85.33	5
Proposed (FIFS)	**-**	**FA**	**Nearest Neighbor**	**96.25**	**6**

The proposed FIFS method outperforms two methods: one by *Xiao-Dong and Wei (2012)* based on Within-class distance and between class distance and the other by *Li et al (2009)* based on Multi-channel Dimension Reduction Scheme (MDRS).

3.7. Conclusion

A novel method of feature selection based on firefly algorithm has been proposed in this work. A firefly is designed to represent a collection of features while fitness of a firefly is defined as the recognition accuracy. A population of fireflies moves in the search space representing a pool of features. The proposed algorithm is validated using four benchmarked face databases, namely, ORL, Yale, AR and LFW. In this chapter, the algorithm parameter γ is investigated for algorithm convergence, Average recognition accuracy and the dimensionality reduction on all four face databases. It is observed that the parameter γ, which controls the speed of the moving fireflies efficiently such that fireflies converge to an optimal feature set in given number of iterations resulting in best recognition accuracies, is of the order of 10^{-6} for all face databases. The finding is novel and significant due to the complex nature of hyper dimensional face recognition problem. The number of iterations in which fireflies converge to produce optimal feature set contributing to the best recognition accuracies is 7 which is significantly low as compared to the other evolutionary algorithms such as PSO and Genetic Algorithm. In this study, it is investigated that DCT features perform better than the Haar wavelets, therefore in Neural Network based Classifier Design proposed and validated in Chapters 4 and 5, DCT features are used. To the end, the performance of the proposed FIFS algorithm is compared with the feature selection methods based on GA and PSO. It is observed that the performance of the proposed algorithm outperforms GA and PSO based feature selection methods. The average recognition accuracy evaluated on ORL, Yale, AR and LFW are 96.12% (± 1.93), 96.25% (± 3.44), 58.13% (± 5.95) and 63.12% (± 1.61) evaluated in four independent runs of simulation. The proposed technique displays about 40% dimensionality reduction and is computationally very efficient as it uses only 7 iterations to converge while a maximum of 20 iterations can be used for complex face databases such as AR and LFW face databases. The strength of the proposed FIFS is in its fast convergence, small computation time, with better recognition accuracy as compared to other evolutionary methods in feature selection.

Chapter 4

Evolutionary Center Selection for Radial Basis Function Neural Network for High Speed Face Recognition

4.1. Introduction

Radial basis function neural networks (RBFNN) have been used for various face processing tasks such as face recognition, facial expression recognition, gender, age determination etc. RBFNN consists of three layers - input layer, hidden layer and the output layer. The design of hidden layer is of utmost importance as it captures the underlying structure of the training data. The parameters of the hidden layer are: center of the radial basis function (RBF) units, spread of the basis function, number of RBF units and the choice of the basis function. The performance of RBFNN depends on the structure of the hidden layer and the weights connecting hidden layer and output layers. Radial basis functions at the hidden layer of the RBFNN perform nonlinear mapping of the input space to the linearly separable hyperspace. RBFNN possesses good approximation capability, has less computational requirements and fast learning speed as compared to other architectures such as Multilayer Feed Forward networks using Backpropagation algorithm.

In this chapter, we propose a firefly inspired evolutionary algorithm, named as FRBFNN, to produce centers and optimal number of RBF units in the design of hidden layer of RBFNN. The algorithm is adaptive as it learns the structure and grouping of the data on its own. The number of RBF units is not required to be known a priori. The proposed algorithm captures the most natural grouping of the high dimensional training face images based on their illumination, pose, expression, accessories qualitatively. The algorithm is validated using the benchmarked face databases namely ORL, AR, Yale and LFW. The conceptual framework for the proposed FRBFNN technique has been laid out in this chapter. The values of algorithm parameters γ, *number of fireflies* and *number of iterations* have been obtained experimentally for all four face databases. These parameters were obtained using convergence criteria for the proposed algorithm. The parameter γ is responsible for movements of fireflies and their convergence by controlling their speed.

4.2. Radial Basis Function Neural Network

Radial basis function neural networks (RBFNN) are feed forward type of neural networks. The architecture of RBFNN consists of three layers: input layer, hidden layer and output layer (Fig.4.1). Input layer receives the input and passes the information to the hidden layer. The hidden layer is responsible for mapping nonlinearly separable input feature vectors to a higher dimension space where the mapped input is linearly separable. The hidden layer consists of radial basis function (RBF) units, also called as hidden neurons. Each i^{th} unit has an associated center c_i, spread σ_i and the basis function ϕ_i. The basis functions ϕ_i are radial in the sense that these are functions of radial distance of input from the center of the radial basis function unit.

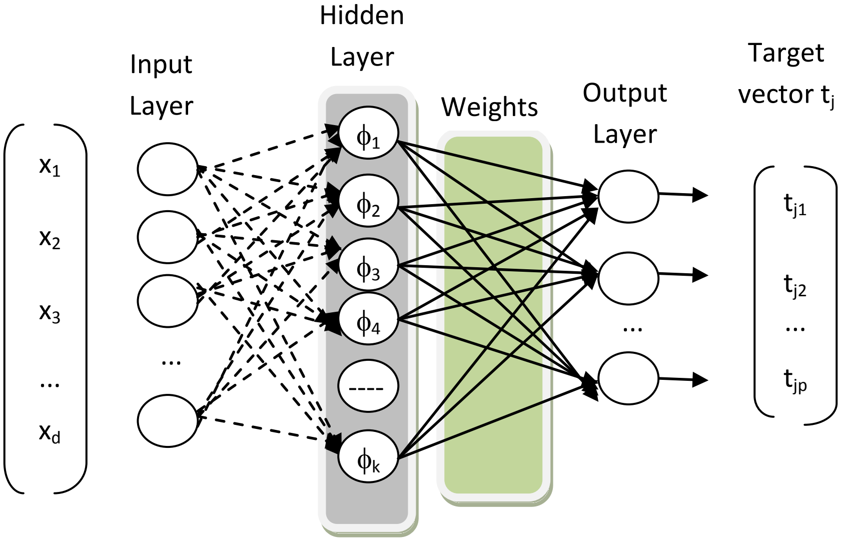


Fig. 4.1 : Three Layered RBFNN Architecture

The basis functions are nonlinear functions. Several forms of the basis functions used in the literature are Gaussian, Multiquadric, Inverse Multiquadric, Thin Spline etc. We have used Gaussian, the most commonly used basis function, represented by

$$\emptyset_j(x) = e^{-\frac{\|x-c_j\|^2}{2\sigma_j^2}} \tag{4.1}$$

Where c_j represents the center of the j^{th} RBF unit. The Gaussian basis function in (4.1) is not normalized, as the factor $\frac{1}{\sqrt{2\pi\sigma}}$ which is used to normalize the function is absorbed into the weights *(Bishop, 1995)*. Figure 4.2 depicts the shape of the Gaussian basis function by plotting the normalized Gaussian distribution. The parameter σ is known as the spread of the radial basis function unit and represents the radius of the d-dimensional ellipsoid with its center at c_j. The center localizes the input feature vector space and corresponds to the mean of the feature vectors in the cluster formed. Let the d-dimensional input feature vector be $V \in \mathbb{R}^d$, and let the number of RBF units be k, then RBFNN can be described as a mapping from $\mathbb{R}^d \rightarrow \mathbb{R}^k$. The output t_j of the RBFNN , corresponding to j^{th} output neuron is given by

$$t_j = \sum_{m=1}^{k} \emptyset_m(x) \times w_{j,m} + w_{j,0} \tag{4.2}$$

where $w_{j,0}$ is the bias, $w_{j,m}$ is the weight of the connection between m^{th} RBF unit and the j^{th} output, obtained by training the RBFNN.

Radial Basis Function Neural Networks(RBFNN) are said to possess the following properties *(Er et al, 2002)*

- They are universal approximators.
- They possess the best approximation property.
- They have compact topology as compared to other Neural Networks.
- They have fast learning capability due to locally tuned neurons.

The basis function is designed in such a way that it can capture the input points only close to centre of the training data points of the same person, and not respond to the data face points of other persons.

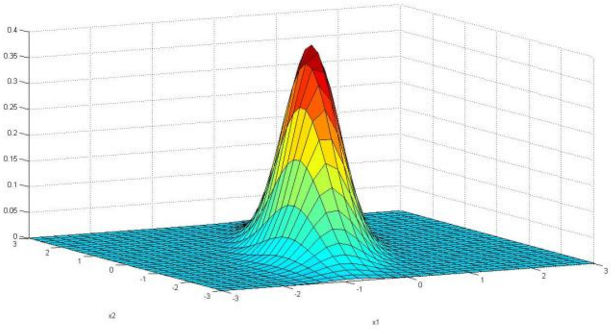


Fig. 4.2 : Gaussian Distribution Function

In general, any basis function is defined by the its center and shape, where shape is measured by the spread σ. Radial Basis Functions Neural Network (RBFNN) and a Multilayer Perceptron (MLP) based neural network differ from each other in that the hidden neurons in MLP form the hyperplanes separating the classes and in RBFNN, the hidden neurons separate classes by kernel functions, e.g., spheroids and ellipsoids (Fig. 4.3). RBFNN is computationally faster than the MLP based neural network.

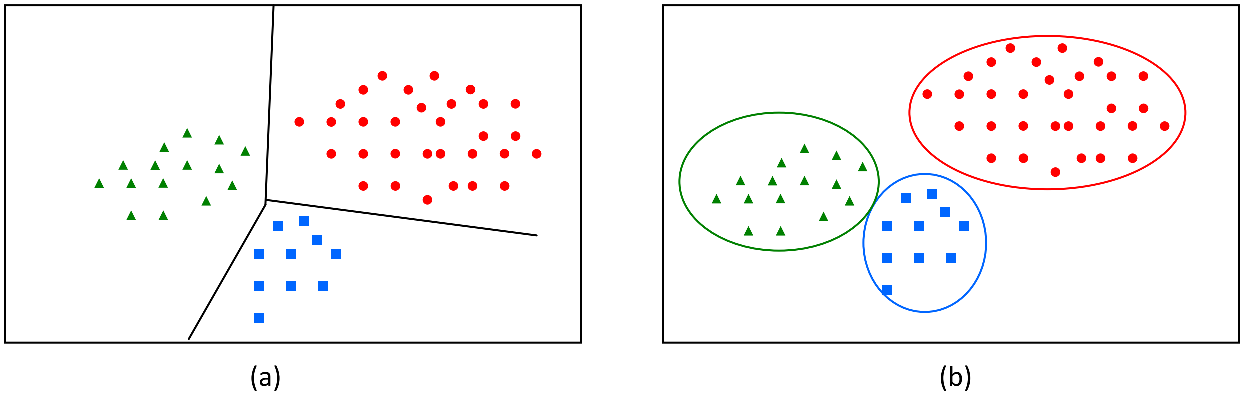


Fig. 4.3 Separation of three classes using (a) Hyper-planes represented by hidden neurons in MLP (b) Radial basis functions at hidden neurons of RBFNN to represent class boundaries. [Figure Adapted from Bishop's Book *(Bishop, 1995)* page 180]

The shape of the basis function affects the area of coverage of points in the d-dimensional space. The points within its spread or span and closer to the center respond to it the maximum. If the design of a basis function is to receive the response of all points belonging to a single class, then its shape should be adapted from the data points in that class. Deciding shape of basis function has been a significant contribution of this thesis and is presented in chapter 5 separately.

Along with the sensitivity of points to the basis functions, it is also important to understand the importance of center localization. One can define the basis functions centers manually or can automate their computations. Manually selected centers may not truly represent the centers of the natural clusters. While automating the center and spread computation requires that the cluster information is available. The cluster formation can be done in unsupervised or supervised way. How accurate and how fast are such computations matter in complex problems such as face recognition or any other pattern recognition tasks. The Center selection accuracy affects the overall performance of the RBFNN.

4.2.1. Role of Hidden Layer

The face recognition problem is a complex pattern recognition problem. The number of features(say 'd') describing a face is usually large making the input feature space a hyper dimensional space. Each training face image is viewed as a point in d-dimensional space. Face images of a person acquired with different illumination and poses should fall in the same cluster of points in the input space, if features are selected carefully. Such clusters belonging to different persons are non-linearly separable in input space (Fig. 4.4).

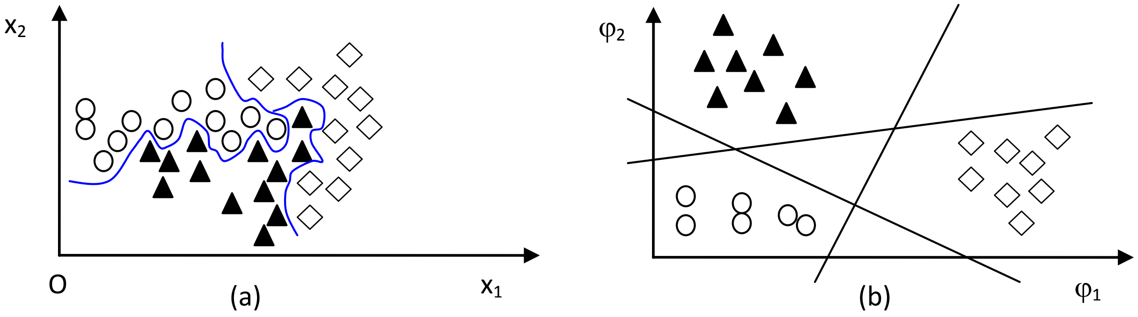


Fig.4.4 Data Separability (a) Non-Linearly Separable face patterns in input space (b) Linearly Separable Transformed points in hidden space

The number of hidden neurons is usually larger than the size of the input layer. The nonlinear transformation to a higher dimensional space is performed by a set of real valued functions represented as ϕ(x) given by:

$$\emptyset(x) = [\emptyset_1(\mathbf{x}), \emptyset_2(\mathbf{x}), \ldots . \emptyset_k(\mathbf{x})]^T \quad (4.3)$$

where pattern **x** is a d-dimensional vector in input space. The vector φ(**x**) then maps the point x into k-dimensional hidden space. Correspondingly the space spanned by the functions $\varphi_i(\mathbf{x})$ (i = 1, 2, ..., k) is referred to as the hidden space. The complex pattern recognition tasks are solved using RBFNN by transforming to the high dimensional space *(Haykin, 1999)*. The justification of this is given by *Cover's Theorem* on the *separability of patterns (Cover, 1965)* stated as follows

> *"A complex pattern classification problem cast in a high-dimensional space nonlinearly is more likely to be linearly separable than in a low-dimensional space".*

Linear separability in the hidden space is captured by linear boundaries or hyper planes. The parameters of these boundaries are learned from the training data in terms of the weights matrix *W*. A d-dimensional test input vector *y* is classified in a two class classification problem (with classes C_1 and C_2) as follows

$$\text{If} \quad W^T\emptyset(y) > 0\,, \quad then\, y \in C_1 \tag{4.4}$$

$$\text{and} \quad W^T\emptyset(y) < 0\,, \quad then\, y \in C_2 \tag{4.5}$$

The hyper plane describing the separating surface in the hidden space (also called as φ-space) is defined by

$$W^T\emptyset(\mathbf{x}) = 0 \tag{4.6}$$

4.2.2. *Input Output Mapping Through the Hidden Layer*

The hidden space is a k-dimensional space where *k* is the number of hidden neurons. The learning process consists of two phases - (1) Training Phase and (2) Generalization Phase. Learning is viewed as obtaining parameters of the approximating hyper surface through an optimization procedure and also as multivariate interpolation problem. Let a total of *N* training face images belonging to *p* authorized persons in the face database , be used to train the RBFNN. Consider a set $\{x_i \in \mathbb{R}^d | x_i\ is\ the\ training\ feature\ vector\ and\ i = 1,2,\ldots,N\}$ of all training samples of face images where each training face image is represented as a point in d-dimensional input space. A corresponding set of *N* target vectors representing class labels for N training face images is taken as set $\left\{t_j \in \mathbb{R}^p \middle| j = 1,2,\ldots,N\right\}$ where t_j is represented as a vector of real values <t_{j1}, t_{j2}, ..., t_{jp}>. An input output mapping describing mapping of input test image x_i to the target output person class t_j is obtained by a function F(x_i) = t$_j$. The function *F* describes the hyper surface passing through all the training points and is characterized by the RBFNN based classifier for face recognition.

The basis functions have the capability to approximate the function *F* by a number of basis functions $\emptyset_1(x_i), \emptyset_2(x_i), \dots . and\ \emptyset_k(x_i)$. These *k* basis functions differ in their formation as their centers and shapes (i.e. spread) differ from each other and represent the total number of neurons in the hidden layer of RBFNN. The function *F* is then given by

$$F(x_i) = \begin{bmatrix} t_{1i} \\ t_{2i} \\ \vdots \\ t_{pi} \end{bmatrix} \tag{4.7}$$

where t_{ji} is the response of the RBFNN for j^{th} output neuron for i^{th} sample. The parameter *w* is the learning weight. The target responses t_{ji}, corresponding to the j^{th} output neuron, for a feature vector x_i are computed as linear combination of the products $w_{jk}\emptyset_k(x_i)$, over all hidden neurons, where $\emptyset_k(x_i)$ is the output triggered by the k^{th} hidden neuron (RBF Unit) and w_{jk} is the weight associated with the k^{th} hidden neuron and j^{th} output neuron. The set of linear equations is obtained as follows

$$w_{11}\emptyset_1(x_i) + w_{12}\emptyset_2(x_i) + \cdots + \ w_{1k}\emptyset_k(x_i) \ = \ t_{1i} \tag{4.8}$$

$$w_{21}\emptyset_1(x_i) + w_{22}\emptyset_2(x_i) + \cdots + \ w_{2k}\emptyset_k(x_i) \ = \ t_{2i} \tag{4.9}$$

....

....

$$w_{p1}\emptyset_1(x_i) + w_{p2}\emptyset_2(x_i) + \cdots + \ w_{pk}\emptyset_k(x_i) \ = \ t_{pi} \tag{4.10}$$

If the function value of $\emptyset_j(x_i)$ is represented as $\emptyset_{ij}$ then the above linear equations for all training samples {x$_i$: i = 1, 2, ..., N} are written in matrix multiplication form as:

$$\begin{bmatrix} w_{11} & w_{12} & \dots w_{1i} & w_{1k} \\ w_{21} & w_{22} & \cdots w_{2i} & w_{2k} \\ \vdots & \vdots & \vdots \ \vdots & \vdots \\ w_{p1} & w_{p2} & \cdots w_{pi} & w_{pk} \end{bmatrix} \begin{bmatrix} \varphi_{11} & \varphi_{12} & \dots \varphi_{1i} & \varphi_{1N} \\ \varphi_{21} & \varphi_{22} & \cdots \varphi_{2i} & \varphi_{2N} \\ \vdots & \vdots & \vdots \ \vdots & \vdots \\ \varphi_{k1} & \varphi_{k2} & \cdots \varphi_{ki} & \varphi_{kN} \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} & \dots t_{1i} & t_{1N} \\ t_{21} & t_{22} & \cdots t_{2i} & t_{2N} \\ \vdots & \vdots & \vdots \ \vdots & \vdots \\ t_{p1} & t_{p2} & \cdots t_{pi} & t_{pN} \end{bmatrix} \tag{4.11}$$

which is represented as

$$\mathrm{W}\,\Phi = \mathrm{T} \tag{4.12}$$

where

$$\Phi = \begin{bmatrix} \varphi_{11} & \varphi_{12} & \dots \varphi_{1i} & \varphi_{1N} \\ \varphi_{21} & \varphi_{22} & \cdots \varphi_{2i} & \varphi_{2N} \\ \vdots & \vdots & \vdots \ \vdots & \vdots \\ \varphi_{k1} & \varphi_{k2} & \cdots \varphi_{ki} & \varphi_{kN} \end{bmatrix}_{k\times N} \tag{4.13}$$

is called as the interpolation matrix. The size of matrix Φ is k × N where *N* is the number of training samples and *k* is the number of hidden neurons. The weight matrix *W* is given by

$$W = \begin{bmatrix} w_{11} & w_{12} & \dots w_{1i} & w_{1k} \\ w_{21} & w_{22} & \cdots w_{2i} & w_{2k} \\ \vdots & \vdots & \vdots \ \vdots & \vdots \\ w_{p1} & w_{p2} & \cdots w_{pi} & w_{pk} \end{bmatrix}_{p\times k} \tag{4.14}$$

The target matrix represents the desired response (class label) and is given by

$$T = \begin{bmatrix} t_{11} & t_{12} & \dots t_{1i} & t_{1N} \\ t_{21} & t_{22} & \cdots t_{2i} & t_{2N} \\ \vdots & \vdots & \vdots \ \vdots & \vdots \\ t_{p1} & t_{p2} & \cdots t_{pi} & t_{pN} \end{bmatrix}_{p\times N} \tag{4.15}$$

Assuming that the interpolation matrix Φ is non singular, we consider the existence of its inverse and obtain the weight matrix as

$$\mathrm{W} = T\Phi^{-1} \tag{4.16}$$

The above computation is the training part using information in *T* and *Φ* matrix. The matrix *Φ* is of size k× N and is not a square matrix, therefore its inverse is obtained using pseudo-inverse method by *Broomhead and Lowe (1988).* Multiplying both sides of (4.12) by the transpose matrix Φ^T, we get

$$\mathrm{W}\Phi\Phi^{\mathrm{T}} = \mathrm{T}\Phi^{\mathrm{T}} \tag{4.17}$$

The product $\Phi\Phi^T$ is of size k×k and is a square matrix and its inverse exists. Therefore multiplying both sides of (4.17) by the inverse of $\Phi\Phi^T$, we get

$$\mathrm{W} = \mathrm{T}\Phi^{\mathrm{T}}\,(\Phi\Phi^{\mathrm{T}})^{-1} \tag{4.18}$$

Let us denote $\Phi^{\mathrm{T}}(\Phi\Phi^{\mathrm{T}})^{-1}$ by Φ^{+} , then

$$W = T\,\Phi^{+} \tag{4.19}$$

Comparing (4.16) and (4.19), the size of matrix Φ^{+} is N×k and that of the right hand side of (4.19) is p×k. The matrix Φ^{+} is considered as equivalent to the inverse of the matrix Φ. The matrix Φ^{+} is called as pseudo inverse of matrix Φ.

The learning of weights is not using error correction method as is done in the back propagation of error in multilayered architectures of neural networks. The testing involves finding the target matrix *S*, not yet known for unseen input test sample *y* and is obtained by using equation (4.12) given below

$$\mathrm{S} = \mathrm{W}\,\Phi \tag{4.20}$$

where the weight matrix *W* is learned through training as is computed in (4.19). The matrix *Φ* for testing samples (say *s* in number) is given by

$$\Phi = \begin{bmatrix} \varphi_{11} & \varphi_{12} & \cdots \varphi_{1i} & \varphi_{1s} \\ \varphi_{21} & \varphi_{22} & \cdots \varphi_{2i} & \varphi_{2s} \\ \vdots & \vdots & \vdots \ \vdots & \vdots \\ \varphi_{k1} & \varphi_{k2} & \cdots \varphi_{ki} & \varphi_{ks} \end{bmatrix}_{k\times s} \tag{4.21}$$

Depending upon modeling of output target vector, the maximum value of the response computed as $t_{ci} = \max_{1\le j\le p}(t_{ji})$ is considered for deciding the predicted class and the test vector x_i is labeled as class *c* corresponding to the maximum value.

4.2.3. *Learning Parameters in RBFNN*

The definition of learning in the context of neural networks given by *Haykin (1999)* (adapted from *Mendel and McLaren (1970))* is defined as follows:

"Learning is a process by which the free parameters of a neural network are adapted through a process of stimulation by the environment in which the network is embedded. The type of learning is determined by the manner in which the parameter changes take place."

The most significant property of a neural network is its capability to learn from its environments and improve its performance. The learning in a neural network takes place through an interactive process by which it adjusts its synaptic weights. Each architecture of a neural network has set of parameters such as number of neurons in each layer, number of layers, synaptic weights associated with each link etc. Radial Basis Function Neural Network has following parameters.

- Number of Radial Basis Function (RBF) Units or Neurons.
- Centers of RBF Units
- Spread and other parameters describing shape of the Basis Functions at each RBF unit.

- Synaptic Weights between hidden and output layers etc.

Learning in RBFNN pertains to finding best parameters such as centers, shape and weights so as to get the best performance of the RBFNN. In the simplest method of center selection, location of the centers in a d-dimensional input space is selected randomly as some of the training data points. This method is useful when training data are distributed in a representative manner. Other approaches are based on self organization of data points in clusters using unsupervised clustering techniques or are based on supervised selection of centers. The unsupervised clustering of the d-dimensional data points clusters the data points into sub-groups, each of which must be as homogeneous as possible. One such unsupervised algorithm is k-Means and has a limitation that it can get stuck in the local optimum solution, i.e. at the centers which may not be optimal. In supervised clustering based center selection, the mean of the patterns in each supervised class is taken as the center. The synaptic weights represented as matrix W are computed using the pseudo inverse method *(Broomhead and Lowe, 1988)*. The weight learning being straight forward does not become part of network learning and the major focus of network learning is in structure learning of the data in finding optimal number and centers of RBF units.

4.2.4. Center Selection of RBF Units Viewed as an Optimization Problem

The centers of the input feature vectors are obtained by taking the average feature vectors of the clusters. The clusters are obtained using unsupervised or supervised clustering methods. The process of automated clustering strives to achieve the subsets of data points such that the sum of squared distances of the points of subset from the subset mean is minimum. Figure 4.5 depicts the center selection viewed as an optimization problem. When the number of data points is large, the problem of subset selection for obtaining cluster center is an NP hard problem and the algorithm complexity is exponential in time. The NP hard problems are solved using approximation and randomized algorithms to obtain near optimal solution in polynomial time.

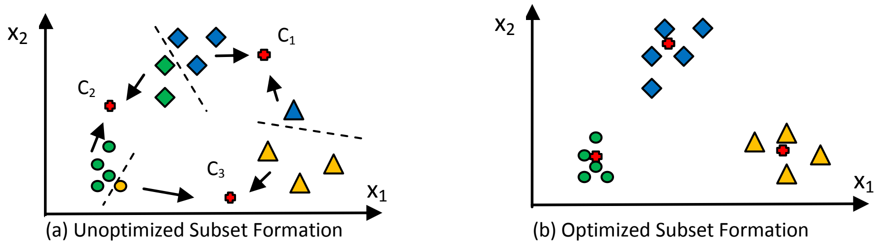

x2
C1
C2
C3
x1
(a) Unoptimized Subset Formation
x2
x1
(b) Optimized Subset Formation

Fig. 4.5. Center Selection viewed as an Optimization Problem

4.3. Proposed FRBFNN Algorithm for Face Recognition

It is observed through literature survey that the potential of Firefly Algorithm *(Yang, 2008)* has not been used in RBFNN center selection for face recognition. Therefore it is proposed to take advantage of the fast and efficient convergence of the Firefly Algorithm and to combine it with the learning capabilities of RBFNN for high speed face recognition. The original firefly algorithm has been discussed in detail in Section 3.4. In this work, the center selection problem is visualized as a hyper-dimensional search problem and mathematical fireflies are designed to move in space to obtain best (optimal) centers.

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 work 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 ORL, Yale , AR and LFW (Labeled Faces in the Wild)

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.

In this module, 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 to sub-clustering and the cumulative number of sub-clusters thus evolved, for all persons, produce the number of neurons in the hidden layer. The details of the proposed FRBFNN algorithm are presented in following subsection.

4.3.1. 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

$$F = \begin{bmatrix} C_1^F \\ C_2^F \\ C_3^F \\ \vdots \\ C_K^F \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & c_{13} & \cdots & c_{1d} \\ c_{21} & c_{22} & c_{21} & \cdots & c_{2d} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ c_{K1} & c_{K2} & c_{K3} & \cdots & c_{Kd} \end{bmatrix} \qquad (4.22)$$

Where center C_i^F is a d-dimensional vector given by $< c_{i1} \quad c_{i2} \quad c_{i3} \quad \cdots \quad c_{id} >$, i = 1, 2, ..K. The firefly proposed above is of dimension K×d which moves in a [K×d] +1 dimensional space. The additional dimension represents the dimension of the fitness function.

4.3.2. Visualization of Proposed Firefly in a Hyperspace

A firefly is viewed as a hyper-dimensional polygon surface comprising of 'K' d-dimensional vertices representing 'K' possible centers of the RBF units, where K > 1. Consider a firefly moving in two dimensional space in search of three possible centers of RBF units represented as three vertices and is therefore visualized as a triangle formed by three vertices each (Fig.4.6). The vertices P, Q and R represent the firefly F_1 and P', Q' and R' represent the firefly F_2. The K-vertex polygon shape for a maximum 'K' RBF unit centers in d-dimensional space associated with each firefly is conceptualized to visualize a firefly and highlight its significance as a single entity holding all feasible centers of the 'K' RBF units.

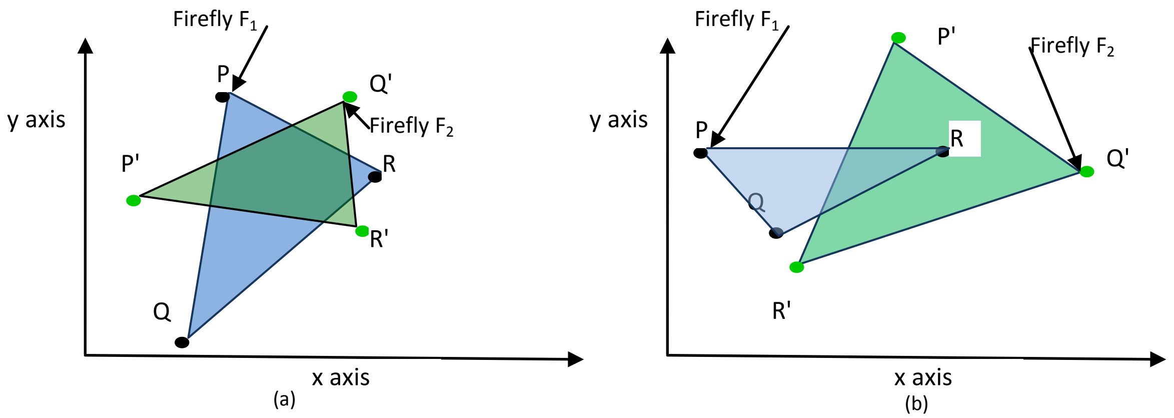


Fig. 4.6: Visualization of Fireflies in the context of RBFNN center Selection(a) Initial positions of the Fireflies F_1 (Blue Triangle) and F_2 (Green Triangle) (b) New positions of the fireflies after movement.

4.3.3. 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, .., M is represented as a d-dimensional feature vector $\langle f_1^u \quad f_2^u \quad f_3^u \quad \dots \quad f_d^u \rangle$. A sub-cluster S_i is constructed as the collection of all feature vectors nearest to the 'i'th center C_i^F given by

$$S_i = \{ \bigcup_{1\le u\le M} T_u \quad | \quad \|C_i^F - T_u\| < \|C_j^F - T_u\| \quad \forall\, j = 1,2,..,K, i \neq j \quad \} \qquad (4.23)$$

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

$$G^F = -\sum_{i=1}^{K} \sum_{\substack{u=1 \\ T_u \in S_i}}^{M} \|C_i^F - T_u\|^2 \qquad (4.24)$$

where $\|C_i^F - T_u\|$ is the Euclidean distance between the 'i'th center represented by firefly F, i.e. C_i^F, and the training image feature vector T_u.

In the context of center selection, the fitness function G^F is considered to be good if the centers arrived at by the movements of the firefly F represent centers close to centers of sub-clusters identified subjectively by human intelligence. Figure 4.7 depicts relative fitness values of the two fireflies in the context of RBFNN center selection. The firefly F_1 (Blue colored triangle) represents more accurate centers (as Red colored points) as compared to the ones represented by firefly F_2 (Green colored triangle), therefore the fitness function value of firefly F_1 is larger than that of F_2. The centers represented by the brightest firefly are considered and the sub-clusters are formed using equation (4.23). The mean vector of the face feature vectors belonging to each sub-cluster is considered as the center of the RBF unit.

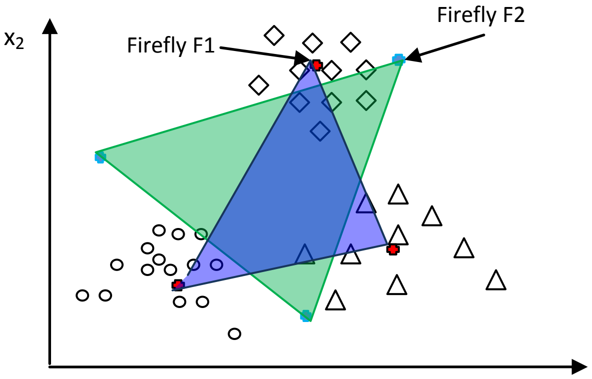


Fig.4.7: Relative fitness values of the two fireflies F_1 and F_2 in the context of Center Selection ($G^{F1} > G^{F2}$)

4.3.4. *Firefly Movement*

The firefly with less fitness value moves towards the brighter firefly with larger fitness value. The entire process of movement of fireflies in the search space is based on the fact that the search space is visited using attractiveness heuristic. Fireflies explore the search space very fast and have the potential to obtain multiple optimal solutions. Fireflies follow two types of movements- one based on attractiveness and the other random. The movement due to attractiveness pushes the firefly in the direction of the brighter firefly so that the firefly explores the path towards the brighter firefly. The random movement of the firefly in its neighborhood provides with an opportunity to explore the nearby positions for better solutions. The firefly, while moving in the [K×d]+1 dimensional search space acquires, different positions and its brightness changes. It is important to note that the fireflies are bound to move within the boundaries of the search space. If any firefly tries to move by a distance that may take it beyond the boundaries of the search space, then the new position of the firefly in that move is the position on the boundary itself. Search space range is defined as

$$SearchSpaceRange = \begin{bmatrix} \min_{\substack{1\le u\le M \\ T_u \in T}} f_1^u & \max_{\substack{1\le u\le M \\ T_u \in T}} f_1^u \\ \min_{\substack{1\le u\le M \\ T_u \in T}} f_2^u & \max_{\substack{1\le u\le M \\ T_u \in T}} f_2^u \\ \vdots & \vdots \\ \min_{\substack{1\le u\le M \\ T_u \in T}} f_d^u & \max_{\substack{1\le u\le M \\ T_u \in T}} f_d^u \end{bmatrix} \tag{4.25}$$

The search space range is represented as a matrix of size d×2 where d is the size of the feature vectors. The first column represents the minimum feature values in each dimension of the training feature vectors. The second column represents the maximum value of the feature values in each dimension of the training feature vectors.

The firefly movement consists of two types of shifts- *heuristic shift* and *random shift.* Heuristic shift is the distance to be travelled by a firefly due to the attractiveness of the other firefly and the other movement is random in nature . The d-dimensional vector 'alpha ' [<α_1, α_2, ..., α_d>] defines randomization parameter of firefly step size given by

$$\alpha_i = \frac{\left| \max\limits_{\substack{1\le u\le M \\ T_u \in T}} f_i^u - \min\limits_{\substack{1\le u\le M \\ T_u \in T}} f_i^u \right|}{10 \times \sqrt{I_{max} \times d}} \tag{4.26}$$

I_{max} is the maximum number of iterations in which fireflies are expected to search input space range in each dimension. The formulation of the random step size as in equation (4.26) makes the firefly movement adaptive to the input space ranges in each dimension. The value of α_i decreases by a factor of 0.03 at each iteration for i = 1,2, ...d. The random shift in ith dimension is then obtained by multiplying α_i by a random value (rand-0.5) *(Yang, 2008).* The function 'rand' is a random number generator giving a random number uniformly distributed in [0,1] *(Yang, 2008).* The position vector between the two fireflies is a directional displacement between the two fireflies. If the fireflies F_i and F_j are represented as:

$$F_i = \begin{bmatrix} c_{11} & c_{12} & c_{13} & \cdots & c_{1d} \\ c_{21} & c_{22} & c_{21} & \cdots & c_{2d} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ c_{K1} & c_{K2} & c_{K3} & \cdots & c_{Kd} \end{bmatrix} \tag{4.27}$$

$$F_j = \begin{bmatrix} d_{11} & d_{12} & d_{13} & \dots & d_{1d} \\ d_{21} & d_{22} & d_{21} & \dots & d_{2d} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ d_{K1} & d_{K2} & d_{K3} & \cdots & d_{Kd} \end{bmatrix} \tag{4.28}$$

The position vector between F_i and F_j is computed as follows

$$\begin{bmatrix} c_{11} - d_{11} & c_{12} - d_{12} & c_{13} - d_{13} & \dots & c_{1d} - d_{1d} \\ c_{21} - d_{21} & c_{22} - d_{22} & c_{21} - d_{21} & \dots & c_{2d} - d_{2d} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ c_{K1} - d_{K1} & c_{K2} - d_{K2} & c_{K3} - d_{K3} & \cdots & c_{Kd} - d_{Kd} \end{bmatrix} \tag{4.29}$$

The distance between the two fireflies F_i and F_j is computed as Euclidean distance in a [K×d] space. The larger distance between fireflies causes less attractiveness between the fireflies.

$$distance = \|F_i - F_j\| = \sum_{p=1}^{K}\sum_{q=1}^{d}(c_{pq} - d_{pq})^2 \tag{4.30}$$

The algorithm for movement of fireflies in the search space is given in Figure 4.8. The parameters of algorithm are described in Table 4.1.

Table 4.1 Details of input Parameters for Movement of Fireflies

Input Parameters	**Description**
F_i	*Less bright Firefly that moves*
F_j	*More bright firefly that attracts F_i, but does not move on its own*
β_0	*Attractiveness at distance r =0, taken as 1*
γ	*Light absorption coefficient*
K	*Maximum Number of possible Centers in a training class*
d	*Number of Training features*
alpha	*Vector containing appropriate values of initial step size taken randomly*
searchSpaceRange	*d× 2 matrix containing search space range*

```
moveFirefly ($F_i$, $F_j$, $\beta_0$ , $\gamma$ , K, d,  alpha, searchSpaceRange )
{
    Step 1:     Compute position vector
                        positionVector = $F_i$ - $F_j$
    Step 2:     Compute distance between fireflies i and j
                        r = $\|F_i - F_j\|$
    Step 3:     Compute attractiveness between fireflies $F_i$ and $F_j$
                        $attractiveness = \beta_0 exp^{-\gamma r^2}$
    Step 4:     Compute Shift Matrix R
                Initialize R as empty matrix of size Kxd
                for p = 1:K     // center p of the firefly
                        for q = 1:d // 'p'th center's 'q'th coordinate
                                heuristic_Shift = attractiveness*positionVector(p,q);
                                random_Shift =  alpha(q)*(rand-0.5);
                                R[p,q] = heuristic_Shift + random_Shift;
                        end
                end
    Step 5:     A = $F_i$  // A is a temporary matrix of size K x d
                for p = 1:K     // center p of the firefly
                        for q = 1:d // 'p'th center's 'q'th coordinate
                                new_Position[p,q] = A[p.q] + R[p,q];
                        end
                end

    Step 6:     Place the moved firefly within the search space range
                for p = 1:K
                        for q = 1:d
                                if(new_Position(p,q) > searchSpaceRange(q,2))
                                        new_Position(p,q) = searchSpaceRange(q,2);
                                elseif(new_Position(p,q) < searchSpaceRange(q,1))
                                        new_Position(p,q) = searchSpaceRange(q,1);
                                end
                        end
        end

}
OutPut : new_Position of firefly $F_i$
```

Fig. 4.8. Algorithm for Movement of Fireflies

4.3.5. *Handling Equally bright Fireflies*

Figure 4.9 shows the movement of firefly (F_i) to two brighter fireflies (say F_j and F_k) of equal brightness but at different distances. Let A and B be the positions of the fireflies F_j and F_k respectively. Let G^{F_j} and G^{F_k} be the fitness values of both fireflies F_j and F_k respectively such that $G^{F_j} = G^{F_k}$.

Let the distance between F_i and F_j be r_{ij} and that between F_k and F_i be r_{ik} (Fig.4.9). Also let us take the ratio between the two distances as

$$\frac{r_{ik}}{r_{ij}} = L \qquad \text{for L} > 1 \tag{4.31}$$

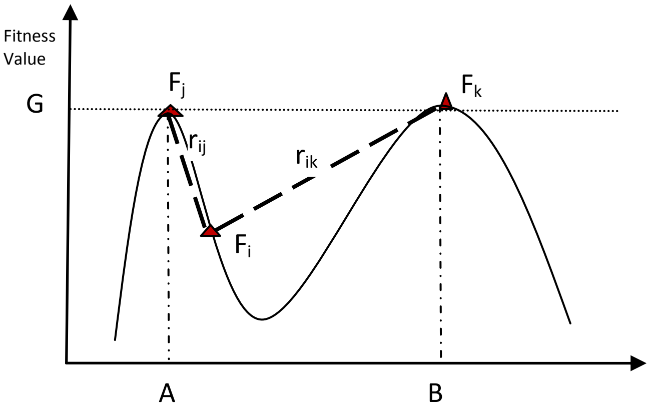


Fig. 4.9: Relative Heuristic Shift

The firefly F_i is less brighter than fireflies F_j and F_k, therefore F_i moves towards both F_j and F_k. The distance moved by F_i towards F_j is given by the term heuristic_Shift$_{ij}$ defined by

$$heuristic\_Shift_{ij} = \beta_0 exp^{-\gamma r_{ij}^2}\left(F_i - F_j\right) \tag{4.32}$$

and

$$heuristic\_Shift_{ik} = \beta_0 exp^{-\gamma r_{ik}^2}\left(F_i - F_k\right) \tag{4.33}$$

Using (4.31), (4.32) and (4.33), we get,

$$\frac{heuristic\_Shift_{ik}}{heuristic\_Shift_{ij}} \propto \quad exp^{-\gamma(L^2-1)r_{ij}^2} \tag{4.34}$$

Since L > 1, the quantity $-\gamma(L^2 - 1)r_{ij}^2$ on the right hand side of (4.34) is a negative number for $\gamma > 0$,

Hence,

$$\text{heuristic\_Shift}_{ij} > \text{heuristic\_Shift}_{ik}$$

This implies that a firefly moves towards the nearer firefly of the two fireflies with similar brightness implying attractiveness of a firefly not only depends on its brightness, but it also depends on its relative distance from the equally bright fireflies.

4.3.6. New Position of a Moving Firefly

A firefly, while moving in the K×d dimensional search space acquires different positions and changes its brightness. The fireflies acquire new position within the search space boundary and are not allowed to go outside the boundary. The attractiveness (β) between the two fireflies at distance ' r ' apart, is expressed by

$$\beta = \beta_0 \exp^{-\gamma r^2} \qquad (4.35)$$

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. 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

$$F_i^{t+1} = F_i^t + \beta_0 exp^{-\gamma r^2} \left\| F_j - F_i \right\| + \alpha \in_i \qquad (4.36)$$

Where F_i^t is the previous position of the firefly. The second term is due to the attractiveness between the two fireflies F_i and F_j (*heuristic shift).* 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 (*random shift).* 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.

4.3.7. 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 very fast, it may miss the optima and if it moves very slowly, 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 suboptimal solution.

(iii). Number ofIterations(I): An iteration consists of pair wise 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.

4.3.8. Convergence metrics

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

$$S_{conv} = \{ F_i \mid 1 \leq i \leq Q \; and \; |G^{F_i} - G^{F_{best}}| < T_{Lim} \} \tag{4.37}$$

where T_{Lim} is the tolerance limit within which fireflies converge. Various types of convergence is defined as given below.

(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).

$$|S_{conv}| \geq T_F \tag{4.38}$$

(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 0. 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 represented by

$$\text{Error} = G^{F_{best}} \times \left(1 - \frac{count}{T_{count}}\right) \tag{4.39}$$

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

4.3.9. Algorithm Development Details

The proposed FRBFNN algorithm is shown as a flow chart in figure 4.9. 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 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, .., M) of each person's training images are obtained using appropriate feature extraction technique.

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 pair-wise and move to the brighter firefly in each iteration. An iteration consists of Q×Q evaluations each comprising of the fitness comparison and firefly movement. The error is computed using (4.39) at each evaluation. Fireflies keep moving till the squared error reaches a threshold of 0.0001, or the number of converged fireflies is greater than T_F or the number of iterations reaches its maximum. 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, .., 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 1. The process is repeated for p = 1, 2, .., K.

Step 4: Evolution of Hidden Neurons for Each Person Class (RBF units)

The centers and number of RBF unit 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

Figure 4.10 shows the flowchart of the proposed FRBFNN algorithm and the variables used in the algorithm are shown in Table 4.2. The output of the algorithm is the total number of hidden neurons or RBF units and their optimal centers.

4.3.10. 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 by

$$Average\ Recognition\ Accuracy = \frac{\sum_{i=1}^{i=q}\left(\frac{n_c^i}{n_t}\right)}{q} \times 100 \qquad (4.40)$$

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. Sensitivity analysis of the proposed integrated classifier is presented in Chapter 5, while average recognition accuracy is computed using balanced data set of all person classes taken as authorized with fixed number of training images used per person.

Table 4.2 Details of Variables used in the proposed FRBFNN Algorithm

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 (4.39)

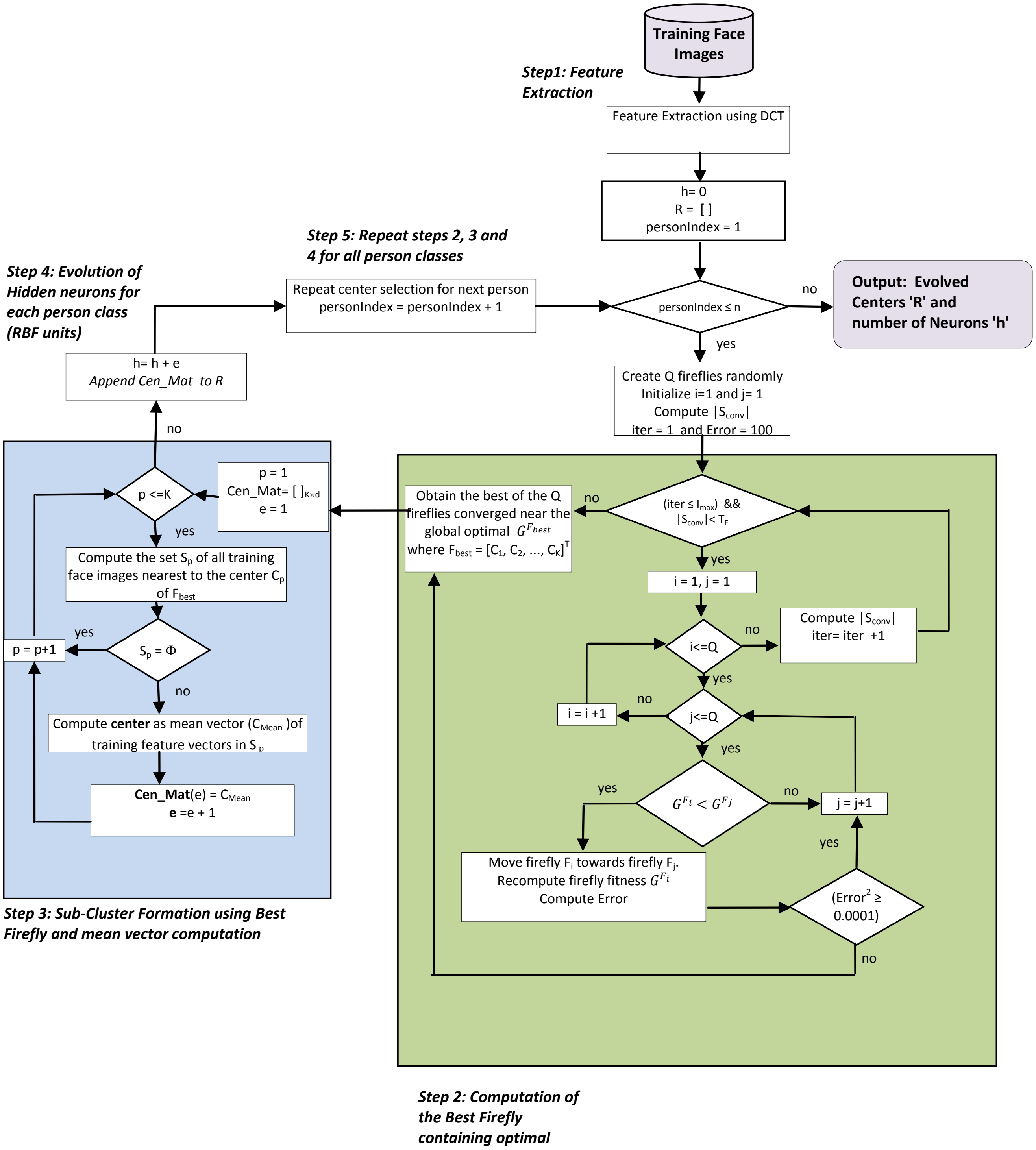

Training Face Images
Step1: Feature Extraction
Feature Extraction using DCT
h= 0
R = [ ]
personIndex = 1
Step 5: Repeat steps 2, 3 and 4 for all person classes
Step 4: Evolution of Hidden neurons for each person class (RBF units)
Repeat center selection for next person
personIndex = personIndex + 1
personIndex ≤ n
no
Output: Evolved Centers 'R' and number of Neurons 'h'
yes
h= h + e
Append Cen_Mat to R
Create Q fireflies randomly
Initialize i=1 and j= 1
Compute $|S_{conv}|$
iter = 1 and Error = 100
no
p = 1
Cen_Mat= $[\ ]_{K\times d}$
e = 1
p <=K
yes
Obtain the best of the Q fireflies converged near the global optimal $G^{F_{best}}$
where $F_{best} = [C_1, C_2, ..., C_K]^T$
no
$(iter \le I_{max})$ &&
$|S_{conv}| < T_F$
yes
i = 1, j = 1
Compute the set S_p of all training face images nearest to the center C_p of F_{best}
yes
p = p+1
$S_p = \Phi$
no
Compute $|S_{conv}|$
iter= iter +1
i<=Q
no
yes
no
i = i +1
j<=Q
yes
Compute **center** as mean vector (C_{Mean}) of training feature vectors in S_p
Cen_Mat(e) = C_{Mean}
e =e + 1
yes
$G^{F_i} < G^{F_j}$
no
j = j+1
yes
Move firefly F_i towards firefly F_j.
Recompute firefly fitness G^{F_i}
Compute Error
$(Error^2 \ge 0.0001)$
no
Step 3: Sub-Cluster Formation using Best Firefly and mean vector computation
Step 2: Computation of the Best Firefly containing optimal

Fig. 4.10. Flow Chart of the Proposed FRBFNN Algorithm for Number of Neurons in Hidden Layer and their Centers Training for Face Recognition

4.3.11. Time Complexity

The worst case time complexity of the proposed technique is given by O($I_{max}KQ^2n^3$) given n^2 >> d and M<< 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. The module wise computation of time complexity expressions are given in Table 4.3.

Table 4.3 : Module wise time complexities used in time complexity computation of the proposed FRBFNN Algorithm

Module	Expression for time complexity
Random generation of 'Q' fireflies	T_1 = QKd
Move Firefly	T_2 = Kd
Firefly Fitness G^F computation	T_3 = KM
Error Computation	T_4 = KMn + KM
Computation of S_{conv}	T_5 = Q
Computation of Best Firefly	T_6 = Q
Sub-cluster formation	T_7 = KMd
Time to add neurons to R	T_8 = constant
Step 5 for all persons	T_9 = n

Assuming that the computation of DCT of each image takes constant time, we compute the time taken by Steps 1-5 individually and then compute the total time taken by the proposed algorithm.

$$\mathrm{T(Step1)} = \text{ number of persons } \times \mathrm{Constt} = \mathrm{O(n)} \qquad (4.41)$$

$$T(step2) = T_1 + I_{max} \times \{Q^2 \times (T_2 + T_3 + T_4) + T_5\} + T_6$$

$$= QKd + I_{max}\{Q^2\,(Kd + KM + KMn + KM) + Q\} + Q \qquad (4.42)$$

where single Q's appearing in the second and third terms in (4.42) are ignored as

$$Q^2 \times (Kd + KM + KMn + KM) \gg Q$$

and for K >1 and d >> 1

$$QKd \gg Q$$

Therefore we can express (4.42) as an inequality to obtain the upper bound on time complexity, we get

$$T(Step2) \leq QKd + I_{max}\,Q^2\,K\,(d + 2M + Mn)$$

For n >> 2, 2M is ignored as compared to M×n and we get

$$T(Step2) \leq \; QKd \; + \; I_{max}\, Q^2\, K\,(d + Mn) \tag{4.43}$$

The time complexity for Step 3 is computed as follows

$$T(Step3) = KMd \tag{4.44}$$

Time to add neurons in matrix R is constant and Step 5 is repeated 'n' times for all persons used for training. Therefore,

$$T(Step4) = constant \tag{4.45}$$

and

$$T(Step5) = T_9 \tag{4.46}$$

$$T(Total\,Time) = T(Step1) + \; T(Step5) \times \{T(Step2) + T(Step3) + T(Step4)\}$$

$$T \leq n + n \times \{\, QKd \; + \; I_{max}\, Q^2 K\,(d + Mn) + KMd + constant\} \tag{4.47}$$

i.e. $T \leq n \times \{\, QKd \;\; + \;\; I_{max}\, Q^2 K\,(d + Mn) \; + \; KMd\}$

Combining the first and the third terms, we get

$$T \leq n \times \{\,(Q + M)Kd \; + \; I_{max}\, Q^2 K\,(d + Mn)\}$$

For M<<n ,

$$T \leq n \times \{\,(Q + n)Kd \; + \; I_{max}\, Q^2 K\,(d + n^2)\}$$

As (Q+n) << Q^2 and since d < (d+n^2), the term (Q + n)Kd is ignored, and we get

$$T \leq n \times \{\; I_{max}\, Q^2 K\,(d + n^2)\}$$

For d<<n^2, we get,

$$T \leq nI_{max}\, Q^2 K\,(n^2)$$

Therefore, the worst case time complexity is given as

$$T = \; O(I_{max}\, K\, Q^2 n^3) \tag{4.48}$$

The proposed FRBFNN is polynomial and is proportional to the square of number of fireflies (Q), cube of number of persons (n) and the maximum number of iterations. The time complexity is also proportional to the maximum number of sub-clusters (K) for each person's training images, which is kept same as the total number of training images used for a person. Equation (4.48) is a polynomial of degree seven and therefore the proposed FRBFNN is a polynomial time complexity algorithm.

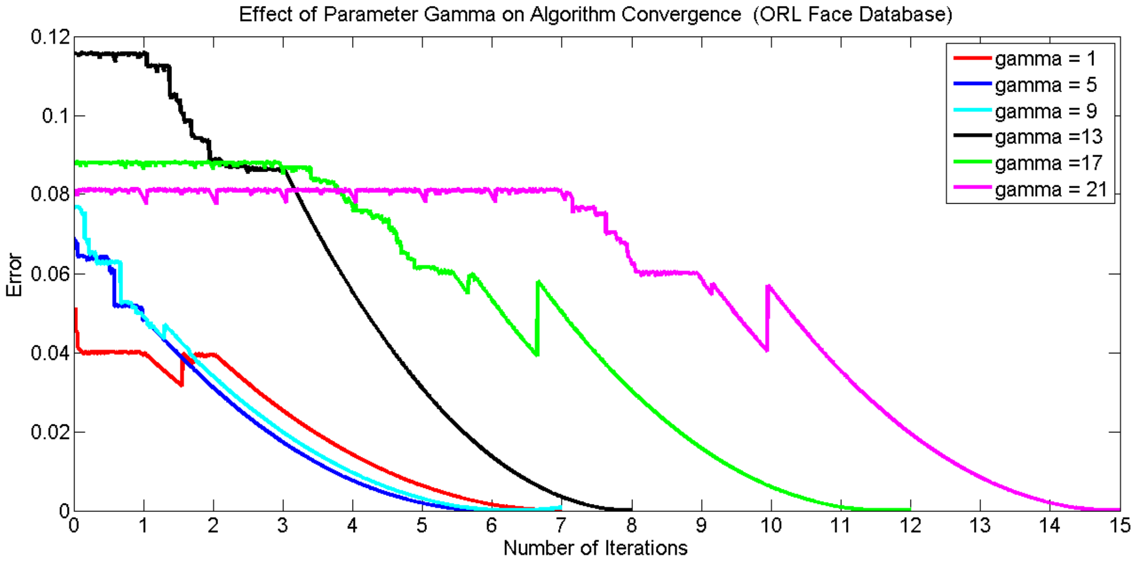
4.4. Parameter Selection for the Proposed FRBFNN Technique

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 $T_{count} = 2000$, where T_{Lim} is the tolerance limit within which fireflies converge, T_F is the threshold for number of fireflies converged and T_{count} is the sub-cluster convergence threshold as defined in section 4.3.8. The automatic tuning of the parameters is a very tough hyper-optimization problem *(Yang et al, 2013)* and is not the focus of present study.

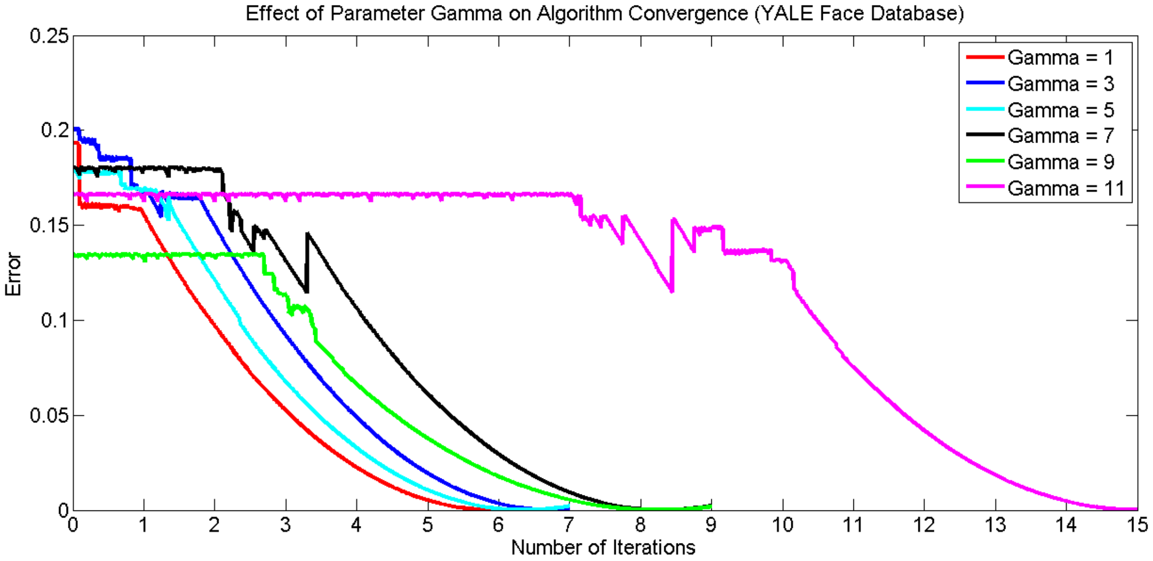
4.4.1. Effect of Parameter Gamma (γ) on Algorithm Convergence

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

It is evident from Figure 4.11 showing effect of γ on algorithm convergence that large values of γ lead to slow convergence. 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. As compared to other methods reported in two different studies *(Sing et al, 2007; Sing et al, 2009)*, where the algorithm converges in 6500 and 15000 epochs respectively, the proposed FRBFNN algorithm converges very fast in 5-15 iterations.

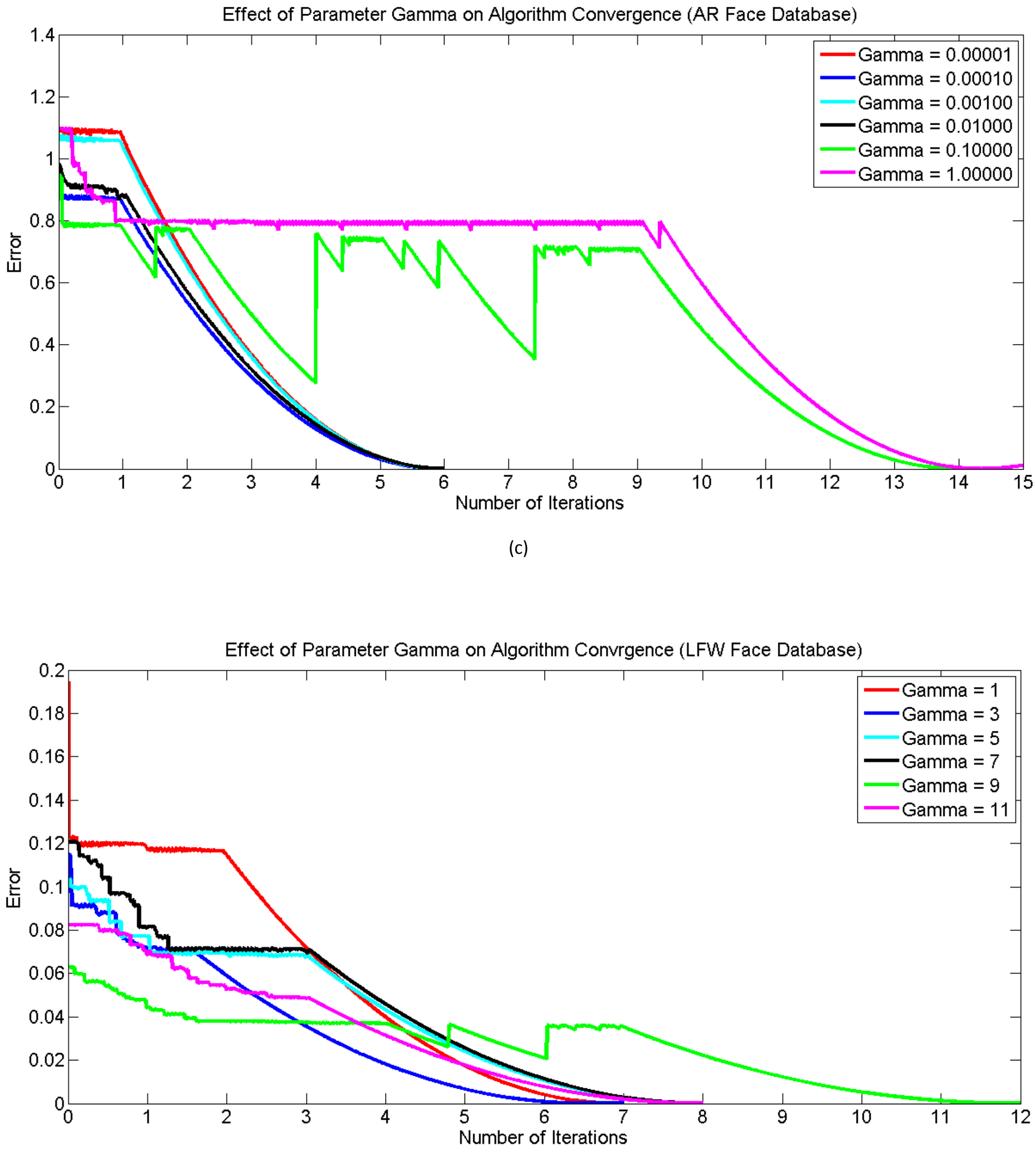


(a)



(b)

Fig.4.11. Effect of Parameter γ on the algorithm convergence (a) ORL (b) Yale Face Databases



(c)

(d)

Fig.4.11. (Continued) Effect of Parameter γ on the algorithm convergence (c) AR and (d) LFW Face Databases

4.4.2. Effect of Parameter Gamma (γ) on Average Recognition Accuracy

Performance in terms of recognition accuracy and stability of proposed algorithm is studied for varying values of γ with maximum number of iterations fixed at 20 and is shown in Table 4.4. The detail of each face database in Table 4.4 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 value of the parameter 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 (± standard Deviation) obtained for ORL, Yale, AR and LFW face databases are 97.35% (± 0.97), 99.83% (± 0.53) , 93.15% (± 3.25) and 55.50% (±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 respectively. However, this number for LFW is 1.05 which is less than the threshold T_F, therefore we select the second best accuracy as 52.50% for LFW obtained with average number of converged fireflies as 16.90 (approx. 17) for γ equal to 1. The above accuracies were arrived in 6.41, 8.78, 5.95 and 7.08 number of iterations (averaged) where average is taken over all person classes and all 10 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.

Table 4.4. Effect of Parameter γ on Recognition Accuracy using the proposed FRBFNN Algorithm on ORL, Yale, AR and LFW face database

Gamma (γ)	**Average Recognition Accuracy ± Standard Deviation**	**Average Number of Converged Fireflies**	**Average number of Iterations**	**Number of Evolved Neurons**
Details of 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
Details of 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
Details of 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
Details of 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

4.4.3. Effect of Number of Fireflies on Algorithm Convergence

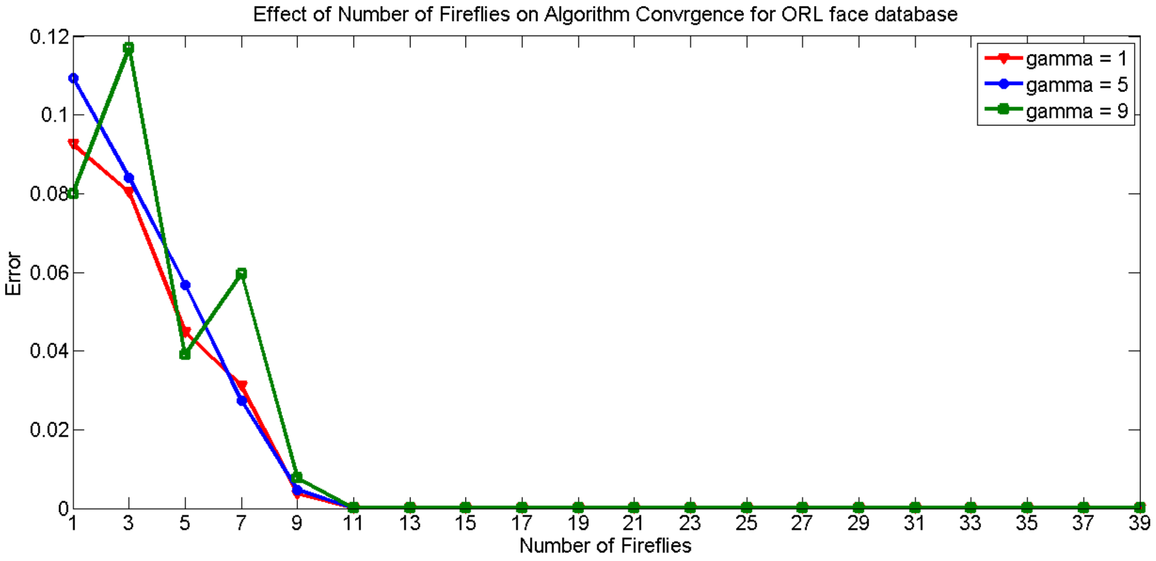
The convergence of fireflies has been investigated with respect to varying values of number of fireflies and is demonstrated in Fig.4.12. The number of iterations is kept fixed as 20 in order to ensure fast computation. The number of fireflies are changed from 1 to 40 in the step of size 2 and the error, as expressed in by equation (4.39), is calculated with respect to number of fireflies. The convergence is investigated with respect to varying values of γ for the given range of fireflies number. 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, e.g. for $\gamma = 5$, the number of fireflies required is greater than 40 and 35 respectively for Yale and LFW face databases which increases the time complexity of the algorithm. The minimum numbers of fireflies required for ORL, Yale, AR and LFW are 11, 13, 11 and 13 respectively using appropriate values of γ which enable algorithm convergence in 20 iterations.

4.4.4. Proposed Parameters

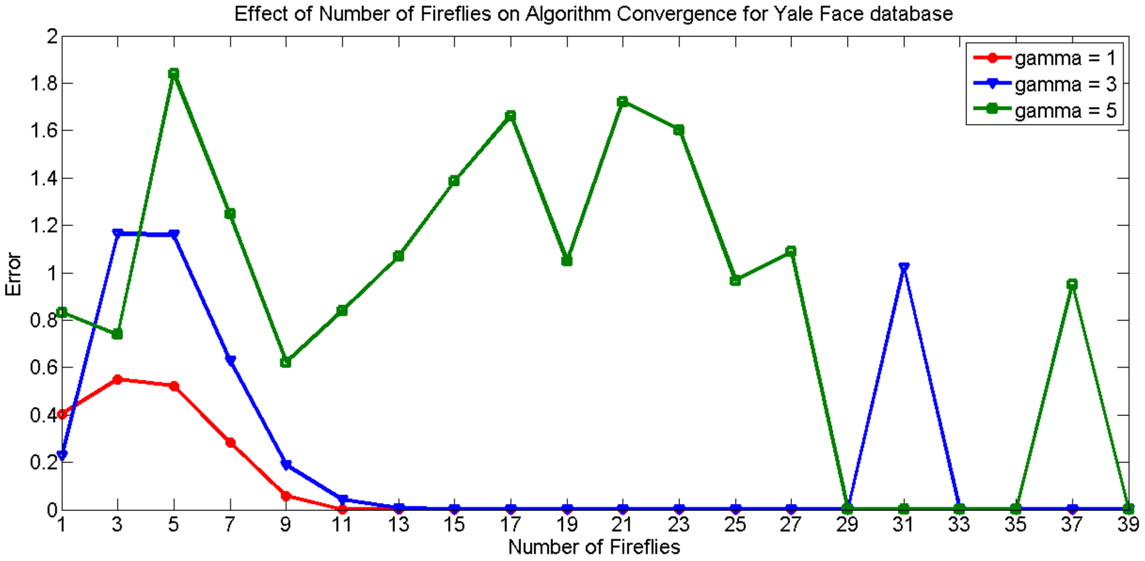
The behavior of the proposed FRBFNN algorithm in hyper-dimensional face training input space shown in Figures 4.11 and 4.12, is summarized in the parameter selection of gamma (γ), number of fireflies (Q) and the number of iterations in Table 4.5. The maximum numbers of fireflies and iterations can be taken as 20 each, taking into consideration limited number of runs.

Table 4.5. Parameter Selection for the proposed FRBFNN Algorithm

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	13	8

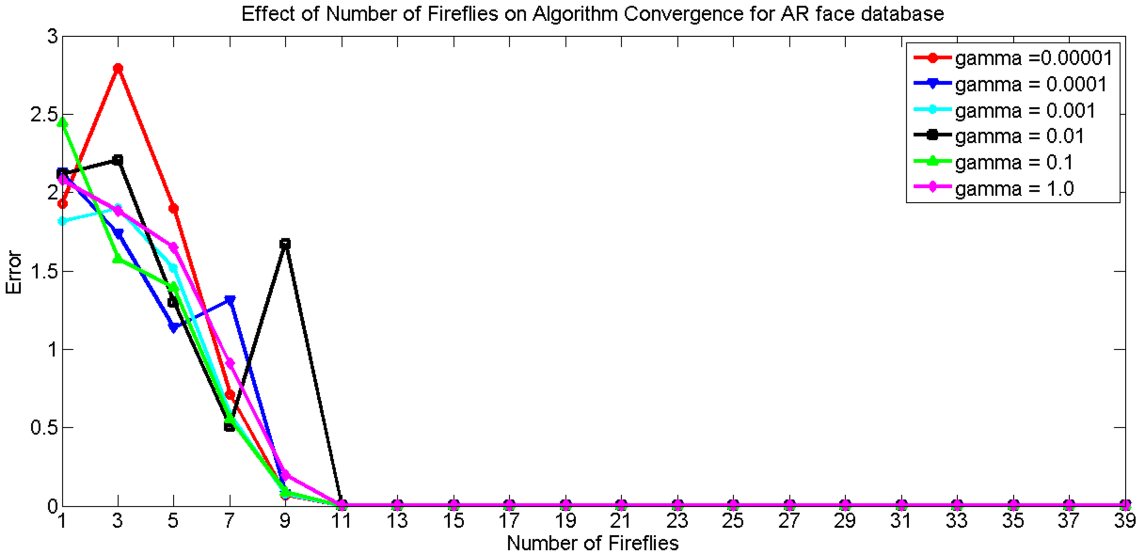


(a)

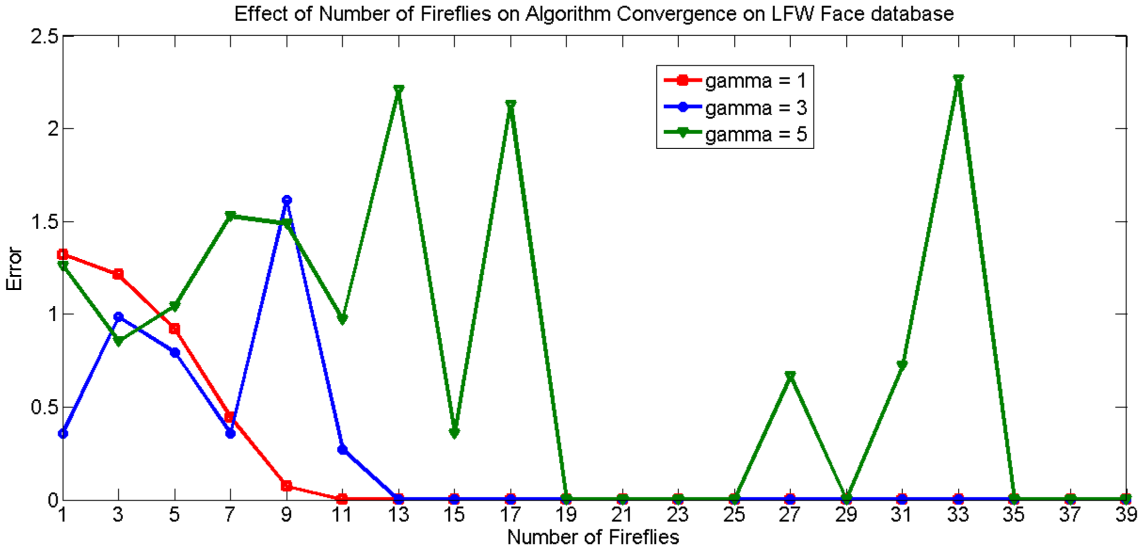


(b)

Fig.4.12. Effect of Number of Fireflies on the algorithm convergence (a) ORL (b) Yale Face Databases



(c)



(d)

Fig.4.12.(Continued). Effect of Number of Fireflies on the algorithm convergence (c) AR and (d) LFW Face Databases

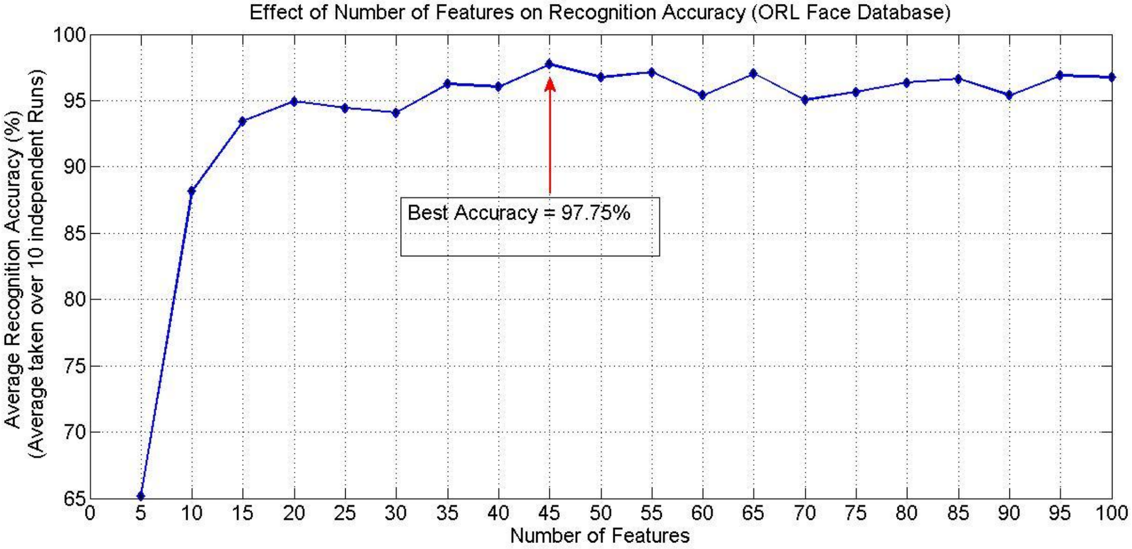
4.5. Results and Discussions

With optimal values of parameter γ chosen for different face databases summarized in Table 4.5, effects of feature dimension, number of training images and sub-clustering on Average Recognition Accuracy, Firefly convergence, evolution of neurons etc. are studied. Maximum numbers of iterations and fireflies are 20 each.

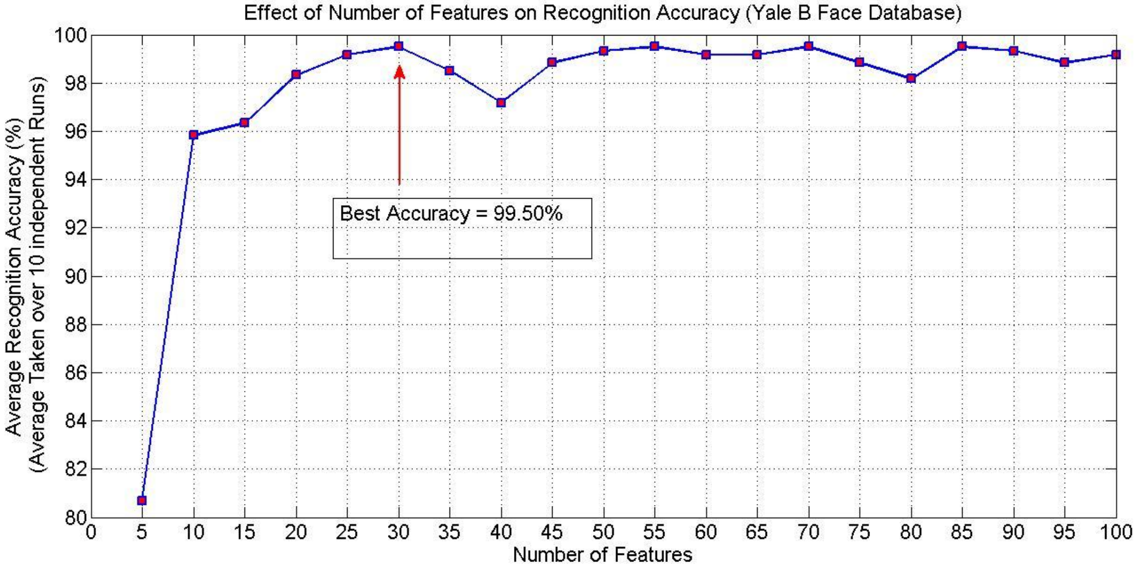
4.5.1. Effect of Number of Features on Average Recognition Accuracy

Feature dimension is investigated for number of features varying from 5 to 100 features for ORL, Yale and LFW and 20 to 300 for AR face databases. Figure 4.13 depicts the effect of number of features on the average number of converged fireflies. It is evident from the curves shown in Figure 4.13 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.

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 4.6 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. 10304. As we have used only 45, 30 and 65 features respectively, dimensionality reduction achieved are 99.5%, 99.7% and 99.4% respectively. For AR face database only 260 features from 4980 are taken 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.

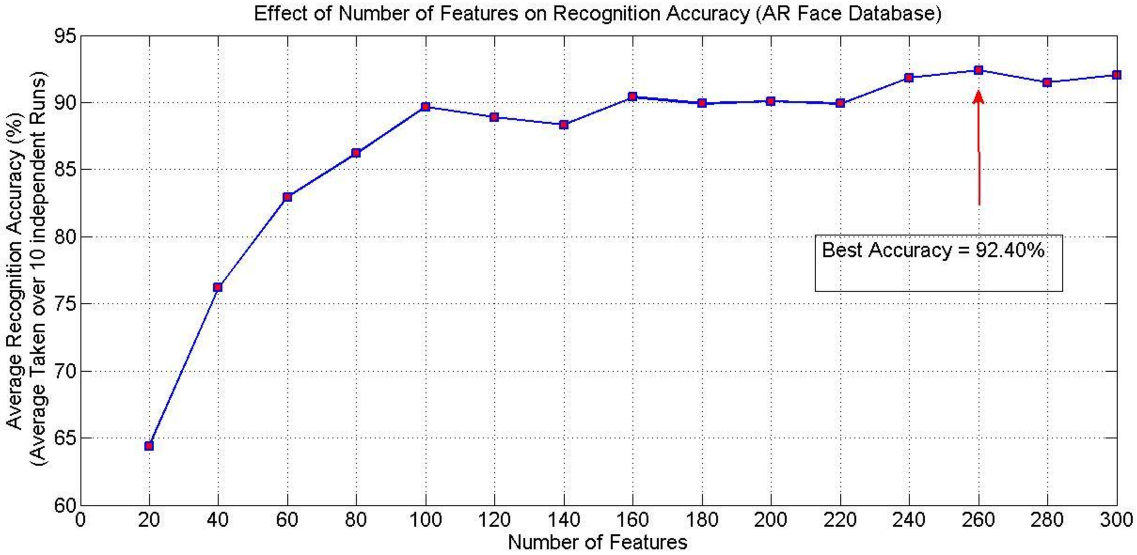


(a)

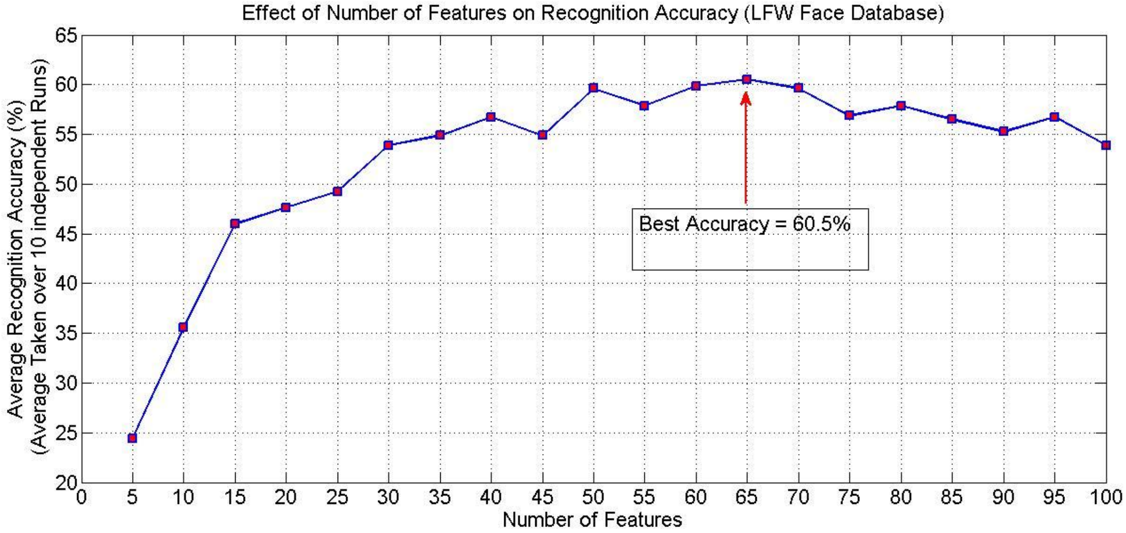


(b)

Fig. 4.13. Effect of feature dimension on Average Recognition accuracy using the proposed FRBFNN (a) ORL (b)Yale face databases

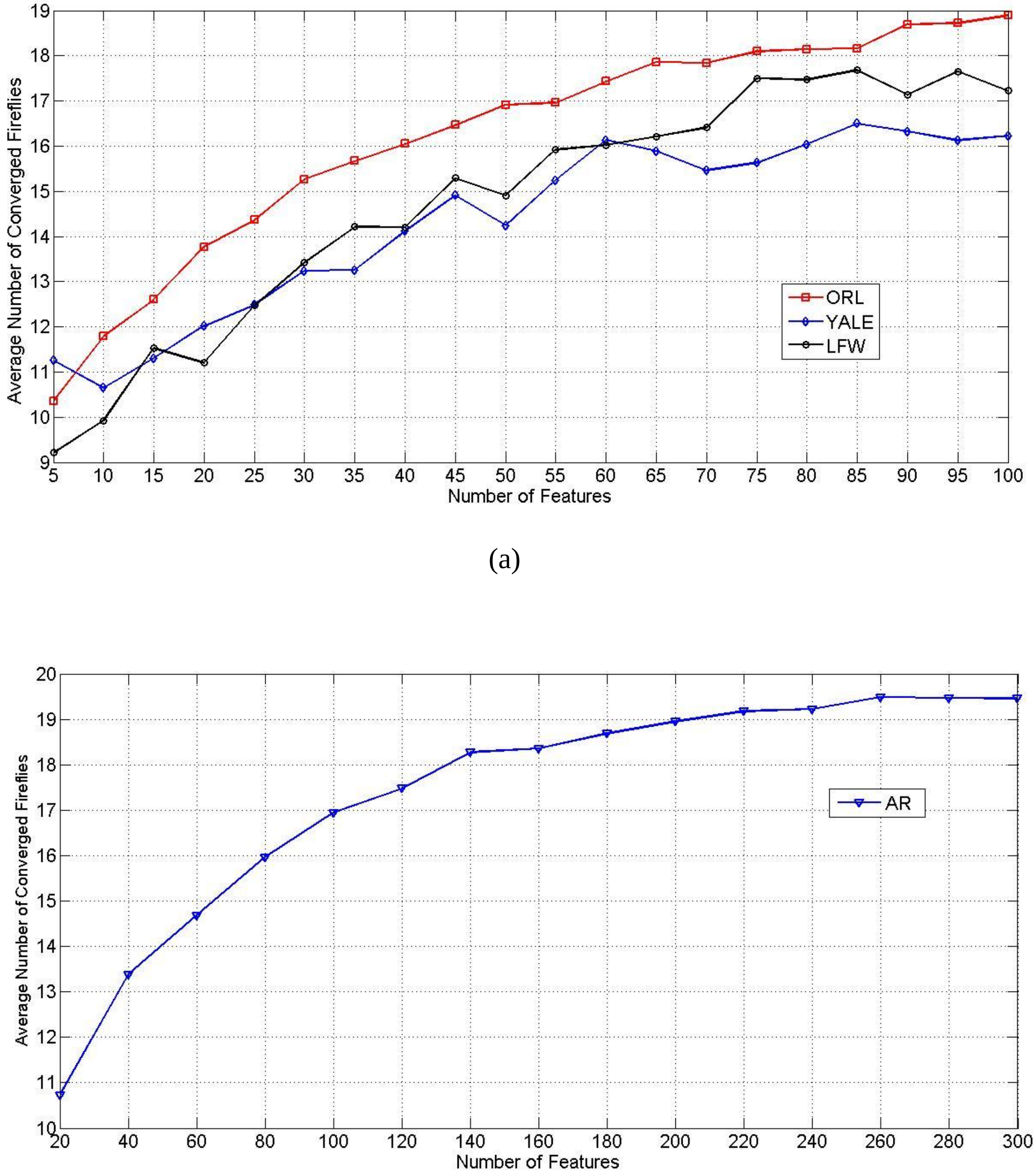


(c)



(d)

Fig. 4.13. (Continued). Effect of feature dimension on Average Recognition accuracy using the proposed FRBFNN (c) AR and (d) LFW face databases



(a)

(b)

Fig. 4.14. Effect of feature dimension on Average Number of Converged Fireflies using proposed FRBFNN (a) ORL, Yale and LFW Face Databases (b) AR Face Database

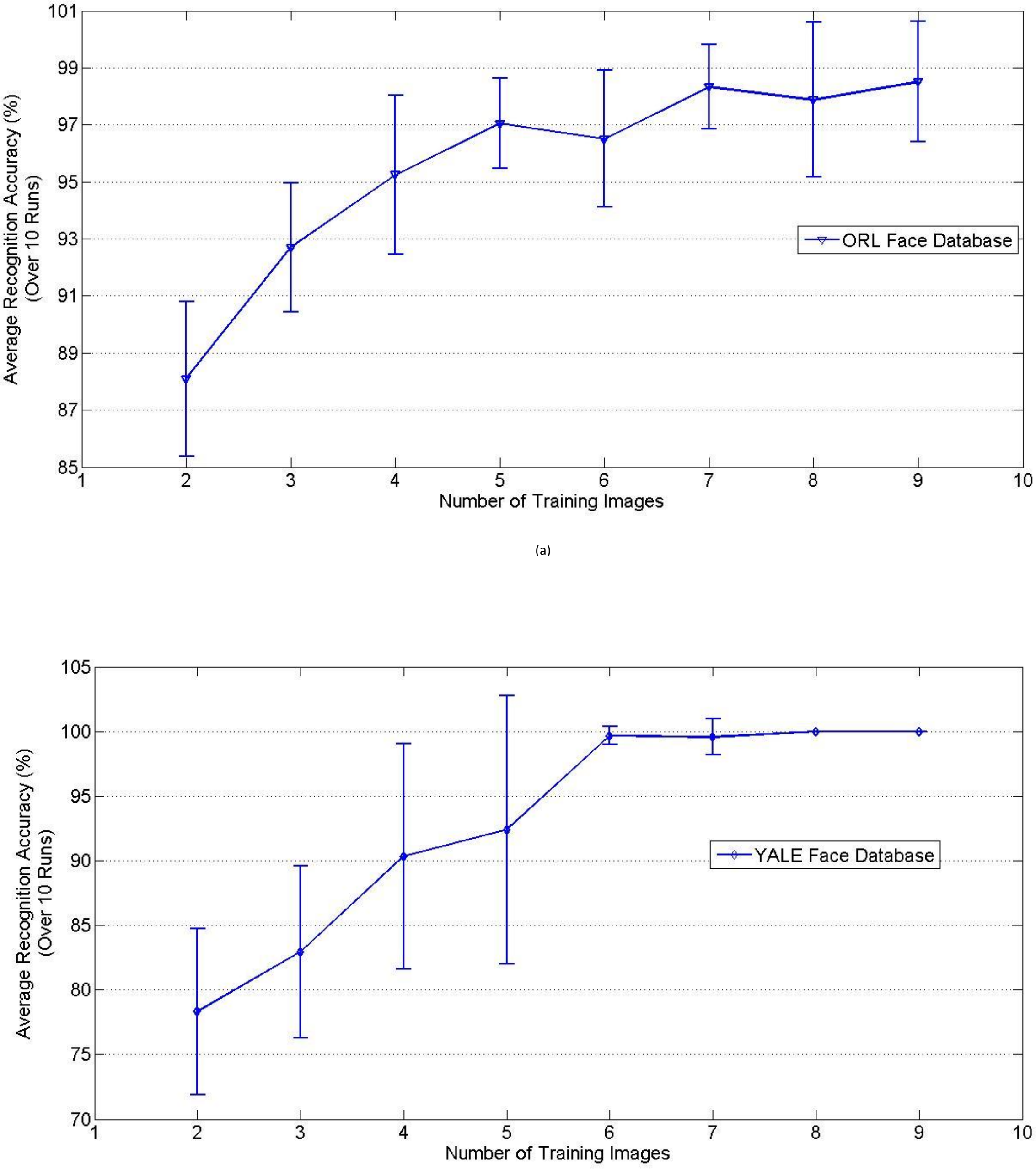
Table 4.6. Performance Evaluation of the Proposed FRBFNN Algorithm with respect to the Number of Features on ORL, Yale, AR and LFW face databases

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

4.5.2. Effect of Number of Training Images 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 Figure 4.15. It is observed that for ORL face database accuracy is maximum 98.50% (± 2.11) using 9 training images (Fig.4.15(a)). The standard deviation with 7 images is minimum among all observations for ORL face database. The maximum accuracy obtained for Yale face database is 100%(± 0.00) with 8 training images and similar observation is achieved with 9 training images (Fig. 4.15(b)).

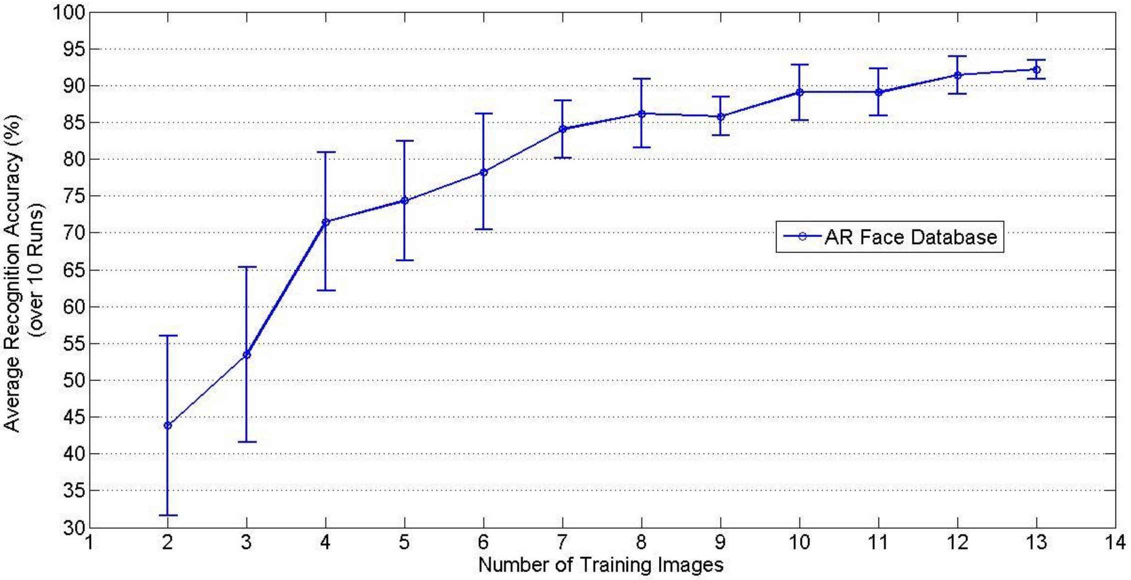
The trend visible in Figure 4.15(c) for AR face database is a continuous increase in average recognition accuracy and a continuous decrease in the standard deviation. The most suitable number of training images for AR face database is 13, having Average Recognition Accuracy as 92.15% (±1.23). LFW has huge variations (Fig. 4.15(d)), 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 increase in number of training images used.



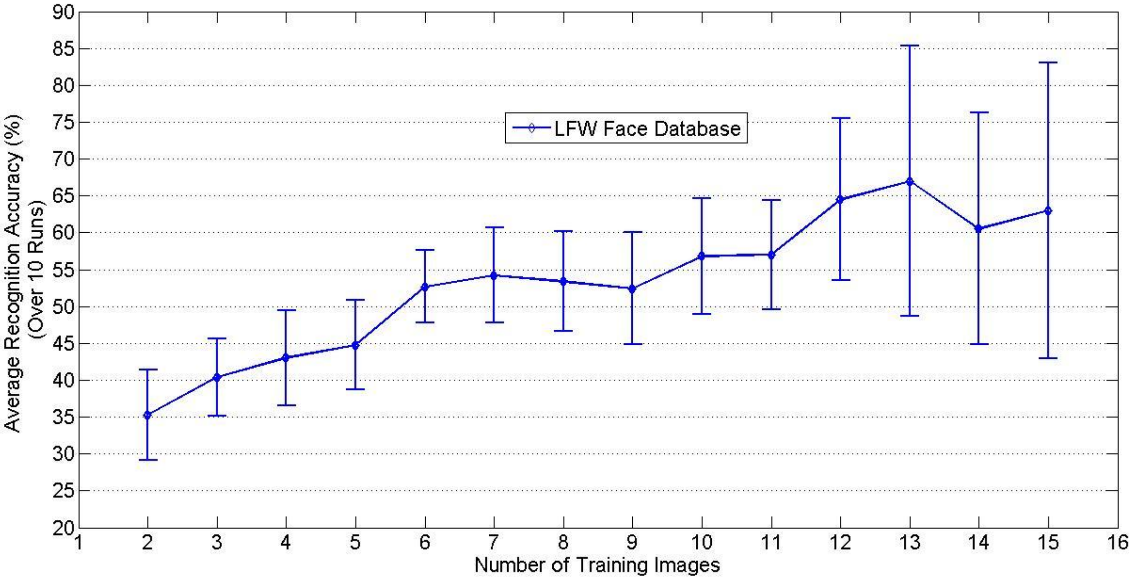
(a)

(b)

Fig. 4.15. Effect of Number of Training Images on Average Recognition accuracy using the proposed FRBFNN (with Standard Deviation Marked) (a) ORL (b)Yale face databases



(c)



(d)

Fig. 4.15. (continued) Effect of Number of Training Images on Average Recognition accuracy using the proposed FRBFNN (with Standard Deviation Marked) (c) AR and (d) LFW face databases

4.5.3. 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.4.16). Figure 4.17(a) shows the trend of evolved hidden neurons averaged over the total number of persons in the databases, with respect to the number of training images. This is evident from Figure 4.17(a) 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 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 Figures 4.17(b-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.

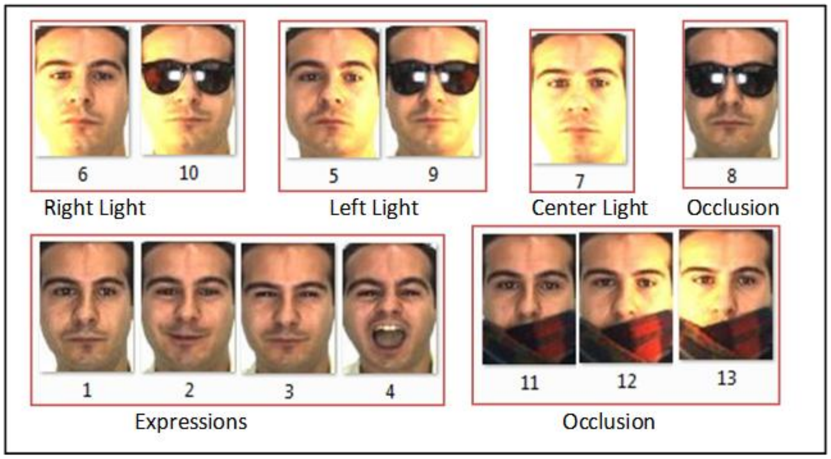
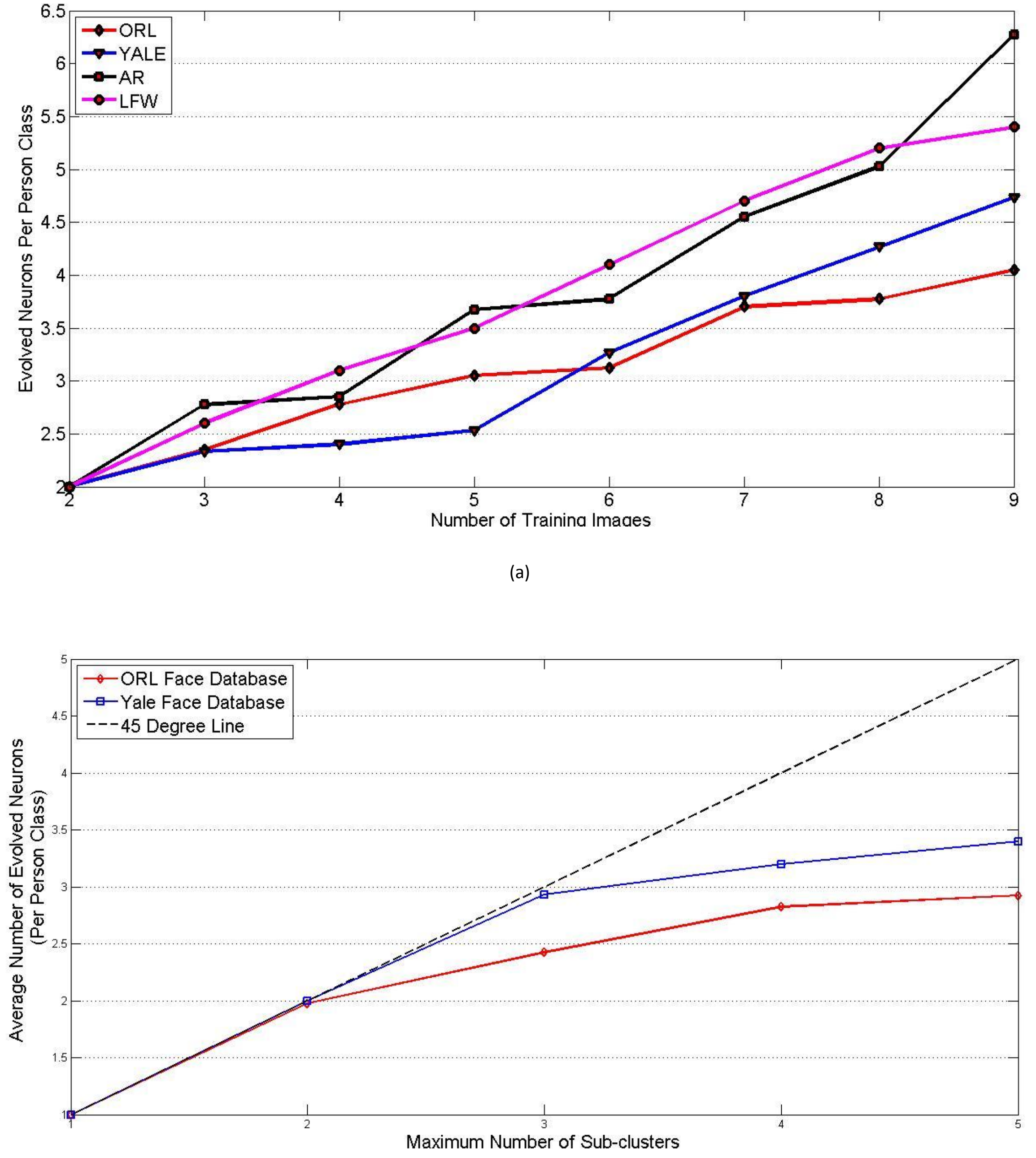


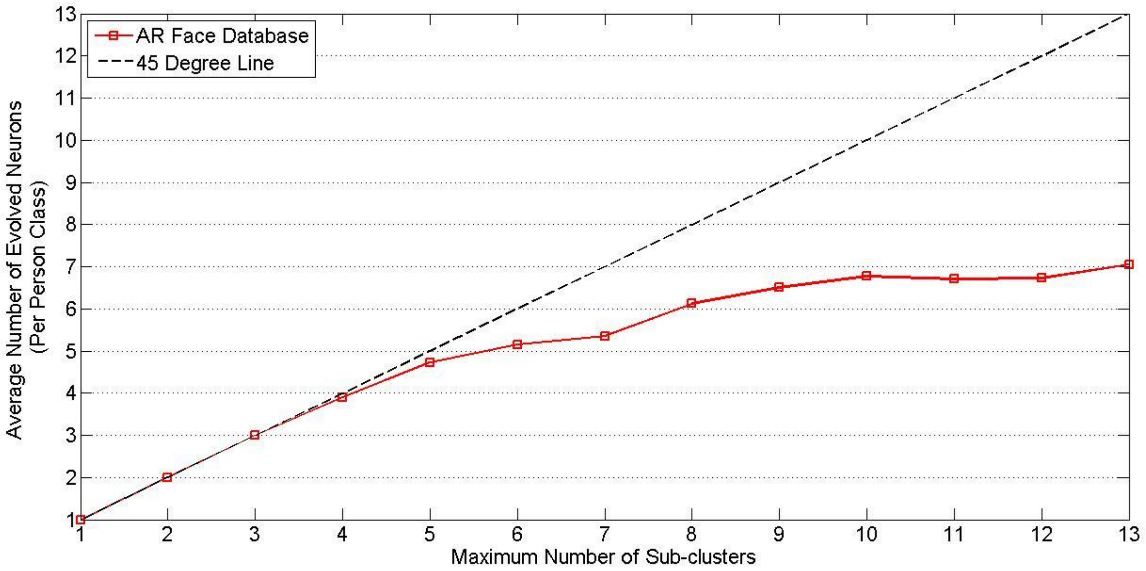
Fig 4.16. Sub-Clusters of a person class from AR face database forming different Hidden Neurons



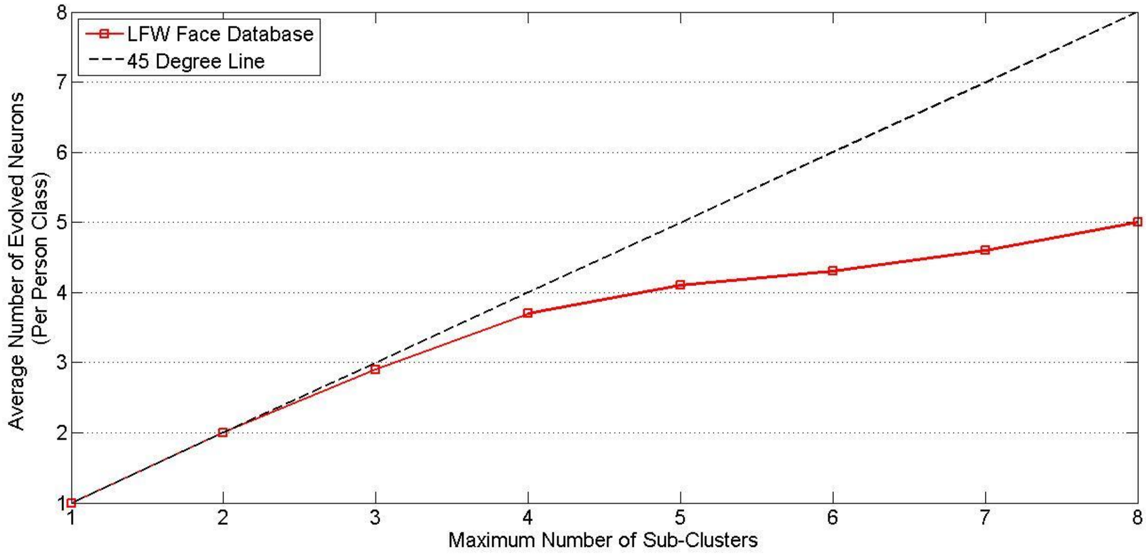
(a)

(b)

Fig.4.17. Average Number of Evolved Neurons for ORL, Yale, AR and LFW Face Databases (a) With Increased Number of Training Images (b) With Maximum Number of Sub-Clusters on ORL and Yale Face Databases



(c)



(d)

Fig.4.17 (Continued). Average Number of Evolved Neurons with Respect to the Maximum Number of Sub-Clusters for (c) AR (d) LFW Face database

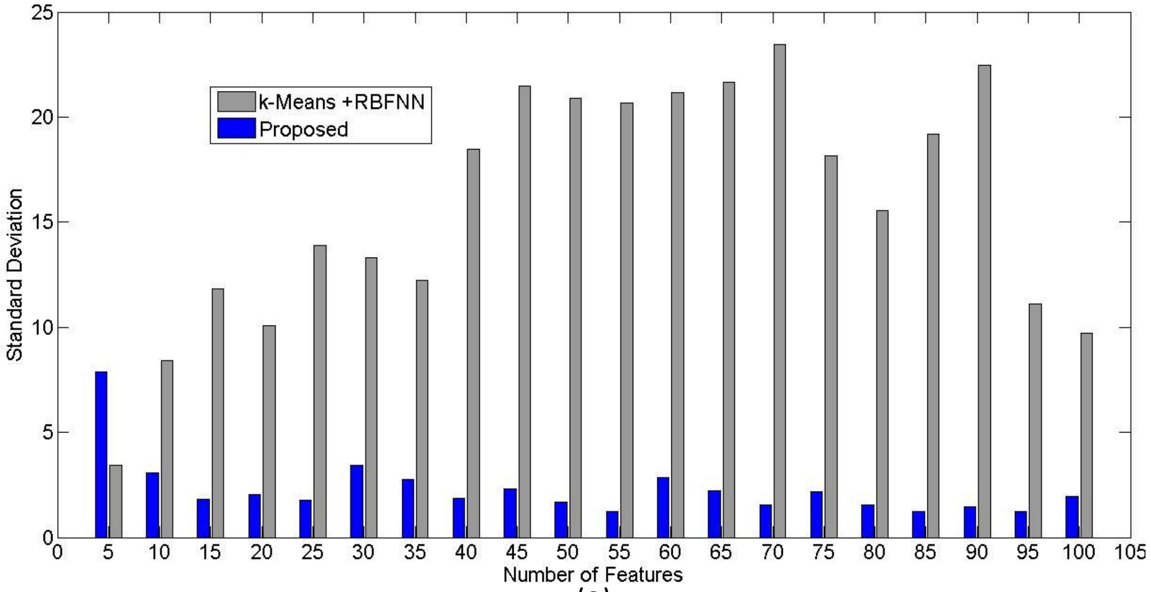
4.5.4. Comparison with other existing face recognition methods

In this section, performance of the proposed FRBFNN technique with the results reported in literature is compared.

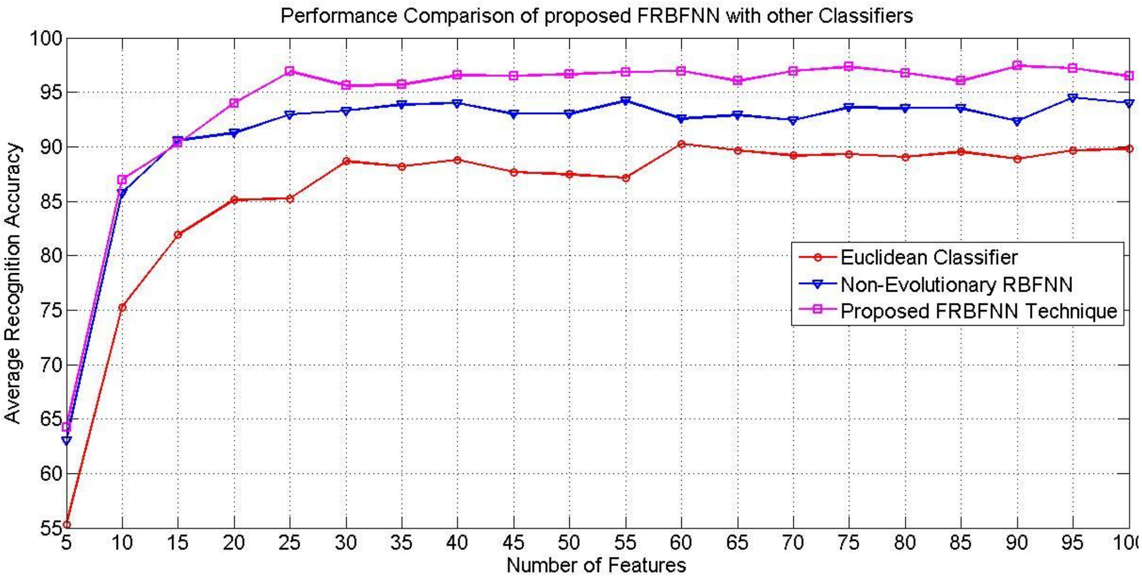
4.5.4.1. Performance Comparison on ORL Face Database

The performance of the proposed FRBFNN technique in terms of the standard deviation (across 10 independent runs) has been compared with the k-Means clustering based RBFNN as shown in Figure 4.18(a) for number of features varying from 5 to 100. It is observed that the proposed technique deviates very less from the average recognition accuracy than that of k-Means based method. Also, the performance of the proposed FRBFNN is superior than that of non-evolutionary classifiers such as Euclidean classifier and non-evolutionary RBFNN with 50 fixed centers (Fig. 4.18(b)).

The performance of the proposed FRBFNN in terms of average recognition accuracy is compared with the four techniques namely DCT + FLD based RBFNN *(Er. et al, 2005),* Point Symmetry Distance based RBFNN *(Sing et al, 2007)*, Self adaptive RBFNN *(Sing et al, 2009)* and polynomial RBFNN *(Oh et al, 2013)* and is shown in Table 4.7. The techniques are selected for comparison due to common attributes available for comparison. The number of training images used in all studies is 5, and the number of runs is 10 except in polynomial RBFNN based study where number of runs is 5. The proposed FRBFNN uses only 20 iterations for algorithm convergence resulting in 97.75% accuracy in ORL face database as against 6500 and 15000 iterations used in Point Symmetry Distance based RBFNN and Self Adaptive RBFNN respectively *(Sing et al, 2007; Sing et al, 2009)*. The number of learning iterations is 100 while another 20 generations are used for Differential Evolution used for optimizing design parameters in polynomial RBFNN *(Oh et al, 2013).*



(a)



(b)

Fig. 4.18. Comparison of (a) Standard deviations of the proposed FRBFNN and k-Means based RBFNN with respect to varying number of features (b) Average Recognition Accuracy of FRBFNN with Non-Evolutionary Classifiers

Table 4.7. Comparison of Performance of the Proposed FRBFNN Algorithm with existing techniques on ORL face database ('-' Information is not available)

Method	**Number of Samples per person**	**Number of Runs**	**Number of Learning Iterations/ Epochs**	**Number of Neurons evolved/ taken fixed**	**Number of Features**	**Average Recognition Accuracy (%)(ORL)**
DCT + FLD based RBFNN *(Er et al, 2005)*	5	10	-	Varied using parameter 1≤ α ≤ 2	30	97.55
Point Symmetry Distance based RBFNN (Sing et al, 2007)	5	10	6500	120 -Fixed	128	97.20
Self Adaptive RBFNN (Sing et al, 2009)	5	10	15000	152-Evolved	64	97.30
Polynomial RBFNN *(Oh et al, 2013)*	5	5	100 /20	-	-	95.25
Proposed FRBNN Technique	**5**	**10**	**20**	**127-Evolved**	**45**	**97.75**

The number of neurons evolved using the proposed FRBFNN is less, as compared to 152 neurons evolved using Self Adaptive RBFNN *(Sing et al, 2009)*. Point Symmetry based study *(Sing et al, 2007)* uses 3 clusters per individual resulting in 120 fixed neurons, while the proposed FRBFNN technique does not fix the number of clusters. Though the DCT + FLD based RBFNN *(Er et al, 2005)* performs with an average recognition accuracy of 97.55%, very close to the one produced by proposed FRBFNN algorithm which is 97.75%, the proposed technique displays its superiority in terms of its less computational overhead of feature selection. The DCT+FLD technique mentioned above 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 proposed 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 64 and 128 in studies by Sing et al *(Sing et al, 2007; Sing et al, 2009)* respectively.

The performance in terms of average number of iterations is also compared with the Fuzzy Hybrid Learning Algorithm(FHLA) *(Haddadnia et al, 2003)* for varying number of features is presented in Table 4.8. The proposed FRBFNN algorithm is run 40 times with randomly selected 5 training and 5 testing images at each run to compare average number of iterations (epochs). To simulate the conditions of FHLA, the number of features were 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.

Table 4.8. Comparison of Average Number of Iterations of FHLA technique with proposed FRBFNN technique on ORL face database

Number of Features	**Average Number of Iterations using FHLA Technique** ***(Haddadnia et al, 2003)***	**Average Number of Iterations using Proposed FRBFNN Technique**	**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

4.5.4.2. Performance Comparison on Yale Face Database

The performance of the proposed FRBFNN on Yale face database is compared with four techniques, namely, FHLA *(Haddadnia et al, 2003),* DCT +FLD based RBFNN *(Er et al, 2005),* IROLS bases RBFNN *(Wong et al, 2011*) and polynomial RBFNN *(Oh et al, 2013)* [Table 4.9]. 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.

Table 4.9. Comparison of Performance of the Proposed FRBFNN Algorithm with existing techniques on Yale face database

Method	**Authors**	**Average Recognition Accuracy (%)(YALE)**
FHLA based RBFNN	*Haddadnia et al, 2003*	99.75
DCT +FLD based RBFNN	*Er et al, 2005*	98.20
IROLS based RBFNN	*Wong et al, 2011*	95.0
Polynomial RBFNN	*Oh et al, 2013*	95.60
Proposed FRBNN Technique	**-**	**99.83 ± 0.53**

4.5.4.3. Comparison on AR Face Database

Two techniques namely KDCV *(Jing et al, 2008)* and IROLS *(Wong et al, 2011)* based RBFNN for face recognition are used in this comparison on AR face database [Table 4.10]. The IROLS based method gives recognition accuracy of 75.5% (± 5.7), while the proposed FRBFNN technique has recognition accuracy of 93.15% (± 3.25), demonstrating better recognition accuracy and stability of the proposed algorithm.

Table 4.10. Comparison of Performance of the Proposed FRBFNN Algorithm with existing techniques on AR face database

Method	**Authors**	**Average Recognition Accuracy (%)(AR)**
KDCV based RBFNN	*Jing et al, 2008*	85.36
IROLS based RBFNN	*Wong et al, 2011*	75.5
Proposed FRBFNN Technique	**-**	**93.15 ± 3.25**

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 4.7 - 4.10.

4.5.5. *Comparison of Training and Testing time for test databases*

The training time for images of the face databases ORL, Yale, AR and LFW with 200, 90, 520 and 80 training images are 10.92 seconds, 9.23 seconds, 59.85 seconds and 4.59 seconds respectively (Table 4.11). The testing time to test one face image for the above databases are 0.0048, 0.0027, 0.0265 and 0.0022 seconds respectively, which are considerably small, hence can be considered for real time face recognition.

Table 4.11: Training and Testing Time using the proposed FRBFNN Algorithm

Face databases	Number of training images	Number of testing images	RBFNN Hidden layer training time (seconds)	Testing time for each image (seconds)	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**

4.6. 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 RBF units in hidden layer. 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 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 values of parameter gamma (γ) for center selection, which contributes to fast convergence and improved performance are obtained as 5, 3, 0.00001 and 1 for ORL, Yale, AR and LFW respectively. The average recognition accuracy obtained in 10 independent runs for each of the four face databases as above are 97.35% (± 0.97), 99.83% (±0.53), 93.15% (± 3.25) and 52.50% (± 7.05) respectively using the above values of parameter gamma (γ). It is found that with values of gamma and number of fireflies, the proposed algorithm converges in as less as 6 to 9 iterations with improved face recognition performance. Also, a maximum of 20 iterations are required for any face database while a minimum of 11 fireflies and a maximum of 20 fireflies are required for the proposed algorithm to converge. The findings are novel in the face recognition problem domain and are also significant from the point of view of the complex non-linear hyper dimensional nature of face recognition problem.

The proposed algorithm outperforms various existing techniques reported in literature on all face databases used. The strength of the proposed FRBFNN algorithm lies in its capability to perform well with a very small number of features, resulting in dimensionality reduction of 99.5%, 99.7%, 94.8% and 99.4% for ORL, Yale, AR and LFW face databases respectively. Also the numbers and centers of RBF units evolve on their own depending on the complex underlying variations of pose, illumination, expressions and occlusion. The proposed algorithm is fast as it takes only 10.92 seconds, 9.23 seconds, 59.85 seconds and 4.59 seconds to train the proposed classifier using 200, 90, 520 and 80 face images of ORL, Yale, AR and LFW face databases respectively. Also, the overall testing time for each test image taken from ORL, Yale, AR and LFW face databases are 0.0048 seconds, 0.0027 seconds, 0.0265 seconds and 0.0022 seconds respectively. We therefore conclude that the proposed FRBFNN technique is efficiently capable of handling variations in face images in real time.

Chapter 5

Radial Basis Function Shape Estimation Algorithm and Integrated Classifier Performance Evaluation

5.1. Introduction

Design of RBFNN has been of research interests to many researchers working in the area of function approximation, machine learning and various other engineering applications due to the best approximation and generalization capabilities of RBFNN. The design not only includes obtaining the optimal number and centers of the hidden layer neurons of RBFNN, it also includes the optimal shape of basis functions used in hidden layer. The shape of the basis functions mainly refers to their spread (width) sensitive to the points closer to the respective centers while overlapping between the neighboring basis functions is important to avoid leaving any point unclassified. In this chapter, a shape parameter estimation technique based on overlapping factor using Gaussian basis functions is proposed.

It is understood that the face data is hyper-dimensional (d >> 1) and is non-linearly separable in input space, that is, despite using good feature selection techniques, it is difficult to separate the face data of different persons using only linear decision boundaries. The basis functions at the hidden layer perform a mapping in a k-dimensional hidden space, where k is the number of neurons in the hidden layer and $k \gg d$. The two class data is represented for two persons in Figure 5.1. The decision boundary in hidden space is a hyper-plane and is able to classify the given test patterns of face data correctly. The centers of RBF units are computed as the mean feature vector of the training images in each sub-clusters correspondingly. The distance of the farthest point in the sub-cluster from its mean is taken as the radius of each RBF unit.

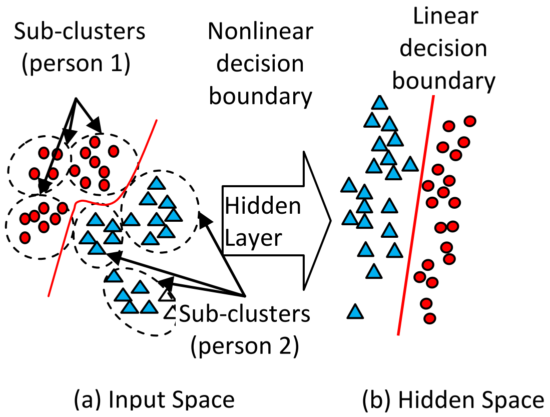


Fig. 5.1: Non-linear mapping using RBFNN hidden layer

In this chapter, the effect of basis functions shape on face recognition performance of Radial Basis Function Neural Network (RBFNN) is investigated and a technique for shape of radial basis functions namely OLAF (Based on **O**ver**LA**pping **F**actor) is proposed. The proposed OLAF algorithm uses Gaussian basis functions and the spread of sub-clusters of each person class is controlled by an overlapping parameter α. The proposed OLAF technique is adaptive as it uses the relative distances between the centers of natural sub clusters of the two different classes and also the maximum span or radius of the sub clusters to estimate the overlapping between the neighboring basis functions. The amount of spread therefore is proportional to the radius of the sub-cluster and the distance of the nearest sub-cluster belonging to a different class. The choice of basis functions has also been investigated in the context of face recognition. The proposed OLAF technique is adaptive and has been integrated with the proposed Center selection technique FRBFNN, proposed in Chapter 4 to obtain an Integrated Face Recognition Classifier. In this chapter, the proposed classifier's performance is assessed using various performance measures such as Sensitivity and Specificity, Confusion Matrix and Receiver Operating Characteristic (ROC) Curve.

5.2. Types of Radial Basis Functions

A variety of radial basis functions such as Gaussian, Multiquadric, Inverse multiquadric, Thin plate Spline functions etc. exist in the literature. These basis functions are functions of the radial distance *r* of the data *x* from the centers c_j where *r* is defined as

$$r = \|x - c_j\| \tag{5.1}$$

The distance can be of any metric such as Euclidean, Mahalanobis, Manhattan distance etc. Some of the radial basis functions commonly used are given below. Parameters used in these basis functions σ, *c* and *k* are called as shape parameters and the value of $\varphi(r)$ is the response of the basis function for a point *x* at distance *r* from center c_j .

- Gaussian Basis Function:

$$\varphi(r) = e^{-\frac{r^2}{2\sigma^2}} \tag{5.2}$$

where σ is known as the spread of the Gaussian function. The varying spread sizes and the corresponding basis function shapes are shown in Figure 5.2.

- Generalized Multiquadric Basis Function:

$$\varphi(r) = (c^2 + r^2)^{\frac{\beta}{2}} \tag{5.3}$$

where β is any odd integer.

- Hardy's Multiquadric Basis Function

$$\varphi(r) = (c^2 + r^2)^{\frac{1}{2}} \tag{5.4}$$

- Inverse Multiquadric Basis Function:

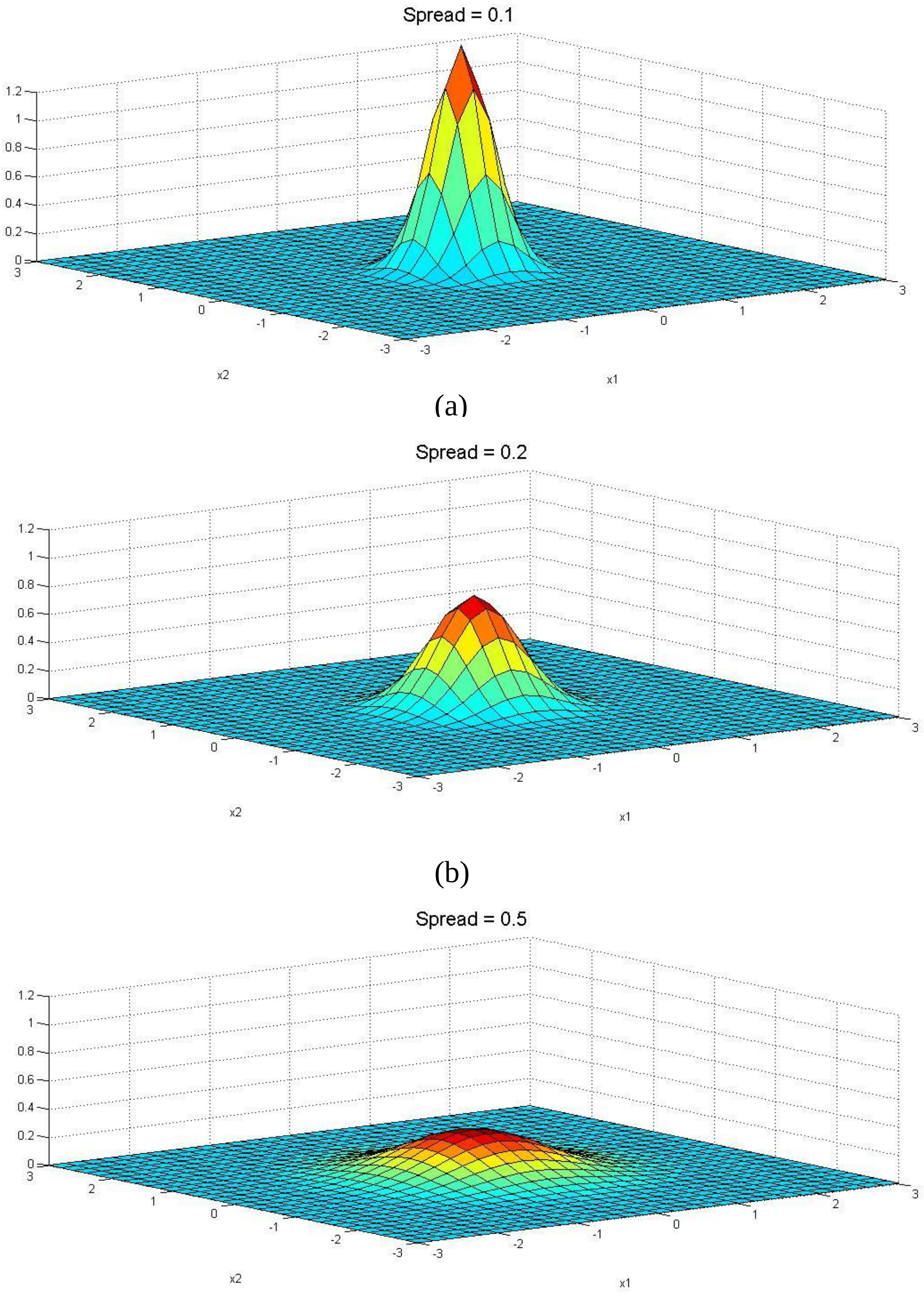
$$\varphi(r) = \frac{1}{(c^2 + r^2)^{1/2}} \tag{5.5}$$

- Thin Plate Spline :

$$\varphi(r) = r^2 \ln r \tag{5.6}$$

- Cubic Spline $\varphi(r) = r^3$ (5.7)

A generalized Multiquadric (MQ) basis function ϕ(r) is defined as a function of radial distance *r* and uses two parameters c^2 and β. The Cubic spline and Thin plate Spline methods are not dependent on any shape parameters.



(a)

(b)

(c)

Fig. 5.2: Gaussian Basis Function with varying Spread (σ) values (a) Spread = 0.1 (b) Spread = 0.2 (c) Spread = 0.5

5.3. Radial Basis Function Shape and Its Significance

A basis function is sensitive to the points in the vicinity of its center and produces large response for the points close to its center as compared to the points away from its center. The spread of a two dimensional Gaussian function, shown in Figure 5.3, is defined by the covariance matrix Sigma (σ) defined as $\begin{bmatrix}0.5 & 0.3\\0.3 & 0.5\end{bmatrix}$ with mean taken as μ = [0 0]. The shape of basis function becomes flat as the spread increases and thus the capability of the basis function to sense a point farther from its center increases. For example, the shape of the Gaussian basis function using Sigma (σ) defined as $\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}$ with mean μ = [0 0] is comparatively flat as is shown in Figure 5.4. In face recognition using RBFNN, the basis functions are expected to be highly sensitive to face images lying close to its center and generate response large enough so as to recognize the given test image correctly. At the same time the same basis function is required to be less sensitive to seen (trained) or unseen (testing) face image points belonging to any other person.

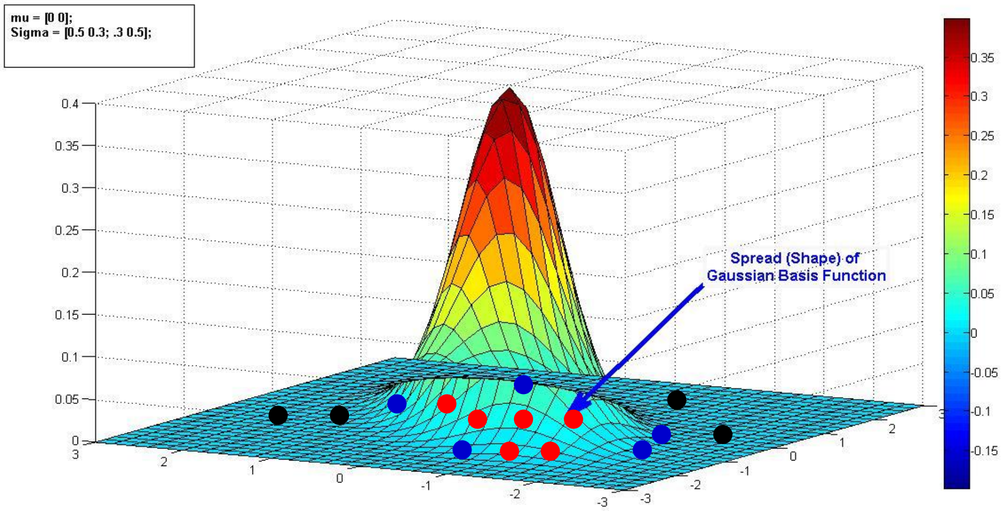


Fig. 5.3. Gaussian Basis Function Spread and its sensitivity to points *(Red colored points are captured with highest response value, Blue colored dots are captured with very low response values and the Black colored dots are not captured)*

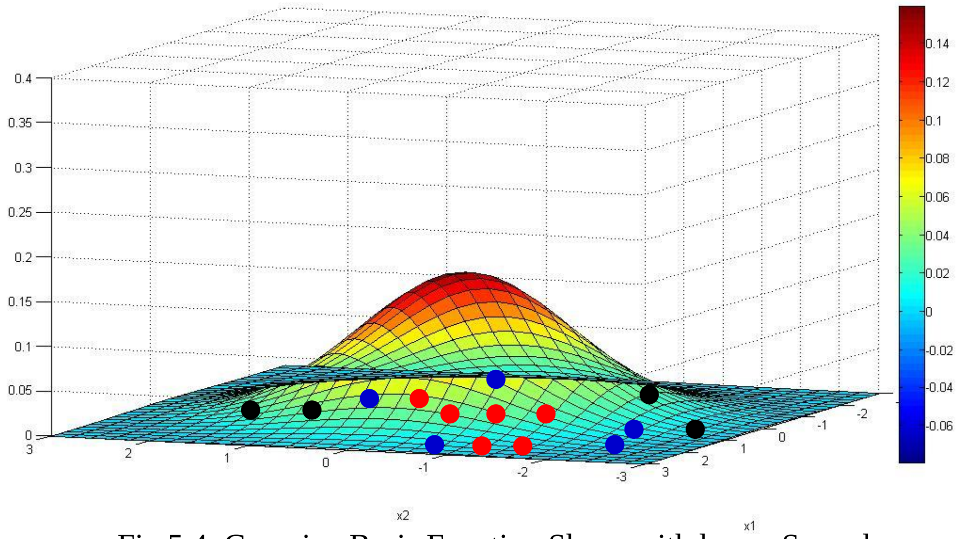
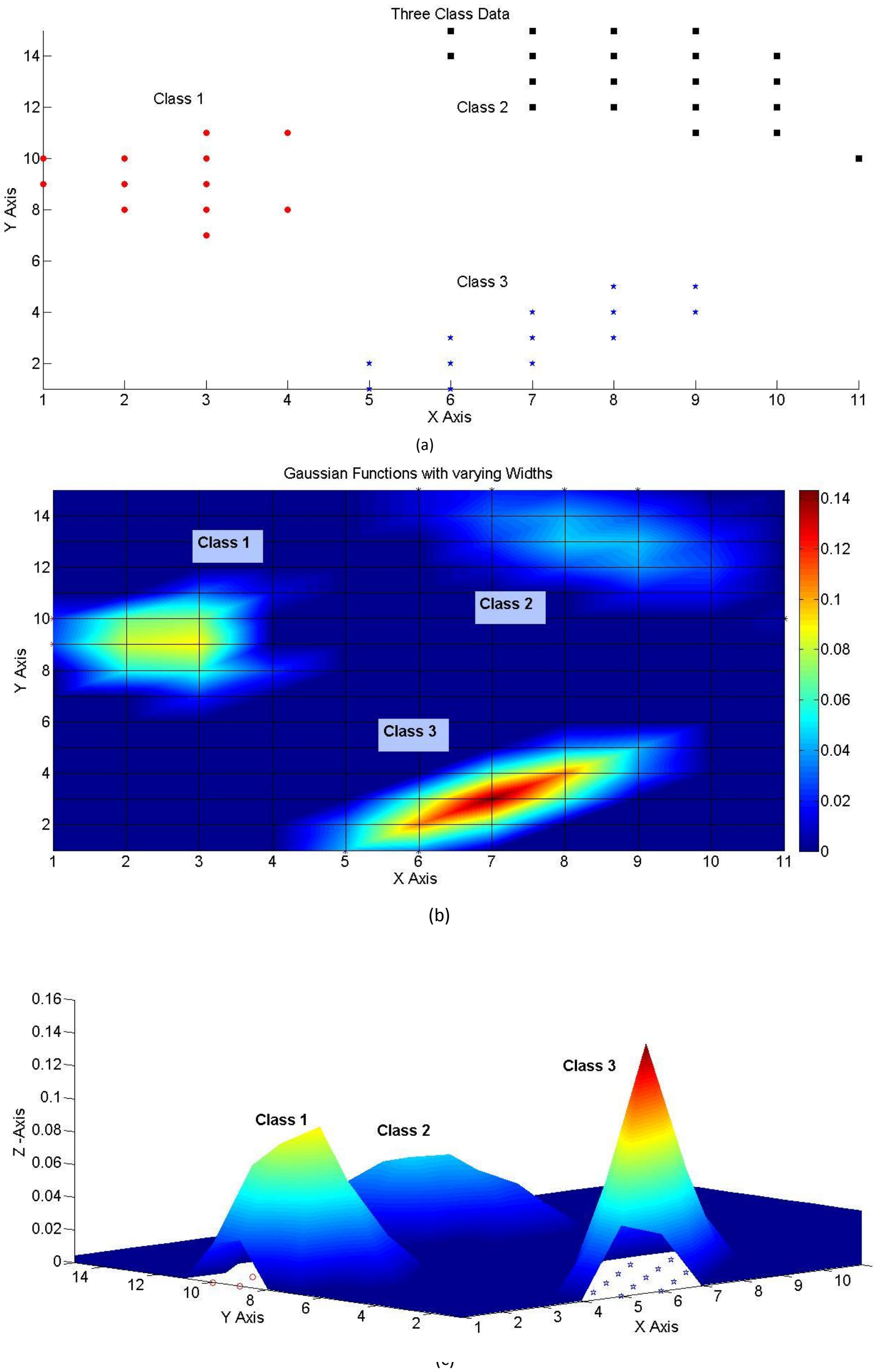


Fig.5.4. Gaussian Basis Function Shape with larger Spread

The basis function (say φ_j) at jth node of the hidden layer corresponding to a person class is required to be sensitive only to points belonging to this class and contributes heavily to the response for receiving the correct output, while at the same time, the above basis function should also be insensitive to the points belonging to other classes contributing less for test samples of other classes. To illustrate the significance of shape of a basis function, training points belonging to three classes in a two dimensional space are shown in Figure 5.5(a). The training points belonging to classes 1, 2 and 3 are represented in Red, Black and Blue colours respectively. For testing the belongingness of a test feature vector, it is required to compute the centers and widths (spreads) of the three basis functions. The Widths of Gaussian functions sensitive to each class from 2-dimensional three class point data are computed and shown in Figure 5.5(b). Wider is the range of the class data, flatter is the surface of Gaussian basis function as is shown in Figure 5.5(c).

A test data point which lies under the surface of the Gaussian function for class j is labelled with the class identity accordingly. If the test data falls on the flat region away from the Gaussian surfaces of the representative classes, then the sum of responses of all basis functions, i.e. $\sum_{j=1}^{j=3} \emptyset(\|x - c_j\|)$ is less than a threshold, indicating that the test data cannot be classified and is treated as impostor. These surfaces are required to be overlapping with the neighbouring basis functions to be able to respond to the data. The amount of overlapping can be controlled using some parameters. More is the overlapping, wider is the basis function, resulting in flatter surfaces.



(a)

(b)

(c)

Fig.5.5 : Gaussian Functions Width for Three Class Data (a) Three Class Data with Varying Span (b) Gaussian Functions of varying widths sensitive to data from each class (c) The bell shaped Gaussian surfaces for each class

The centers of the RBFNN were obtained as the class means using the supervised information about the classes *(Er et al, 2002)* in which the authors defined the basis function shape using the distance of the furthest point from the cluster mean (d_k) and the distance between center of the cluster and its nearest cluster's center $d_{min}(k,l)$. The width (signifying shape) of the k^{th} basis function is defined as follows

$$\sigma^k = \max\left(\sigma_W^k, \sigma_B^k\right) \tag{5.8}$$

where σ_W^k and σ_B^k signify the intra-data distribution and inter-data variations and are given as

$$\sigma_W^k = \frac{d_k}{\sqrt{|\ln \beta|}} \tag{5.9}$$

and

$$\sigma_B^k = \eta \times d_{min}(k, l) \tag{5.10}$$

Where β and η are confidence coefficient and overlapping factor respectively, the values of β and η normally ranging in intervals as $0.5 \le \beta < 1$ and $0.7 \le \eta \le 1.8$ respectively.

Face recognition using point symmetry distance based RBF network and adaptive RBFNN for high speed face recognition were used by Sing et al in two different studies *(Sing et al, 2007; Sing et al, 2009)*.The spread or the width of the j^{th} cluster is defined by the authors as follows

$$\sigma_j = \beta \times \left\|c_j - c_i\right\| \tag{5.11}$$

where β is a constant ($1 \le \beta \le 3$) which controls the amount of overlap between the Gaussian basis functions.

Shape parameters of the Radial basis functions improve the accuracy of approximation of high non-linear problems.

5.4. Proposed Overlapping Factor based Shape Estimation (OLAF) Technique

In this section, an algorithm for shape parameter estimation, based on the overlapping factor of the spread of the Gaussian basis functions is proposed. The objective of finding the most appropriate shape of the radial basis function is to improve generalization in terms of identifying an unseen face image correctly. This image may fall as a point in the region close to the sub-clusters of the actual person class. The amount of overlapping creates the regions of decision boundary. If a test image near to the center is within this region, then the corresponding basis function generates a response and contributes largely for correct recognition of the test image. If the test image lies outside the decision boundary, then the response is minimal and the test image is misclassified. While it is important to have decision boundary which is sensitive to the unseen test faces, it is also important that the region within decision boundary does not respond to the test samples belonging to other classes.

The proposed technique uses the distance between the center of natural cluster of a person's images from another person's cluster of training feature vectors (visualized as d-dimensional points) and proportionately increases or decreases the spread of the basis function. The proposed technique thus arriving at varying spread values for each cluster outperforms some existing techniques using different methods for computing spread.

Training face images are used to construct d-dimensional feature vectors mapped as points in d-dimensional space. The sub-clusters are formed using FRBFNN proposed and described in Chapter 4. Let S_i^j be the *i*th sub-cluster of class *j* and M_i^j be the total number of training face image feature vectors in S_i^j shown as small triangles in red colours in Figure 5.6. Let the total number of classes be L and the total number of sub-clusters in class *j* be N_j. The radius d_i^j of S_i^j is computed as the distance of the furthest point from its center c_i^j. Let each sub-cluster S_i^j consists of T_i^j training images.

$$d_i^j = \max_{1 \leq \mathrm{m} \leq \mathrm{T}_\mathrm{i}^\mathrm{j}} \|\mathrm{x}^\mathrm{m} - \mathrm{c}_\mathrm{i}^\mathrm{j}\| \tag{5.12}$$

where $\|.\|$ is the distance measure and can be computed using Euclidean or Mahalanobis distance.

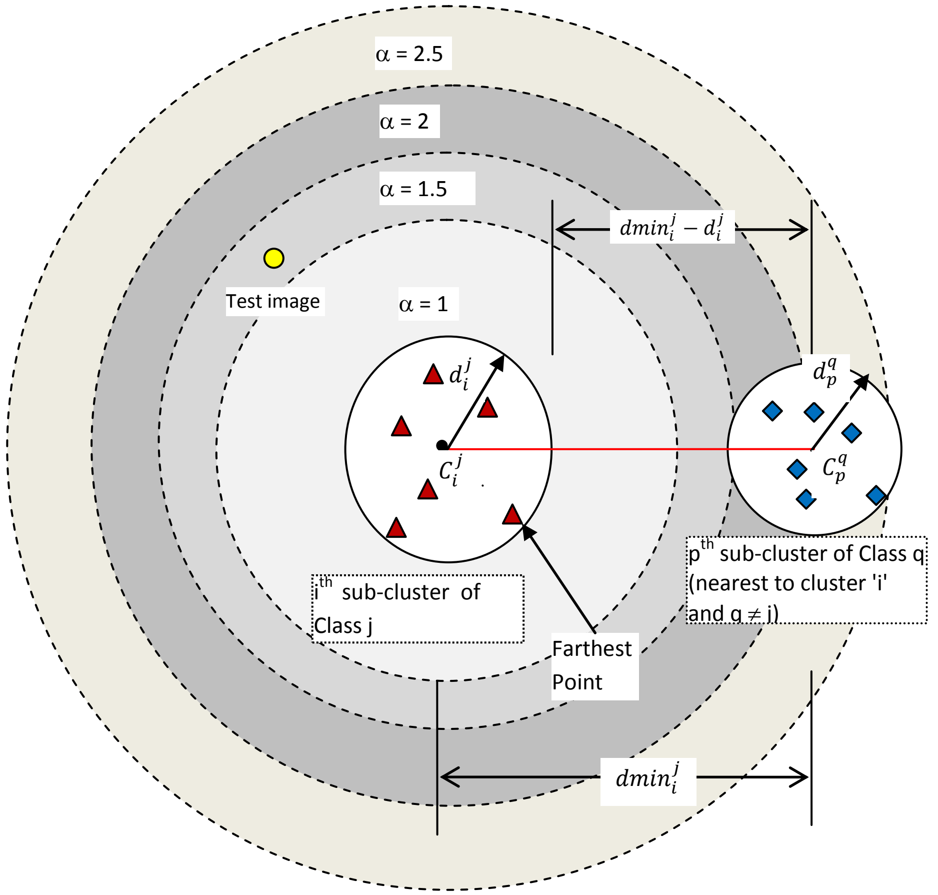


Fig. 5.6. Schematic of the proposed Shape Estimation Technique OLAF

The m^{th} training image in sub-cluster S_i^j is represented by the feature vector x^m which is a d-dimensional vector obtained using appropriate feature extraction method. The distance $dmin_i^j$ between the centers of the two sub-clusters belonging to different classes is defined as

$$dmin_i^j = \min_{\substack{1\leq q\leq L \\ q\neq j \\ 1\leq p\leq N_q}} \left\| c_p^q - c_i^j \right\| \tag{5.13}$$

The spread or the width σ_i^j of Gaussian function correspondingly is defined as

$$\sigma_i^j = d_i^j + \left(\frac{dmin_i^j - d_i^j}{2} \right) \times \alpha \tag{5.14}$$

where $\alpha > 0$ is the overlapping factor. The performance of RBFNN depends on the spread as it defines the shape of the Gaussian function capable of sensing the test data falling within its spread located near to its center. The proposed OLAF algorithm is described in Figure 5.7.

```
INPUT: Sub-clusters and means of each Person Class obtained using FRBFNN (Chapter 4)
    Step 1.    for each class j, $\forall$j=1,2,.., L repeat steps 2-5.
    Step 2.    for each sub-cluster $S_i^j$,$\forall$i=1,2,.., $N_j$ repeat steps 3-5.
    Step 3.    Compute the radius $d_i^j$ of ith sub-cluster of jth class using equation (5.12)
    Step 4.    Compute the minimum distance $dmin_i^j$ between $c_i^j$ and $c_p^q$ using (5.13),
               where q =1,2,..L; and $\forall$p =1,2,..$N_q$ where q$\neq$j
    Step 5.    Compute spread $\sigma_i^j$ of sub-cluster $S_i^j$ using equation (5.14)
OUTPUT: Spread $\sigma$ for Gaussian basis function for all RBF nodes
```

Fig. 5.7: Proposed Shape Estimation Algorithm OLAF

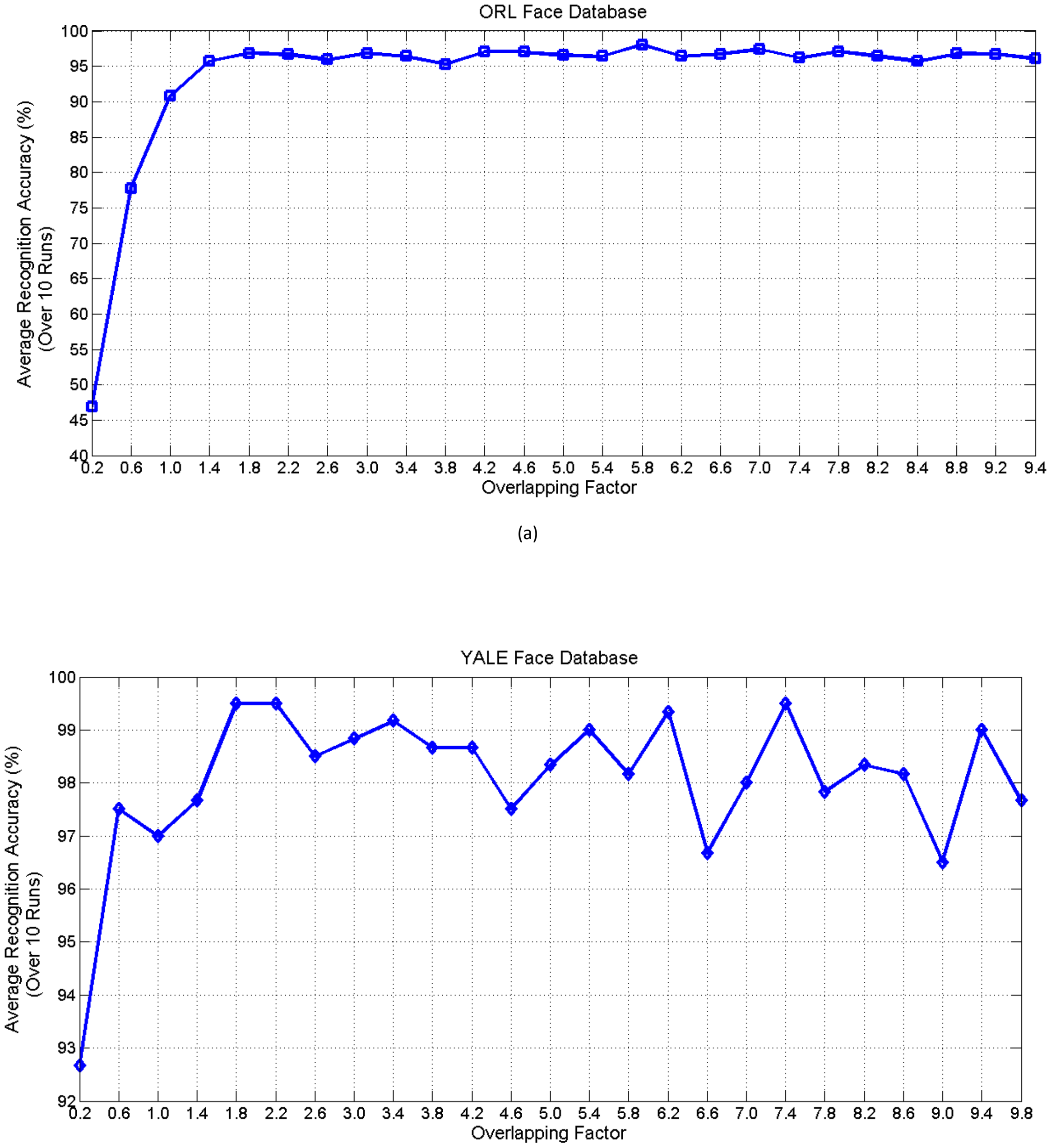
5.5. Results and Discussion

5.5.1. *Effect of Overlapping Factor on Average Recognition Accuracy*

The spread of the sub-clusters were computed using equation (5.14) for varying values of the overlapping factor α. The effect of α is investigated on the average recognition accuracy on ORL, Yale, AR and LFW face databases [Fig. 5.8]. In this experiment, all classes are considered as authorized. The Average Recognition Accuracy here refers to the definition given in equation (4.40). The feature dimensions for ORL, Yale, AR and LFW face databases are 45, 30, 260 and 65 respectively. The basis for the choice of number of features here is number of features reported in Table 4.6 (Chapter 4) . Also values of gamma (γ) are used as were proposed in Table 4.6 (Chapter 4) while number of fireflies are taken as 20. The FRBFNN algorithm proposed in section 4.3 (Chapter 4) is used to obtain the hidden layer neuron centers, i.e. RBF unit centers. The maximum number of iterations i.e. 20 is used to allow the algorithm to converge. The training images used for ORL, Yale, AR and LFW are 5, 6, 13 and 8 respectively.

The best performance of the proposed shape estimation algorithm is observed as 98.05% (± 1.17) for ORL face database using $\alpha = 5.8$ [Fig.5.8(a)]. The average recognition accuracy for Yale face database is 99.50% (± 0.81) at $\alpha = 1.8$ after which the performance fluctuates [Fig.5.8(b)]. The average recognition accuracy is 89.25% (± 2.15) at $\alpha = 9.8$ for AR face database. It is evident from Figure 5.8(a) that the average recognition accuracy curve for ORL face database is stable after $\alpha = 1.8$ and ranges within 96.00% - 98.05% . It is observed that the performance of AR face database improves as the overlapping factor α increases, reaching maximum of 90.0% at $\alpha = 9.8$ [Fig.5.8(c)]. The effect of overlapping factor on average recognition accuracy measured for LFW face database is observed and it is obtained as 52.0% (± 9.0) at $\alpha = 4.6$. The performance does not improve even after increasing the overlapping factor for LFW face database[Fig.5.8(d)].

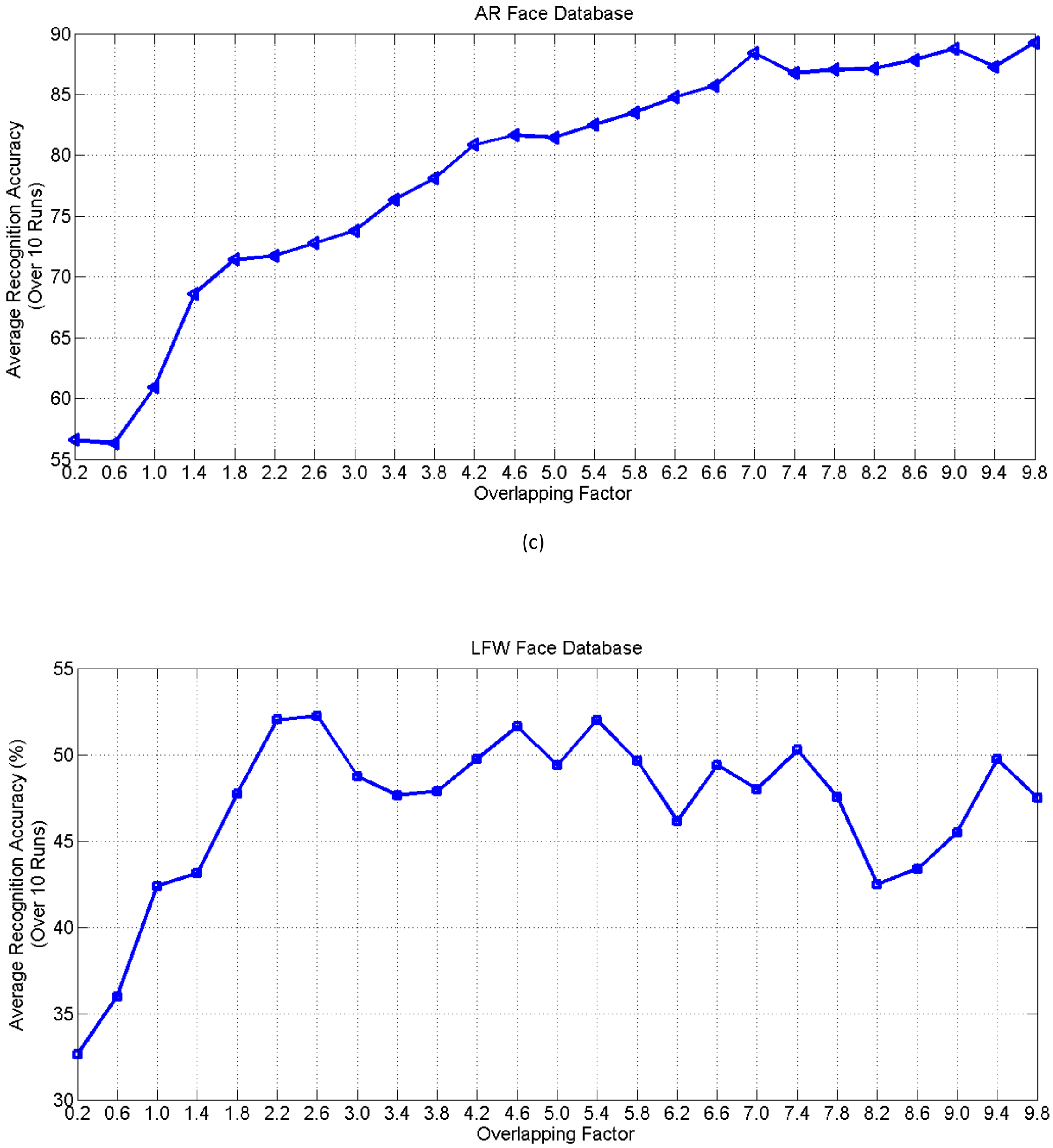
Figure 5.8 thus depicts the significance of overlapping factor on overall performance of the fact recognition system.



(a)

(b)

Fig. 5.8. Effect of Variation of overlapping factor α on Average Recognition Accuracy on (a) ORL Face (b) YALE



(c)

(d)

Fig. 5.8. (Continued). Effect of Variation of overlapping factor α on Average Recognition Accuracy on (c) AR (d) LFW face databases

5.5.2. *Comparison of the proposed OLAF technique with performance using Fixed Sized Spread for all basis functions*

The performance of the proposed integrated classifier using varying spread values of the basis functions obtained by the proposed OLAF technique is compared with the performance of the RBFNN using constant shapes for all basis functions. The Receiver Operating Characteristic (ROC) curves are plotted using varying overlapping factors in the range of 0.5 to 10.0 [Fig.5.9]. It is observed that the ROC curve corresponding to the proposed OLAF technique, which uses the relative distance between other RBF units and computes shapes for all basis function, performs better than the RBFNN using fixed sized spreads for all basis functions irrespective to the distances between the RBF units. The ROC curve in Fig.5.9 displays superiority of the proposed OLAF technique in terms of its closeness boundary of the ROC space as compared to the ROC curve obtained using fixed sized spread.

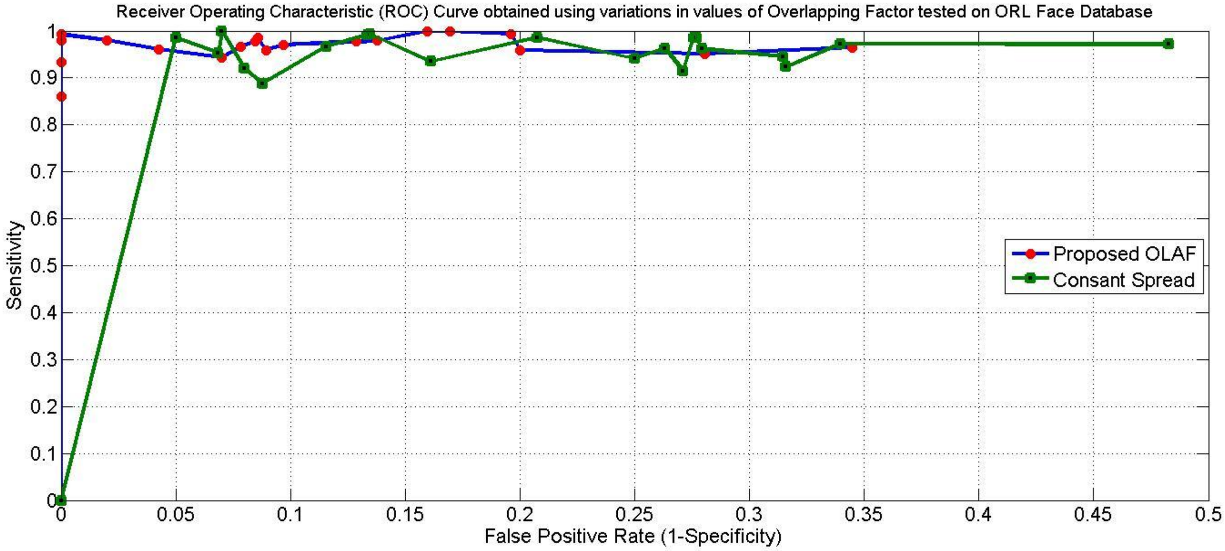
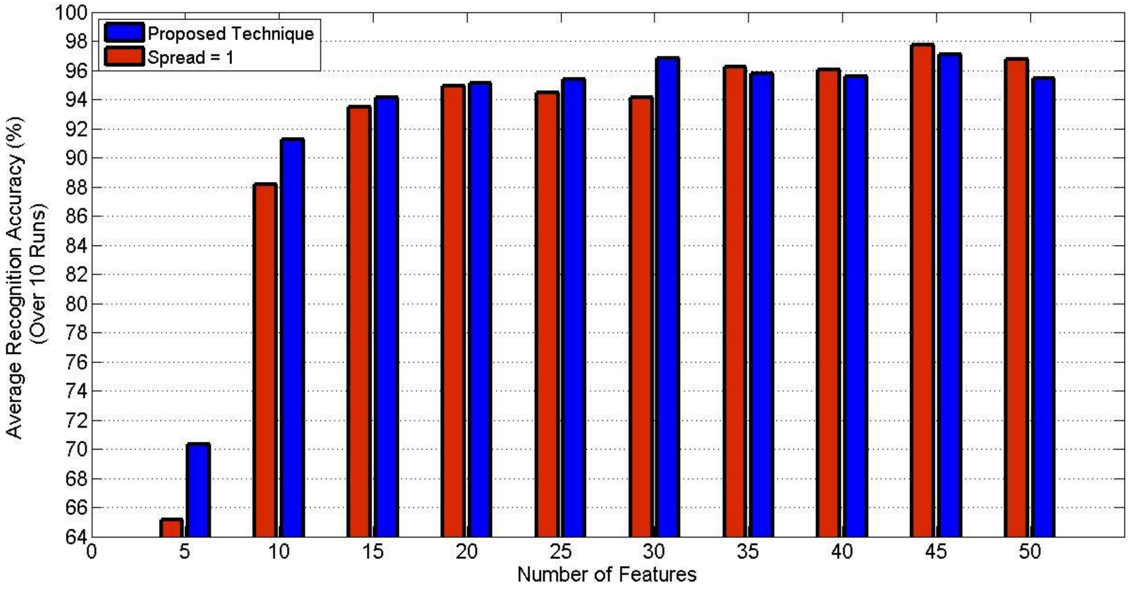
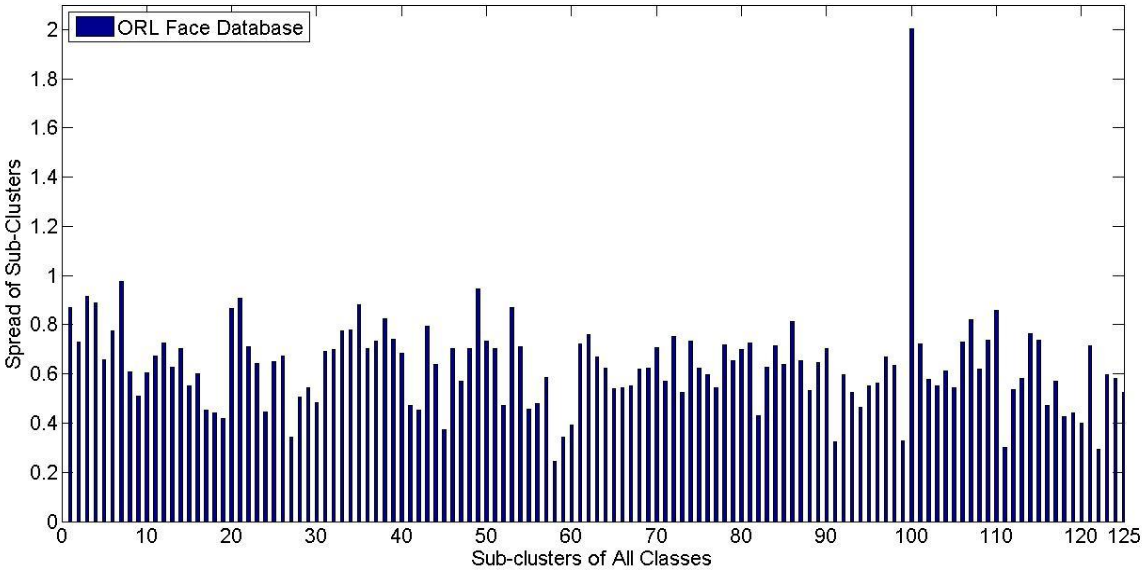


Fig. 5.9. Receiver Operating Characteristic (ROC) Curve with variations in Overlapping Factor [0.5 - 10.0]

In another independent experiment, the comparison of the above two techniques is performed using varying feature vector dimensions. The proposed shape estimation technique OLAF performed better than the constant spread for 5≤ d ≤ 30 using overlapping factor $\alpha = 5.8$ (Fig.5.10(a)]. Varying spread sizes of the evolved 125 RBF nodes (shown on x-axis) for 40 person classes in ORL face database using 5 training images is shown in Fig.5.10(b).



(a)



(b)

Fig.5.10 (a) Comparison of the Average Recognition Accuracy produced on ORL face database by constant spread of value 1 for all sub-clusters and that produced by the Proposed OLAF technique (b) Varying Spreads using the proposed OLAF

5.5.3. *Comparison of the proposed OLAF Technique with Existing Techniques*

The performances of the proposed OLAF algorithm is compared with some of the existing techniques and it is observed that the proposed technique outperforms them [Tables 5.1 - 5.3]. The proposed technique outperforms some other existing techniques reported in literature with accuracies obtained for both ORL, Yale and AR face databases as 98.05% , 99.50% and 89.25% respectively.

Table 5.1. Comparison of the Proposed (OLAF) Technique with some of the existing techniques on ORL Face Database

Method	Accuracy(%)
DCT+FLD based RBFNN *(Er et al, 2005)*	97.55
Point Symmetry Distance based RBFNN *(Sing et al, 2007)*	97.20
Self Adaptive RBFNN *(Sing et al, 2009)*	97.30
Polynomial RBFNN *(Oh et al, 2013)*	95.25
Proposed OLAF Technique	**98.05**

Table 5.2. Comparison of the Proposed (OLAF) Technique with some of the existing techniques on Yale Face Database

Method	Accuracy(%)
DCT +FLD based RBFNN *(Er et al, 2005)*	98.20
IROLS based RBFNN *(Wong et al, 2011)*	95.0
Polynomial RBFNN *(Oh et al, 2013)*	95.60
Proposed OLAF Technique	**99.50**

Table 5.3. Comparison of the Proposed (OLAF) Technique with some of the existing techniques on AR Face Database

Method	Accuracy(%)
KDCV based RBFNN *(Jing et al, 2008)*	85.36
IROLS based RBFNN *(Wong et al, 2011)*	75.5
Proposed OLAF Technique	**89.25**

5.6. Performance Evaluation of the Integrated Classifier

The techniques proposed for RBFNN design in this thesis therefore are integrated as follows.

(a) Selection of Centers and total number of neurons using FRBFNN proposed in Chapter 4.

(b) Estimation of Shape of Basis Function of each hidden neuron using OLAF proposed in Chapter 5.

The proposed design of Integrated Classifier using the RBFNN evolves the centers of the hidden neurons (RBF Units) and their total number using FRBFNN and obtains basis function shapes of the hidden layer neurons using OLAF. The integrated classifier's performance is assessed using parameters such as sensitivity (Recall), specificity, fall out, precision and accuracy etc. Confusion matrix and Receiver Operating Characteristic (ROC) curve are the two other significant methods of Classifier performance evaluation.

5.6.1. Sensitivity and Specificity

Sensitivity and *specificity* are the two statistical measures of the performance of a classification test. Sensitivity, also called as *Recall*, measures the *True Positive Rate* (TPR) of the classification while Specificity is the measure of *True Negative Rate* (TNR). Consider the total number of test images from the authorized database as *P* and the number of test images from the impostor database as *N*. The terms TP and TN refer to the number of true positives (TP) and true negatives (TN) respectively. If a test image of an authorized person is recognized correctly, it counts to the number of **True Positives (TP)**, while if it is rejected and labeled as impostor due to classifier's limited ability to recognize correctly, the count adds to the number of **False Negatives (FN)**. Similarly, if a test image of a person from the impostor database is recognized correctly as impostor, then it counts **True Negatives (TN),** while if an impostor is recognized as an authorized person, it counts as **False Positives (FP)** . These terms are summarized as identification action in Table 5.4.

Table 5.4: Summary of Terms used in Sensitivity Analysis

Term	Meaning
True Positive (TP)	Correct Identification
True Negative (TN)	Correct Rejection
False Negative (FN)	Incorrect Rejection
False Positive (FP)	Incorrect Identification

A classifier is said to be efficient when not only does it recognize the authorized test image correctly, but also rejects the impostors (unauthorized persons) in the test samples. A perfect classifier is said to be 100% sensitive, if all test samples belonging to authorized database are recognized by the classifier as authorized and is said to be 100% specific, if all impostors are not identified as authorized or are identified as unauthorized.

Various measures of classifier performance are given below.

Sensitivity is defined as the ratio of true positives and the total number of positive samples used in training.

$$Sensitivity\ (Recall) = \frac{TP}{P} = \frac{TP}{(TP + FN)} \tag{5.15}$$

Precision is computed as Positive Predictive Value (PPV) and is given as

$$Precision\ = \frac{TP}{(TP + FP)} \tag{5.16}$$

Specificity is defined as follows

$$Specificity = \frac{TN}{N} = \frac{TN}{(TN + FP)} \tag{5.17}$$

Fall out is a measure of False Positive Rate (FPR) and is defined as 1-Specificity given below

$$Fall\ Out = 1 - \ Specificity = \frac{FP}{(TN + FP)} \tag{5.18}$$

Accuracy is a measure of overall number of correctly recognized face image test samples.

$$Accuracy = \frac{(TP + TN)}{(TP + FN + TN + FP)} \tag{5.19}$$

Percentage accuracy is obtained by multiplying the accuracy computed using equation (5.19) by 100.

5.6.2. Confusion Matrix

A Classifier performance is evaluated more precisely by an error matrix called as *Confusion Matrix*. The columns of the matrix depict the instances of the actual classes and the rows of the matrix represent the instances in the predicted classes. The confusion matrix is also known as *Contingency Table* or *Error Matrix*. The performance of the classifier for two class classification problem is visualized as a 2×2 matrix as shown in Table 5.5. The

Table 5.5 : Two Class Classification based Confusion Matrix

		Actual Classes		
		Authorized (Positive)	Impostor (Negative)	
Predicted Classes	Authorized (Positive)	TP	FP	Positive Predictive Value = TP/(TP+FP)
	Impostor (Negative)	FN	TN	Negative Predictive Value = TN/(TN+FN)
		Sensitivity = TP/(TP+FN)	Specificity = TN/(TN+FP)	

A classifier is said to be perfect if it produces non-zero values in the diagonal of the Confusion matrix and has all zeros at the upper and lower triangular matrix entries. This means that the classifier is *not confused* and knows who is who correctly. A perfect classifier must not identify incorrectly an impostor, which means a perfect classifier must have as FP = 0. Similarly a perfect classifier must not incorrectly classify an authorized person as imposter, i.e. FN =0. The sensitivity for a perfect classifier is calculated as follows,

$$Sensitivity\ (Recall) = \frac{TP}{(TP + FN)} = \frac{TP}{(TP + 0)} = 1 \tag{5.20}$$

and

$$Fall\ Out = 1 - Specificity = \frac{FP}{(TN + FP)} = 0 \tag{5.21}$$

Therefore the perfect classifier must produce sensitivity as 1 and Fallout as 0. If it is a many class classification, with multiple persons (say *p* persons) in the face database, then the confusion matrix is of size p × p as shown in Table 5.6 below.

Table 5.6 : Many Class Classification based Confusion Matrix

		Actual Classes									
		C1	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**	**C9**	**C10**
Predicted Classes	**C1**	3	0	0	0	0	0	0	0	0	0
	C2	0	4	0	0	0	0	0	0	0	0
	C3	0	0	5	0	0	0	3	0	0	0
	C4	0	0	0	3	0	0	0	0	0	0
	C5	0	0	0	0	6	0	0	0	0	0
	C6	0	0	0	0	0	3	0	0	0	0
	C7	0	0	0	0	0	0	5	0	0	0
	C8	0	0	0	0	0	0	0	5	0	0
	C9	0	0	0	0	0	0	0	0	6	0
	C10	0	0	0	0	0	0	0	0	0	6

The above table depicts three false positives which belong to Class 7 in actual but is recognized as person represented by Class 3, and is considered as classifier's poor performance. To the extent, the Confusion matrix is sparsely filled with such entries, and the diagonal entries are non-zeros, the classifier performance can be considered satisfactory. The sum of the entries in the diagonal of Confusion Matrix represents the number of True Positives(TP), while the sum of the non-diagonal entries represents False Negatives (FN).

5.6.3. Receiver Operating Characteristic (ROC) Curve

ROC curve is a graph between False Positive Rates (FPR) and True Positive Rates (TPR) which are plotted on x-axis and y-axis respectively. The FPR is also known as Fall Out and is equal to (1-Specificity) while TPR is measured as Sensitivity. The ROC curve is the plot of Sensitivity as a function of Fall out. A classifier is said to be perfect if it correctly recognizes all positive test samples (TPR = 1) and rejects all negative samples correctly (FPR = 0). The curve with an observation resulting in extreme upper left corner point on the ROC (Sensitivity or TPR = 1 and Fallout or FPR = 0) is said to display the performance of the classifier as perfect (Fig. 5.11).

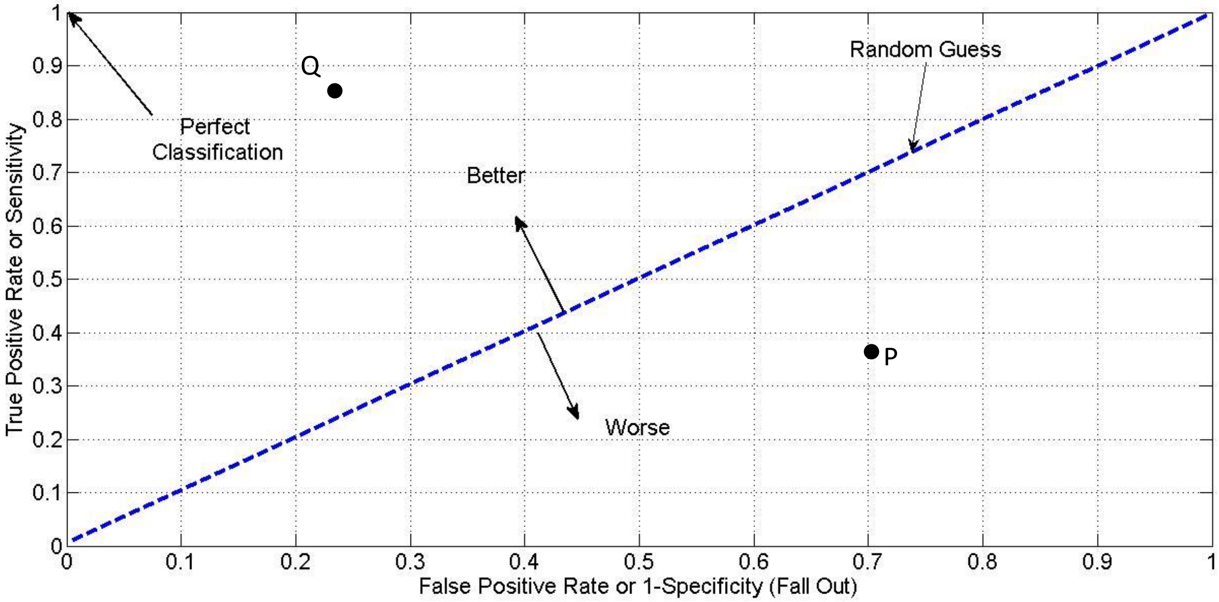


Fig.5.11 ROC Space

Figure 5.11 depicts a hypothetical graph with a dotted line starting from the lower left corner of the graph stretching upto the upper right corner. A classifier with fall out and sensitivity taken as point P(x,y) is said to be performing worse as P falls below the dotted line. If the classifier performs in terms of fall out and sensitivity taken as point Q(x,y), then it is said to be performing better as Q falls above the dotted line. If observations about the classifier performance are taken for a varying parameter, the ROC graph must stretch from lower left corner to the upper left corner of the graph and then to upper right corner. The area under ROC curve for a perfect classifier ideally is equal to 1, while it is 0.5 or less for an imperfect classifier.

The integrated classifier design (FRBFNN + OLAF) is validated using four benchmarked face databases namely ORL, Yale, AR and LFW. The overall performance of the proposed classifier is evaluated using performance evaluation techniques such as sensitivity, precision, specificity, fall out, accuracy, confusion matrix and ROC curve. The performance of the proposed integrated classifier is compared with Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) based classifiers. All the three classifiers used for comparison use RBFNN and use respective evolutionary methods to evolve the neurons. The basic versions of the algorithms of GA and PSO were used in corresponding classifiers for RBFNN center selection. The features used in all of the following experiments are DCT coefficients taken from the upper left corner as shown in Fig.5.12.

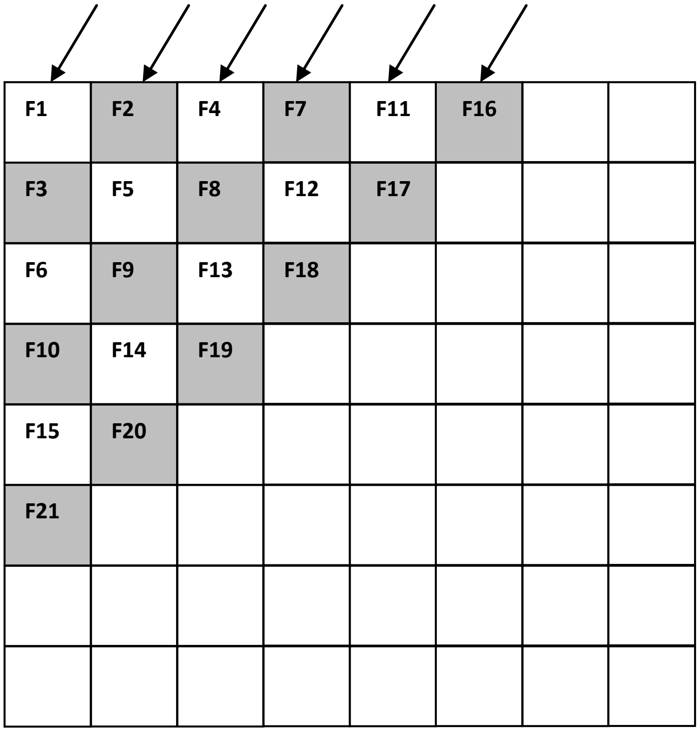


Fig. 5.12 Order of DCT coefficients starting from F_1 to F_n for selecting n features.

The parameters used for the evolutionary classifiers using GA, PSO and the proposed integrated classifier are listed in Table 5.7

Table 5.7. Parameters for Evolutionary Algorithms

PSO based Classifier for Face Recognition	**Genetic Algorithm based Classifier for Face Recognition**	**Proposed Integrated Classifier for Face Recognition**
C1 = 2.0	$p_c = 0.7$	$\beta_0 = 1.0$
C2 = 2.0	$p_m = 0.001$	$\gamma = 5$ (ORL), 3 (Yale)
W = 10.0	number of crossover Points = 2	γ =1 (LFW), γ =0.00001(AR)
Number of particles = 20	Number of Chromosomes = 20	Number of Fireflies = 20
Iterations = 7 (140 Epoch)	Iterations = 7 (140 Epochs)	Iterations = 7 (140 rounds)

The relation between Iteration and epoch is explained in Section 3.6.6. The consistency of rounds of interactions among the population of chromosomes, swarms and fireflies has been maintained by using 140 epochs for GA and PSO while the proposed integrated classifier uses 7 iterations. The GA and PSO algorithms based methods therefore have 2800 evaluations (=140×20) and the proposed integrated classifier also has 2800 evaluations (=7×20×20) where each evaluation corresponds to computations associated to each chromosome, particle or firefly in the respective methods.

The spread for all the three classifiers as above has been computed using proposed OLAF which is non-evolutionary in nature but contributes to improvement of the performance of the GA and PSO based classifiers as well as the proposed integrated classifier. Therefore the benefits of the proposed OLAF are contributed to all the three techniques uniformly for all face databases namely ORL, Yale, AR and LFW. The overlapping parameter α is varied from 0.5 to 9.5 at step size of 0.5 and ROC curves are obtained. Sensitivity analysis and Confusion matrix for all are presented in Sections 5.6.4 to 5.6.7. The sensitivity analysis is performed by randomly selecting 80 percent of the total persons to be the authorized persons and the remaining 20 percent are considered as impostors.

5.6.4. Performance Evaluation on ORL Face Database

The total number of features (as suggested in section 4.5.1) used for ORL face database is 45 while 5 out of 10 face images selected randomly for each person are used in the experiment. The sensitivity analysis is performed using 32 authorized and remaining 8 impostor person classes and generating training data using authorized persons images as per the description in Table 3.3. The fall out and sensitivity of the methods based on PSO, GA and the proposed integrated classifier are recorded in Table 5.8. The Sensitivity values were plotted against the fallout values in sorted order of fall out values. If two or more fall out values were same, then the sensitivity values were also sorted respectively. Thus the sorted <Sensitivity, fallout> points were plotted and curves were drawn for the three methods to be compared appropriately (Fig. 5.13) .

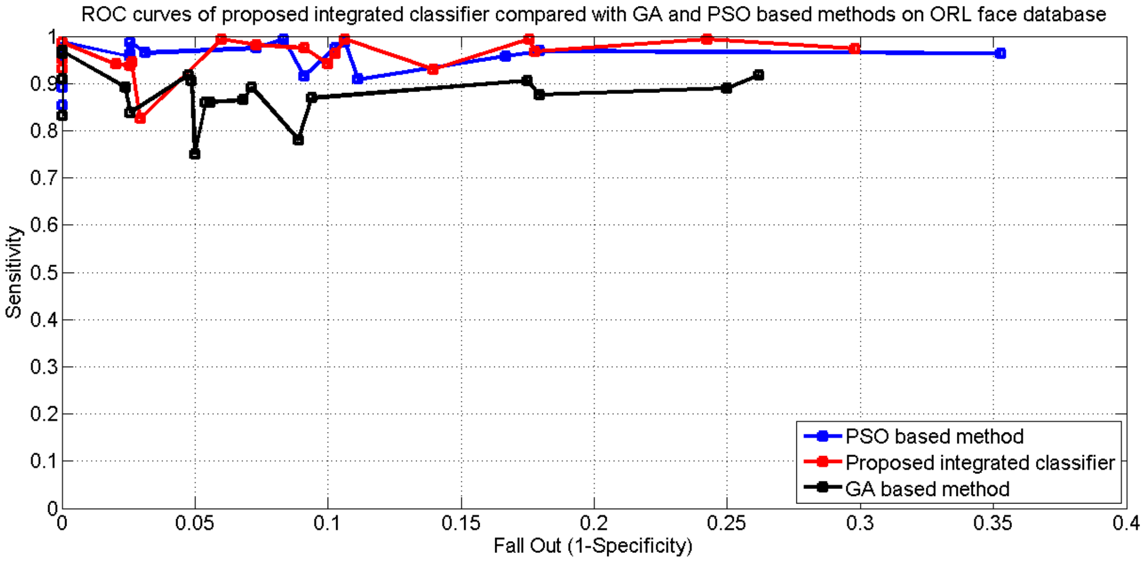


Fig.5.13. ROC curves of proposed integrated classifier compared with GA and PSO based methods on ORL face database

It is clearly visible that the proposed integrated classifier outperforms the other two methods as the curve (in red color) is closest to the boundary of the ROC space for most of the values of the overlapping factor. A classifier is said to be perfect if its *fall out* value is 0 and its *sensitivity* is 1. The best performance of PSO, GA based classifiers and that of proposed classifier are observed from Table 5.8 and are given in Table 5.9.

Table 5.8. Comparative Performance of the Integrated Classifier compared with GA and PSO based methods evaluated as Fallout and Sensitivity with respect to variations in Overlapping Factor on ORL Face Databases

α	PSO based Classifier		GA based Classifier		Proposed Integrated Classifier	
	Fall Out	**Sensitivity**	**Fall Out**	**Sensitivity**	**Fall Out**	**Sensitivity**
0.5	0	0.85443038	**0**	**0.968944099**	0.029411765	0.825301205
1.0	0.025641026	0.956521739	0.261904762	0.917721519	0	0.931677019
1.5	0	0.974193548	0.025641026	0.838509317	0	0.950920245
2.0	0	0.890909091	0.055555556	0.859756098	**0**	**0.98757764**
2.5	0.025641026	0.98757764	0.047619048	0.917721519	0.025641026	0.937888199
3.0	0.102564103	0.97515528	0.054054054	0.858895706	0.020408163	0.940397351
3.5	0	0.955414013	0	0.910179641	0.06	0.993333333
4.0	0.111111111	0.908536585	0	0.830065359	0.1	0.941176471
4.5	0.03125	0.964285714	0.023809524	0.892405063	0.106382979	0.993464052
5.0	0.083333333	0.993902439	0.068181818	0.865384615	0.026315789	0.944444444
5.5	0.166666667	0.957317073	0.048780488	0.905660377	0.090909091	0.974358974
6.0	0	0.974522293	0.179487179	0.875776398	0.073170732	0.981132075
6.5	0.090909091	0.916167665	0	0.832335329	0.139534884	0.929936306
7.0	0.106382979	0.986928105	0.09375	0.869047619	0.175438596	0.993006993
7.5	0.179487179	0.968944099	0.25	0.890243902	0.090909091	0.976047904
8.0	0.073170732	0.974842767	0.175	0.90625	0.102564103	0.962732919
8.5	0.025641026	0.962732919	0.071428571	0.892405063	0.242424242	0.994011976
9.0	**0**	**0.987654321**	0.088888889	0.780645161	0.177777778	0.967741935
9.5	0.352941176	0.963855422	0.05	0.75	0.29787234	0.973856209

Table 5.9. Best Performances of PSO, GA based methods and that of proposed integrated classifier evaluated on ORL face database

Classifier	α	Fallout	Sensitivity
PSO based method	9.0	0	0.9876
GA based method	0.5	0	0.9689
Proposed integrated classifier	2.0	0	**0.9876**

It is observed that the proposed algorithm is at par with the PSO based method in terms of the best Fallout-sensitivity pair values, while GA based classifier displays less sensitivity, which means it is able to classify all authorized persons with correctness in less percentage as compared to that of PSO based method and that of the proposed classifier. Though all the three methods are able to reject the impostors accurately using the proposed overlapping factor (fallout = 0). However, the ROC curve corresponding to the proposed integrated classifier shows better performance as compared to the other two ROC's as shown in Fig.5.13.

The confusion matrix using the proposed integrated classifier using best value of the overlapping factor as 2.0 is shown in Fig. 5.14. The total number of incorrectly recognized persons is 6 (i.e. FN = 6) as shown in Fig.5.14, The number of false positives is 0 (FP = 0), i.e no impostor is recognized as authorized person. All 39 test images of impostors were rejected fully (i.e. TN=39) and the specificity therefore is 1.0. Of all 161 test images belonging to the authorized persons, 155 test images were correctly classified, (i.e TP = 155). This is demonstrated as the diagonal entries of the Confusion Matrix shown in Fig.5.14. The Sensitivity analysis on ORL face database is done as follows

$$Sensitivity\ (Recall) = \frac{TP}{P} = \frac{TP}{(TP+FN)} = \frac{155}{(155+6)} = 0.962733$$

$$Precision = \frac{TP}{(TP+FP)} = \frac{155}{(155+0)} = 1.0$$

$$Specificity = \frac{TN}{N} = \frac{TN}{(TN+FP)} = \frac{39}{(39+0)} = 1.0$$

8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	2	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4

Fig.5.14. Confusion Matrix of the proposed integrated classifier evaluated on ORL face database using α = 2.0

$$Fall\ Out = 1 - Specificity = \frac{FP}{(TN + FP)} = \frac{0}{(39 + 0)} = 0$$

$$accuracy\ (\%) = \frac{(TP + TN)}{(TP + FN + TN + FP)} \times 100 = \frac{(155 + 39)}{(155 + 6 + 39 + 0)} \times 100$$

$$i.e. accuracy\ (\%) = \frac{194}{200} \times 100 = 97\%$$

5.6.5. Performance Evaluation on Yale Face Database

The total number of features (as suggested in section 4.5.1) used for Yale face database is 30 while 6 out of 10 face images selected randomly for each person are used in the experiment for training. The sensitivity analysis performed using 12 authorized and remaining 3 impostor person classes is shown in Table 5.10 and the corresponding ROC curves are shown in Fig.5.15.

Table 5.10. Comparative Performance of the Integrated Classifier compared with GA and PSO based methods evaluated as Fallout and Sensitivity with respect to variations in Overlapping Factor on Yale Face Databases

α	PSO based Classifier		GA based Classifier		Proposed Integrated Classifier	
	Fall Out	**Sensitivity**	**Fall Out**	**Sensitivity**	**Fall Out**	**Sensitivity**
0.5	**0**	**0.907407407**	0	0.851851852	0	0.653061224
1.0	0	0.547169811	0.142857143	1	0	0.86
1.5	0	0.693877551	0	0.666666667	0	0.88
2.0	0	0.586956522	**0**	**0.980769231**	**0**	**1**
2.5	0	0.833333333	0.285714286	0.673913043	0.1	0.8
3.0	0.166666667	0.895833333	0	0.422222222	0	0.941176471
3.5	0	0.632653061	0	0.510204082	0	0.814814815
4.0	0.166666667	0.979166667	0	0.38	0.117647059	1
4.5	0.181818182	0.755102041	0	0.32	0	0.961538462
5.0	0	0.814814815	0.111111111	0.431372549	0.3	0.76
5.5	0.166666667	0.952380952	0	0.727272727	0	0.980769231
6.0	0.5625	0.954545455	0	0.576923077	0	0.769230769
6.5	0.285714286	0.830188679	0	0.846153846	0.363636364	0.93877551
7.0	0.555555556	0.882352941	0	0.907407407	0.5	1
7.5	0.9	0.76	0	0.471698113	0.75	1
8.0	0.9	0.96	0	0.638297872	0.285714286	0.891304348
8.5	0.5	0.98	0	0.836734694	0	0.979591837
9.0	0.5	0.615384615	0	0.422222222	0.3	0.72
9.5	0.888888889	0.980392157	0.5	0.660714286	0.090909091	0.693877551

The performance of the proposed integrated classifier is better than the other two methods based on GA and PSO [Table 5.11]. The ROC curves shown in Fig.5.15 are condensed to Y-axis due to a large number of zeros in the Fall out columns of the Table 5.10 and therefore for remaining few pairs of <fallout, sensitivity> values, the curves seem slightly zigzagged. But it is clearly visible that a large number of points are plotted near the ROC space boundary for the proposed method.

Table 5.11. Best Performances of PSO, GA based methods and that of proposed integrated classifier evaluated on Yale face database

Classifier	α	Fallout	Sensitivity
PSO based method	0.5	0	0.9074
GA based method	2.0	0	0.9808
Proposed integrated classifier	2.0	**0**	**1.0**

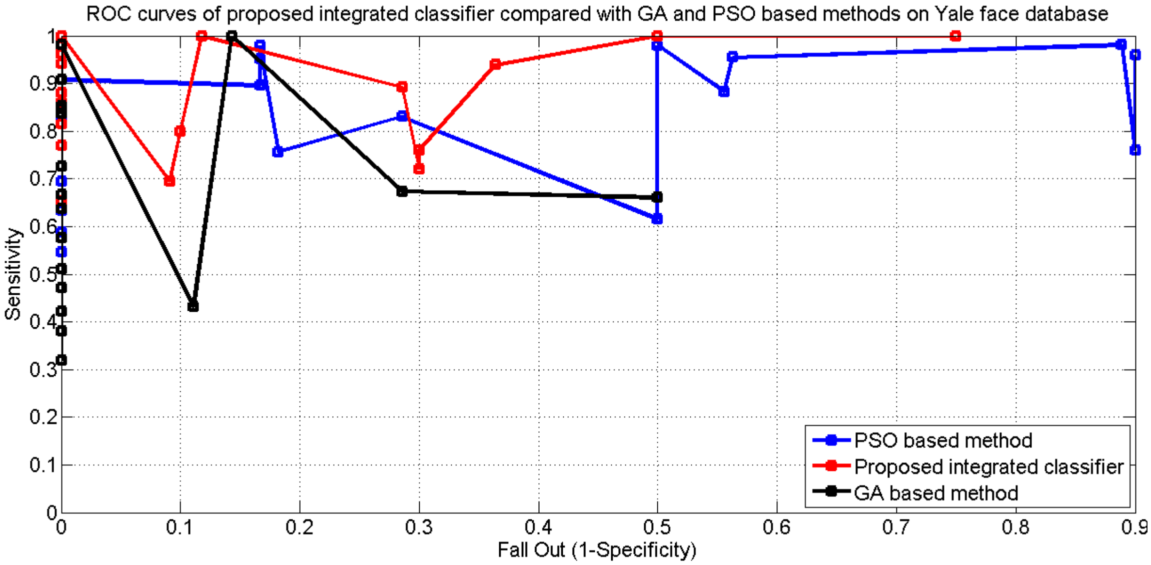
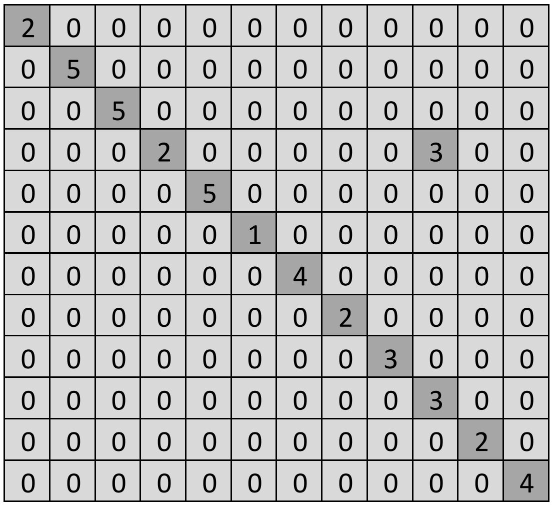


Fig.5.15. ROC curves of proposed integrated classifier compared with GA and PSO based methods on YALE face database

The ROC curves displayed in Fig.5.15 are explained by presenting the fall out and sensitivity pair values in the increasing order of fall out and sensitivity in Table 5.12. First we analyze the Sensitivity values corresponding to the Fall Out value equal to 0 for all the three methods. It is observed that PSO and GA based Classifiers start with the low sensitivity values of 0.55 (rounded value) and 0.32 respectively. While the proposed integrated classifier displays a trend of high sensitivity values from its start and reaches perfect sensitivity of 1.0 at $\alpha = 2.0$. The sensitivity at Fall Out equal to 0.5 is obtained as 1.0 using the proposed integrated classifier while PSO and GA based techniques perform low at sensitivity values equal to 0.98 and 0.66 respectively at their best. It is concluded that the proposed classifier, therefore, outperforms the PSO and GA based techniques on many parameters. The confusion matrix for the proposed integrated classifier based recognition using overlapping factor as 2.0 is shown in Fig. 5.16.

Table 5.12. The Fallout, Sensitivity pair values in increasing order on Yale Face Database

PSO based Classifier		GA based Classifier		Proposed Integrated Classifier	
Fall Out	**Sensitivity**	**Fall Out**	**Sensitivity**	**Fall Out**	**Sensitivity**
0	0.54717	0	0.32	0	0.653061
0	0.586957	0	0.38	0	0.769231
0	0.632653	0	0.422222	0	0.814815
0	0.693878	0	0.422222	0	0.86
0	0.814815	0	0.471698	0	0.88
0	0.833333	0	0.510204	0	0.941176
0	0.907407	0	0.576923	0	0.961538
0.166667	0.895833	0	0.638298	0	0.979592
0.166667	0.952381	0	0.666667	0	0.980769
0.166667	0.979167	0	0.727273	0	1
0.181818	0.755102	0	0.836735	0.090909	0.693878
0.285714	0.830189	0	0.846154	0.1	0.8
0.5	0.615385	0	0.851852	0.117647	1
0.5	0.98	0	0.907407	0.285714	0.891304
0.555556	0.882353	0	0.980769	0.3	0.72
0.5625	0.954545	0.111111	0.431373	0.3	0.76
0.888889	0.980392	0.142857	1	0.363636	0.938776
0.9	0.76	0.285714	0.673913	0.5	1
0.9	0.96	0.5	0.660714	0.75	1



2	0	0	0	0	0	0	0	0	0	0	0
0	5	0	0	0	0	0	0	0	0	0	0
0	0	5	0	0	0	0	0	0	0	0	0
0	0	0	2	0	0	0	0	0	3	0	0
0	0	0	0	5	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	4	0	0	0	0	0
0	0	0	0	0	0	0	2	0	0	0	0
0	0	0	0	0	0	0	0	3	0	0	0
0	0	0	0	0	0	0	0	0	3	0	0
0	0	0	0	0	0	0	0	0	0	2	0
0	0	0	0	0	0	0	0	0	0	0	4

Fig.5.16. Confusion Matrix of the proposed integrated classifier evaluated on Yale face database using α = 2.0

The total number of incorrectly recognized persons is 3 (i.e. FN = 3), computed as sum of non-diagonal entries of the Confusion Matrix shown in Fig.5.16. The number of false positives is 0 (FP = 0), i.e no impostor is recognized as authorized person. All 19 test images of impostors were rejected fully i.e. TN=19 and the specificity therefore is 1.0. Of all 41 test images belonging to the authorized persons, 38 test images were correctly classified, (i.e TP = 38, sum of diagonal entries of the Confusion Matrix). The Sensitivity analysis on Yale face database is done as follows

$$Sensitivity\ (Recall) = \frac{TP}{P} = \frac{TP}{(TP + FN)} = \frac{38}{(38 + 3)} = 0.926829$$

$$Precision = \frac{TP}{(TP + FP)} = \frac{38}{(38 + 0)} = 1.0$$

$$Specificity = \frac{TN}{N} = \frac{TN}{(TN + FP)} = \frac{19}{(19 + 0)} = 1.0$$

$$Fall\ Out = 1 - Specificity = \frac{FP}{(TN + FP)} = \frac{0}{(19 + 0)} = 0$$

$$accuracy = \frac{(TP + TN)}{(TP + FN + TN + FP)} = \frac{(38 + 19)}{(38 + 3 + 19 + 0)} = \frac{57}{60} = 0.95$$

5.6.6. Performance Evaluation on AR Face Database

The total number of features (as suggested in section 4.5.1) used for AR face database is 260 while 13 out of 26 face images selected randomly for each person are used in the experiment for training while 6 images from the remaining images are used for testing. The sensitivity analysis performed using 32 authorized and 8 impostor person classes is shown in Table 5.13 and the corresponding ROC curves are shown in Fig.5.17. The proposed integrated classifier outperforms GA based method and is comparable to the PSO based method as most of its points of <Sensitivity, Fall out> pair fall closer to the boundary of the ROC space as compared to the other two techniques. Table 5.14 displays the best performances of the three techniques.

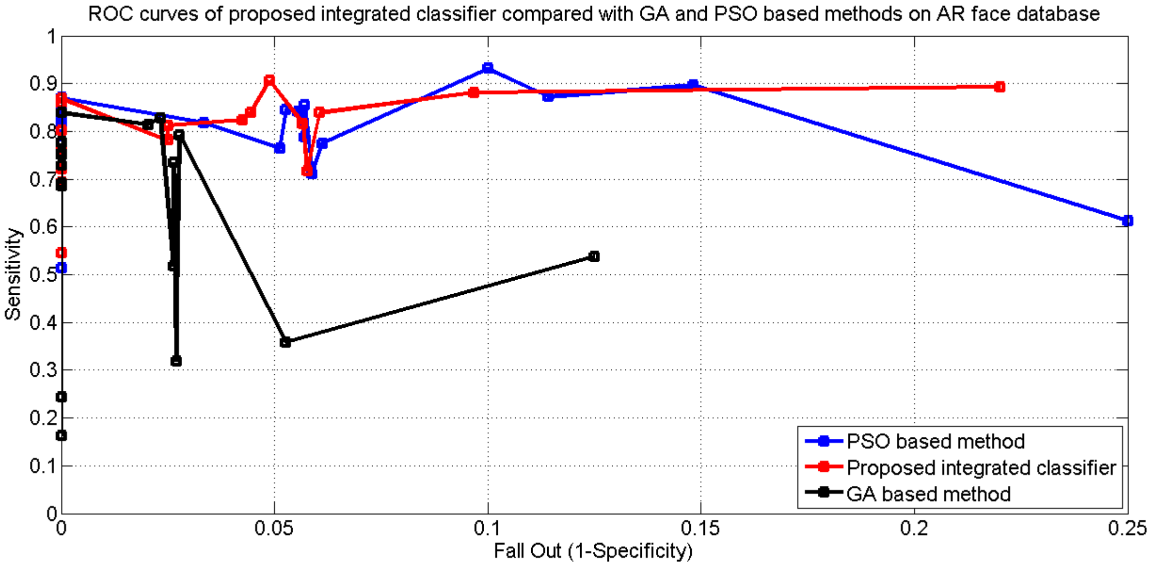


Fig.5.17. ROC curves of proposed integrated classifier compared with GA and PSO based methods on AR face database

Table 5.13. Comparative Performance of the Integrated Classifier compared with GA and PSO based methods evaluated as Fallout and Sensitivity with respect to variations in Overlapping Factor on AR Face Databases

α	PSO based Classifier		GA based Classifier		Proposed Integrated Classifier	
	Fall Out	**Sensitivity**	**Fall Out**	**Sensitivity**	**Fall Out**	**Sensitivity**
0.5	0	0.512987	0.027778	0.792683	0	0.545455
1.0	0.25	0.613095	0	0.685535	0	0.761905
1.5	0.058824	0.710843	0.125	0.5375	0	0.694268
2.0	0.033333	0.817647	0.052632	0.358025	0.025	0.8125
2.5	0.057143	0.787879	0	0.163265	0.057692	0.716216
3.0	0.051282	0.763975	0	0.24359	0	0.720497
3.5	0.061224	0.774834	0.027027	0.319018	0	0.755814
4.0	0	0.832335	0.026316	0.518519	0.060606	0.838323
4.5	0	0.774648	0	0.729032	0	0.726744
5.0	0	0.807947	0	0.69186	0.044444	0.83871
5.5	0	0.816993	0	0.778481	0.042553	0.823529
6.0	0.148148	0.895954	0	0.754601	0.056604	0.816327
6.5	0	0.857143	0.020408	0.81457	0	0.801282
7.0	0.055556	0.841463	0	0.75	0	0.858896
7.5	**0**	**0.87037**	0.026316	0.734568	**0**	**0.86875**
8.0	0.052632	0.845679	0	0.775641	0.025	0.78125
8.5	0.1	0.93125	0.023256	0.828025	0.096774	0.881657
9.0	0.114286	0.872727	0	0.75	0.22	0.893333
9.5	0.057143	0.854545	**0**	**0.839286**	0.04878	0.90566

3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0
0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	4	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	2	0	0	0	0	0	0	0	2	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	2	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	8	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0
0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	3	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7

Fig.5.18. Confusion Matrix of the proposed integrated classifier evaluated on AR face database using $\alpha = 7.5$

Table 5.14. Best Performances of PSO, GA based methods and that of proposed integrated classifier evaluated on AR face database

Classifier	α	Fallout	Sensitivity
PSO based method	7.5	0	0.8704
GA based method	9.5	0	0.8393
Proposed integrated classifier	7.5	0	0.8688

The confusion matrix is shown in Fig. 5.18. The total number of incorrectly recognized persons is 29 (i.e. FN = 29, identified as sum of non-diagonal entries of the Confusion Matrix). The number of false positives is 3 (FP = 3), i.e three impostor persons are recognized as authorized persons. A total of 35 out of 38 test images of impostors were rejected fully i.e. TN=35 and the specificity therefore is 0.92. Of all 162 test images belonging to the authorized persons, 133 test images were correctly classified, (i.e TP = 133, sum of the diagonal entries of the Confusion Matrix). The Sensitivity analysis on AR face database is done as follows:

$$Sensitivity\ (Recall) = \frac{TP}{P} = \frac{TP}{(TP+FN)} = \frac{133}{(133+29)} = 0.820988$$

$$Precision = \frac{TP}{(TP+FP)} = \frac{133}{(133+3)} = 0.9779$$

$$Specificity = \frac{TN}{N} = \frac{TN}{(TN+FP)} = \frac{35}{(35+3)} = 0.921053$$

$$Fall\ Out = 1 - Specificity = \frac{FP}{(TN+FP)} = \frac{3}{(38+3)} = 0.078947$$

$$accuracy = \frac{(TP+TN)}{(TP+FN+TN+FP)} = \frac{(133+35)}{(133+29+35+3)} = \frac{168}{200} = 0.84$$

5.6.7. Performance Evaluation on LFW Face Database

The total number of features (as suggested in section 4.5.1) used for LFW face database is 65 while 8 out of 16 face images selected randomly for each person are used in the experiment for training and 8 images are used for testing. The sensitivity analysis performed using 6 authorized and remaining 2 impostor person classes is as shown in Table 5.15 and the corresponding ROC curves are shown in Fig.5.19. Again it is evident that the proposed integrated classifier outperforms GA and PSO based methods as most of its points of <Sensitivity, Fall out> pair fall closer to the boundary of the ROC space as compared to the other two techniques. Table 5.16 displays the best performances of the three techniques.

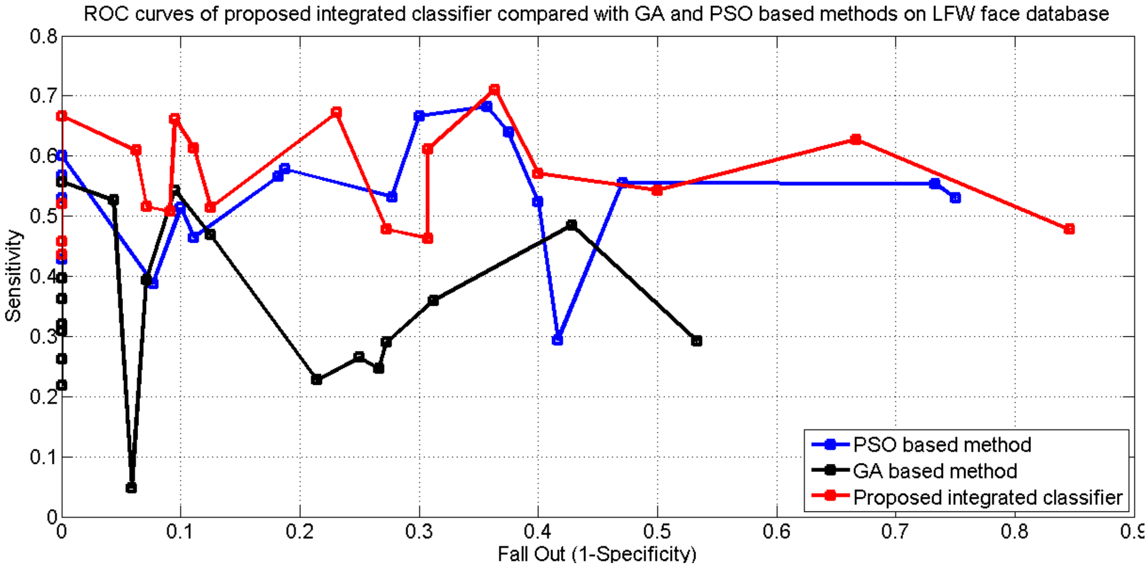


Fig.5.19. ROC curves of proposed integrated classifier compared with GA and PSO based methods on LFW face database

10	0	0	0	0	0	0	0
5	4	0	0	1	0	0	0
0	0	3	0	0	0	0	0
0	0	0	5	0	0	0	0
0	0	1	0	2	0	0	0
0	0	0	1	0	3	0	0
0	3	0	1	1	0	7	0
0	2	1	0	2	0	0	4

Fig.5.20. Confusion Matrix of the proposed integrated classifier evaluated on LFW face database using $\alpha = 7.0$

Table 5.15. Comparative Performance of the Integrated Classifier compared with GA and PSO based methods evaluated as Fallout and Sensitivity with respect to variations in Overlapping Factor on LFW Face Databases

α	PSO based Classifier		GA based Classifier		Proposed Integrated Classifier	
	Fall Out	**Sensitivity**	**Fall Out**	**Sensitivity**	**Fall Out**	**Sensitivity**
0.5	0	0.530303	0.095238	0.542373	0.071429	0.515152
1.0	0	0.396552	0.428571	0.484848	0	0.457627
1.5	**0**	**0.6**	0	0.397059	0	0.520548
2.0	0	0.567164	0.3125	0.359375	0.111111	0.612903
2.5	0.1	0.514286	0.272727	0.289855	0.125	0.513889
3.0	0.111111	0.464789	0.214286	0.227273	0.230769	0.671642
3.5	0	0.428571	0	0.308824	0.095238	0.661017
4.0	0.181818	0.565217	0.058824	0.047619	0	0.43662
4.5	0.1875	0.578125	0	0.21875	0.307692	0.462687
5.0	0.076923	0.38806	0.25	0.264706	0.5	0.542857
5.5	0.277778	0.532258	0.125	0.46875	0.4	0.571429
6.0	0.416667	0.294118	0.533333	0.292308	0.666667	0.627119
6.5	0.357143	0.681818	0	0.262295	0.846154	0.477612
7.0	0.470588	0.555556	0	0.362069	**0**	**0.666667**
7.5	0.75	0.529412	0	0.319444	0.363636	0.710145
8.0	0.3	0.666667	0.071429	0.393939	0.307692	0.61194
8.5	0.375	0.638889	0.266667	0.246154	0.0625	0.609375
9.0	0.733333	0.553846	**0**	**0.557143**	0.090909	0.507246
9.5	0.4	0.523077	0.043478	0.526316	0.272727	0.478261

Table 5.16. Best Performances of PSO, GA based methods and that of proposed integrated classifier evaluated on LFW face database

Classifier	α	Fallout	Sensitivity
PSO based method	1.5	0	0.6000
GA based method	9.0	0	0.5571
Proposed integrated classifier	7.0	0	0.6667

The confusion matrix is shown in Fig.5.20. The total number of incorrectly recognized persons is 18 (i.e. FN = 18, sum of non-diagonal entries of the Confusion Matrix). The number of false positives is 2 (FP = 2), i.e two impostor persons are recognized as authorized persons. A total of 22 out of 24 test images of impostors were rejected fully i.e. TN = 22 and the specificity therefore is 0.916667. Of all 56 test images belonging to the authorized persons, only 38 test images were correctly classified, (i.e TP = 38, sum of diagonal entries of the confusion matrix) .The Sensitivity analysis on LFW face database is done as follows:

$$Sensitivity\ (Recall) = \frac{TP}{P} = \frac{TP}{(TP+FN)} = \frac{38}{(38+18)} = 0.678571$$

$$Precision = \frac{TP}{(TP+FP)} = \frac{38}{(38+2)} = 0.95$$

$$Specificity = \frac{TN}{N} = \frac{TN}{(TN+FP)} = \frac{22}{(22+2)} = 0.916667.$$

$$Fall\ Out = 1 - Specificity = \frac{FP}{(TN+FP)} = \frac{2}{(22+2)} = 0.083333$$

$$accuracy = \frac{(TP+TN)}{(TP+FN+TN+FP)} = \frac{(38+22)}{(38+18+22+2)} = \frac{60}{80} = 0.75$$

The accuracy of LFW face database is low as expected because of large variations in expression, pose and illumination, and the persons being celebrities are also wearing artificial makeup.

5.7. Conclusion

In this chapter, we proposed novel shape estimation technique, named as OLAF for improved face recognition. The proposed OLAF technique is adaptive as it computes spread of a basis function based on its radius and distance between its center and that of the nearest sub-cluster of a different class. The proposed technique performed well on the benchmark face databases namely ORL, Yale, AR and LFW yielding average recognition accuracy of 98.05% (± 1.17) with overlapping factor α = 5.8 , 99.50% (± 0.81) with α = 1.8, 89.25% (± 2.15) with α = 9.8 and 52.0% (± 9.0) with α = 4.6 respectively. The average recognition accuracy was computed over 10 independent runs and the low standard deviations of 1.17, 0.81, 2.15 and 9.0 were obtained for ORL, Yale AR and LFW face databases respectively demonstrating the stability of the proposed algorithm. It is established that the proposed OLAF technique outperforms the method based on fixed shape (spreads). The proposed shape estimation technique also outperforms some of the existing techniques on RBFNN design for face recognition.

An integrated classifier for improved face recognition is developed which uses the proposed FRBFNN technique for center selection and number of RBF units and the proposed OLAF technique for basis function shape at each RBF unit. The proposed integrated classifier performance is evaluated using various performance measures such as sensitivity analysis, confusion matrix and ROC curves. The performance of the proposed integrated classifier is compared with the methods based on Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). It is observed that the proposed technique outperforms GA and PSO for Yale, AR, and LFW, while for ORL proposed method outperforms GA and has almost same as performance as PSO. The proposed classifier obtains sensitivity measure for ORL face database as 0.9878 which is better than 0.9689 produced by GA. Similarly the sensitivity values of the proposed integrated classifier for Yale, AR and LFW face databases are 1.0, 0.8688 and 0.6667 respectively with fall out as obtained as 0 in all observations. GA based method produces sensitivity as 0.9808, 0.8393 and 0.5571 respectively for Yale, AR and LFW face databases, while the method produces fallout values as 0 for all the three face databases. PSO based method produces sensitivity of

0.9074, 0.8704 and 0.6 respectively for Yale, AR and LFW face databases and fallout values as 0 for all the three face databases.

Chapter 6

Conclusion and Future Directions

In this thesis, the face recognition problem has been addressed using computational intelligence (CI) techniques. Radial Basis Function Neural Networks (RBFNN) and Evolutionary Algorithm (EA) are used in the present research. Advantages of generalization abilities of RBFNN and fast convergence of EA are integrated for improved and robust face recognition. The focus of this research is on RBFNN design using Firefly Algorithm (FA) and the work includes Feature Selection (FS), Optimal number and centers of the RBF units, and Basis Function shape. The three major problems are addressed as described below

(A) Firefly Inspired Feature Selection

- A polynomial time algorithm FIFS (Firefly Inspired Feature Selection) for feature selection for face features is proposed. The algorithm is novel and uses Firefly Algorithm in feature selection for improved average recognition accuracy.
- A detailed analysis of the proposed FIFS technique is presented. *Effect of parameter* γ on algorithm convergence, average recognition accuracy and dimensionality reduction is investigated using four benchmark face databases namely ORL, Yale, AR and LFW. The algorithm is adaptive and converges fast in maximum seven iterations using proposed value of γ as 1×10^{-6} for ORL face database and 1×10^{-7} for the remaining three face databases (Yale, AR and LFW). The maximum number of fireflies is as less as 20 in obtaining the best performance.
- The performance of FIFS technique is compared with that of the feature selection methods based on Genetic Algorithm (GA) and Particle Swarm Algorithm (PSO). It is established that

the proposed FIFS algorithm *converges faster* than GA and PSO based feature selection methods. The proposed FIFS technique *outperforms* the GA and PSO based methods for feature dimensions in the range 60 to 100, and selects optimal features to contribute to the best performances in terms of average recognition accuracy. The accuracies obtained using GA based feature selection are 77.88%, 69.17%, 15.75% and 22.19% for the four face databases ORL, Yale, AR and LFW respectively while in PSO based method the average recognition accuracies are 91.63%, 87.92%, 53.88% and 55% respectively. The proposed FIFS algorithm outperforms the above two methods and average recognition accuracies obtained for ORL, Yale, AR and LFW face databases are 94.00%, 97.50%, 58.75% and 64.38% respectively. The proposed algorithm also outperforms some of the existing research work reported in literature.

(B) RBFNN Center Selection

- A novel framework to address the problem of center selection for the RBF units using Firefly Algorithm is proposed. This work includes representation of center selection as an optimization problem, the design of artificial firefly in the context of center selection and the firefly movement algorithm which obtains optimal number and centers of the Radial Basis Function (RBF) Units (using Gaussian basis functions). The proposed algorithm is named as FRBFNN (Firefly inspired RBFNN design).
- The *strength* of the proposed FRBFNN lies in its capability to *handle variations* such as illumination, pose, expression and accessories etc. The algorithm is *adaptive* as it captures the subjective similarity of training face images and learns the structure of the training face data on its own. This results in *evolution of the natural sub-clusters* for each person resulting in obtaining the optimal number of RBF units automatically. The centers of these sub-clusters are used as RBF centers. The proposed center selection algorithm *does not have the overhead of feature selection* and is therefore *computationally efficient.*
- A detailed analysis on *parameter selection* and firefly convergence has been performed to find the most suitable values of the parameters Gamma(γ) and the number of fireflies. The values of γ are 5, 3, 0.00001 and 1 while minimum *numbers of fireflies* are proposed to be 11, 13, 11 and 13 for the ORL, Yale, AR and LFW face databases respectively. A maximum of 20 fireflies can be taken for the most stable results for all face databases. The algorithm

converges in 6 to 9 iterations while a maximum of 20 iterations can be used to obtain the best results.

- The *effect* of various other parameters such as number of features, number of training images and number of iterations using the proposed FRBFNN algorithm has also been investigated. The best performances using ORL, Yale, AR and LFW face databases were achieved using only 45, 30, 260 and 65 features, resulting in *dimensionality reduction* of 99.5%, 99.7%, 94.8% and 99.4%. The strength of the proposed FRBFNN is evident with huge dimensionality reduction at *no extra cost of feature selection*. The features were taken deterministically from the upper left corner of the Discrete Cosine Transformed image and no overhead incurred on selecting the best features.
- The evolution of hidden layer neurons is also demonstrated through an experiment and it is established that more complex face databases such as AR and LFW capturing *large variations have more neurons* evolved and less complex face databases such as ORL and Yale have less neurons evolved for the corresponding RBFNN. This supports our claim that the algorithm is adaptive and handles variations.
- The proposed evolutionary FRBFNN technique outperforms various other classifiers such as k-Means based RBFNN and Euclidean (Nearest Neighbor) Classifier. It is observed that the proposed FRBFNN outperforms some of the existing algorithms in terms of *number of iterations* or epochs used such as Point Symmetry Distance based RBFNN *(Sing et al, 2007)* and Self Adapting RBFNN *(Sing et al, 2009)* which use 6500 and 1500 epochs respectively with ORL face database while the proposed FRBFNN algorithm used only a maximum of 20 iterations. Also, the proposed FRBFNN technique outperforms Fuzzy Hybrid Learning Algorithm (FHLA) *(Haddadnia et al, 2003)* technique in terms of *speed of convergence*. It is observed that the proposed algorithm is faster by a percentage ranging 73.47% to 94.15% as compared to FHLA technique over the range of features used.
- The proposed FRBFNN algorithm is a *real time solution* and the face recognition using a test face image is achieved in almost one hundredth of a second, while training, being a onetime task, is also efficiently done in few seconds. It is observed that it takes only 10.92 seconds to train the proposed FRBFNN based classifier using 200 training images of ORL face database and takes 0.0048 seconds to test a face image on an average taken over 10 independent runs. Similar observations for Yale, AR and LFW face databases are 9.23, 59.85 and 4.59 seconds

for training using 90, 520 and 80 training face images, testing times for each test face image are 0.0027, 0.0265 and 0.0022 seconds respectively.

- The *average recognition accuracies* (± standard deviation) obtained for the ORL, Yale, AR and LFW face databases are 97.75% (± 2.31), 99.83% (± 0.53), 93.15% (± 3.25) and 60.50% (± 9.65). It is observed that the proposed FRBFNN algorithm *outperforms* some of the existing algorithms.
- The proposed FRBFNN algorithm is fast in its convergence and produces better accuracies which leads to a conclusion that the proposed work can be used to train the face recognition system with large number of training faces to handle large environmental variations.

(C) Basis Function Shape Estimation

- Overlapping factor based shape estimation algorithm for Gaussian basis functions (OLAF) is proposed for shape estimation of Gaussian basis functions. This technique takes advantage of the sub-clusters formed using FRBFNN technique and is based on the relative distances between sub-clusters of different person classes and the distance of the furthest point from the sub-cluster mean in the same class. The proposed OLAF technique outperforms some of the existing techniques in face recognition.
- The effect of overlapping factor on average recognition accuracy is investigated using the proposed OLAF algorithm of shape estimation. The proposed OLAF technique outperforms the constant spread based RBFNN in face recognition.
- The proposed FRBFNN algorithm of center selection and OLAF algorithm of basis function shape estimation of RBF unit are combined to form an integrated classifier and contribute to the RBFNN hidden layer design for face recognition.
- Sensitivity Analysis is performed to measure the performance of the RBFNN based integrated classifier designed by combining FRBFNN and OLAF algorithms. The sensitivity values computed using GA based classifier are 0.9689, 0.9808, 0.8393 and 0.5571 for ORL, Yale, AR and LFW face databases respectively while the values using PSO based classifier are 0.9876, 0.9074, 0.8704 and 0.6 respectively for each database. The proposed Integrated Classifier (FRBFNN + OLAF) outperforms the GA and PSO based classifiers as sensitivity

values are obtained as 0.9876, 1.0, 0.8688 and 0.6667 respectively for ORL, Yale, AR and LFW face databases.

- Receiver Operating Characteristics (ROC) curves and confusion matrix methods were also used for analyzing the performance of the proposed integrated classifier. It is observed that the proposed firefly inspired method outperforms other evolutionary methods.

Directions for Future Work

- This work can be used in various applications such as speech recognition, gesture recognition, character recognition, signature recognition, emotion recognition to detect pain especially in patients who are unable to express it while in Intensive Care Unit etc. as the proposed work provides a *general framework* for any recognition task.
- This work supports *real time face recognition* and can be used in airport surveillance systems to identify a person from the given criminal face database. Since surveillance cameras cover a wide range, the face of the persons may not be clear, therefore there is a need to extend the present work using *incomplete information*.
- The present research work can be extended with use of *Multiquadric basis functions* in Face recognition as it is learnt through literature review that these possess potential for better performance. Use of various other basis functions also needs to be explored for improvement in face recognition accuracy.
- Work can be extended with multimodal biometric traits such as fingerprint and iris, which will enhance the recognition performance.
- LFW is the most challenging face database which captures huge variations in facial expressions, lighting conditions, pose and makeup or disguise. The present study has a limitation in handling such huge variations, therefore this work may be extended to look at several issues relating to the LFW face database.

List of Publications

1. Agarwal, V.; Bhanot, S., "Radial Basis Function Neural Network Based Face Recognition Using Firefly Algorithm", Neural Computing and Applications, Springer (*Communicated*)
2. Agarwal, V.; Bhanot, S., "Basis Function Shape Estimation Algorithm for Improved Face Recognition using Radial Basis Function Neural Network", International Journal of Computational Vision and Robotics, Inderscience (*Communicated*)
3. Agarwal, V.; Bhanot, S., "Firefly Inspired Feature Selection for Face Recognition", International Conference on Contemporary Computing (IC3-2015), Jaypee Institute of Information Technology (JIIT), Noida, India, 20-22 August 2015.(**SCOPUS**) [doi: 10.1109/IC3.2015.7346689]
4. Agarwal, V.; Bhanot, S., "Evolutionary design of Multiquadric radial basis functions neural network for face recognition," 2013 Fourth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), pp.1,5, 18-21 Dec. 2013 (**SCOPUS**) [DOI: 10.1109/NCVPRIPG.2013.6776196]
5. Agarwal, V.; Bhanot, S., "Firefly Inspired Center Initialization of Radial Basis Function Neural Network for Face Recognition", IEEE Workshop on Computational Intelligence: Theories, Applications and Future Directions, IIT Kanpur, India, pp. 9-15, July 2013.
6. Agarwal, V.; Bhanot, S., "A Dynamic Data Structure for Real Time Face Recognition," 2012 International Conference on Computer Technology and Science (ICCTS 2012), New Delhi, IPCSIT vol. 47 (2012) © (2012) IACSIT Press, Singapore DOI: 10.7763/IPCSIT.2012.V47.20

REFERENCES

Ahmad, A.; Amira, A.; Nicholl, P.; Krill, B. (2011), "Dynamic partial reconfiguration of 2-D Haar wavelet transform (HWT) for face recognition systems," Consumer Electronics (ISCE), 2011 IEEE 15th International Symposium on, pp.9-13

Ahmed, N.; Natarajan, T.; Rao, K.R. (1974), "Discrete Cosine Transform," in Computers, IEEE Transactions on, vol.C-23, no.1, pp.90-93

Ajit Krisshna, N.L. ; Deepak V. K.; Manikantan, K.; Ramachandran, S. (2014), "Face recognition using transform domain feature extraction and PSO-based feature selection," Applied Soft Computing, Volume 22, Pages 141-161, ISSN 1568-4946

Ajitha, S.; Annis Fathima, A.; Vaidehi, V.; Hemalatha, M.; Karthigaiveni, R. (2014), "Face recognition system using Combined Gabor Wavelet and DCT approach," Recent Trends in Information Technology (ICRTIT), 2014 International Conference on, pp.1-6

Alexandridis, A.; Chondrodima, E.; Sarimveis, H. (2013), "Radial Basis Function Network Training Using a Nonsymmetric Partition of the Input Space and Particle Swarm Optimization," Neural Networks and Learning Systems, IEEE Transactions on, vol.24, no.2, pp.219-230.

Allinson, N.M.; Ellis, A.W. (1992), "Face recognition: combining cognitive psychology and image engineering," *Electronics & Communication Engineering Journal*, vol.4, no.5, pp.291-300

Amira, A.; Farrell, P.(2005), "An automatic face recognition system based on wavelet transforms," *Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on*, vol. 6, pp.6252-6255

AT&T Laboratories Cambridge, ORL Face Database, *AT&T Laboratories Cambridge*, U.K. (Downloaded in 2010, www.cl.cam.ac.uk/Research/DTG/attarchive/pub/data/att_faces.zip)

Atta, R.; Ghanbari, M. (2012), "An efficient face recognition system based on embedded DCT pyramid," Consumer Electronics, IEEE Transactions on, vol.58, no.4, pp.1285-1293

Ayyavoo, T.; Jayasudha, J.S. (2013), "Face recognition using enhanced energy of Discrete Wavelet Transform," in Control Communication and Computing (ICCC), 2013 International Conference on, pp.415-419

Balasubramanian, M. ; Palanivel, S. ; Ramalingam, V. (2009), "Real time face and mouth recognition using radial basis function neural networks," Expert Systems with Applications, Volume 36, Issue 3, Part 2, Pages 6879-6888.

Bayona, V.; Moscoso, M.; Kindelan, M. (2011), "Optimal constant shape parameter for multiquadric based RBF-FD method," Journal of Computational Physics, Volume 230, Issue 19, 10, pp 7384-7399

Belhumeur, P.N.; Hespanha, J.P.; Kriegman, D. (1997), "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol.19, no.7, pp.711-720

Bhatt, H.S.; Bharadwaj, S.; Singh, R.; Vatsa, M. (2013), "Recognizing Surgically Altered Face Images Using Multiobjective Evolutionary Algorithm," Information Forensics and Security, IEEE Transactions on, vol.8, no.1, pp.89-100

Bishop, C. (1995), Neural Networks for Pattern Recognition, Oxford University Press, Chapter 5, pp 164-193.

Bouattour, H.; Fogelman Soulie, F.; Viennet, E. (1992), "Neural nets for human face recognition," *Neural Networks, 1992. IJCNN., International Joint Conference on*, vol.3, no., pp.700-704

Broomhead, D.S.; Lowe, D. (1988), "Multivariate functional interpolation and adaptive networks". *Complex Systems,* 2:321-355

Brunelli, R.; Poggio, T. (1993), "Face recognition: features versus templates," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol.15, no.10, pp.1042-1052

Cao, X. ;Shen, W. ;Yu, L.G.;Wang, Y.L.; Yang, J.Y. ; Zhang, Z.W. (2012), "Illumination invariant extraction for face recognition using neighboring wavelet coefficients", Pattern Recognition, Volume 45, Issue 4, Pages 1299-1305

Chakraborti, T.; Chatterjee, A. (2014), "A novel binary adaptive weight GSA based feature selection for face recognition using local gradient patterns, modified census transform, and local binary patterns", Engineering Applications of Artificial Intelligence, Volume 33, Pages 80-90

Chang, C.; Hsu, H. (2008), "Apply an Adaptive Center Selection Algorithm to Radial Basis Function Neural Network for Face Recognition," in Innovative Computing Information and Control, ICICIC '08. 3rd International Conference on, pp.171-171

Chelali, F.Z.; Djeradi, A. (2014), "Face recognition system using neural network with Gabor and discrete wavelet transform parameterization," in Soft Computing and Pattern Recognition (SoCPaR), 2014 6th International Conference of, pp.17-24

Chellappa, R.; Sinha, P.; Phillips, P.J. (2010), "Face Recognition by Computers and Humans," *Computer*, vol.43, no.2, pp.46-55

Chellappa, R.; Wilson, C.L.; Sirohey, S. (1995), "Human and machine recognition of faces: a survey," *Proceedings of the IEEE*, vol.83, no.5, pp.705-741

Chen, C.; Zhang, J. (2007), "Wavelet Energy Entropy as a New Feature Extractor for Face Recognition," in Image and Graphics, 2007. ICIG 2007. Fourth International Conference on, pp.616-619

Chen, W.; Meng, J.E.; Wu, S. (2006), "Illumination compensation and normalization for robust face recognition using discrete cosine transform in logarithm domain," Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, vol.36, no.2, pp.458-466

Cheng, A.H.-D. (2012), "Multiquadric and its shape parameter-A numerical investigation of error estimate, condition number, and round-off error by arbitrary precision computation", Engineering Analysis with Boundary Elements, 36, pp 220-239

Cheng, G.; Shi, C.; Zhu, K; Gong, K (2011), "The Application of Binary Particle Swarm Algorithm in Face Recognition," Computational Intelligence and Security (CIS), 2011 Seventh International Conference on, pp.1229-1233

Choi, W.; Tse, S.; Wong, K.; Lam, K. (2008), "Simplified Gabor wavelets for human face recognition", Pattern Recognition, Volume 41, Issue 3, Pages 1186-1199

Cover, T.M. (1965), "Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition," in Electronic Computers, IEEE Transactions on, vol.EC-14, no.3, pp.326-334

Dabbaghchian, S.; Ghaemmaghami, M. P. ; Aghagolzadeh, A. (2010), "Feature extraction using discrete cosine transform and discrimination power analysis with a face recognition technology," Pattern Recognition, Volume 43, Issue 4, Pages 1431-1440

Darestani, M.R.Y.; Sheikhan, M.; Khademi, M. (2013), "Face recognition using contourlet-based features and hybrid PSO-neural model," in Information and Knowledge Technology (IKT), 2013 5th Conference on , pp.181-186

Dawoud, N.N.; Samir, B.B. (2011), "Best wavelet function for face recognition using multi-level decomposition," Research and Innovation in Information Systems (ICRIIS), 2011 International Conference on, pp.1-6

De Marsico, M.; Nappi, M.; Riccio, D.; Wechsler, H. (2013), "Robust Face Recognition for Uncontrolled Pose and Illumination Changes," *Systems, Man, and Cybernetics: Systems, IEEE Transactions on*, vol.43, no.1, pp.149-163

De Silva, C.R.; Ranganath, S. ; De Silva, L.C. (2008), "Cloud basis function neural network for holistic facial expression recognition",Pattern Recognition, 41, pp 1241-1253

Dong, J.; Zhao, L.; Zhang, L. (2013), "Face recognition based on neural network ensemble and feature fusion," *Information Science and Technology (ICIST), 2013 International Conference on*, pp.59-62

Du, G.; Gong, L.; Su, F. (2009), "An effective Gabor-feature selection method for face recognition," Network Infrastructure and Digital Content, 2009. IC-NIDC 2009. IEEE International Conference on, pp.722-725

Ekenel, H.K.; Stiefelhagen, R. (2006), "Analysis of Local Appearance-Based Face Recognition: Effects of Feature Selection and Feature Normalization," Computer Vision and Pattern Recognition Workshop, 2006. CVPRW '06. Conference on, pp.34,34

Elazhari, A.; Ahmadi, M. (2014), "A neural network based human face recognition of low resolution images," *World Automation Congress (WAC), 2014*, pp.185-190

El-Bakry, H.M.; Abo-Elsoud, M.A.; Kamel, M.S. (2000), "Integrating Fourier descriptors and PCA with neural networks for face recognition," *Radio Science Conference, 2000. 17th NRSC '2000. Seventeenth National*, pp.C22/1-C22/8

Er, M. J.; Chen, W.; Wu, S. (2005), "High-speed face recognition based on discrete cosine transform and RBF neural networks," Neural Networks, IEEE Transactions on, vol.16, no.3, pp.679-691

Er, M. J.; Wu, S.; Lu, J.; Toh, H.L. (2002), "Face recognition with radial basis function (RBF) neural networks," Neural Networks, IEEE Transactions on, vol.13, no.3, pp.697,710, May 2002.

Fatahi, S.; Zadkhosh, E.; Chalechale, A. (2013), "Face recognition with Linear Discriminant Analysis and neural networks," *Pattern Recognition and Image Analysis (PRIA), 2013 First Iranian Conference on*, pp.1-4

Feng, H. (2006), "Self-generation RBFNs using evolutional PSO learning", Neurocomputing, Volume 70, Issues 1–3, Pages 241-251

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", Frontiers of Computer Science in China, 2010, 4(4), 560-570

Fernandez-Navarro, F.; Hervas-Martinez, C. ; Gutierrez, P. A. ; Pena-Barragan, J. M. ; Lopez-Granados, F. (2012), "Parameter estimation of q-Gaussian radial basis functions neural networks with a hybrid algorithm for binary classification", Neurocomputing, 75, pp 123-134

Fister, I.; Fister I.Jr.; Yang, X.; Brest, J. (2013), "A comprehensive review of firefly algorithms", Swarm and Evolutionary Computation, Volume 13, December 2013, Pages 34-46.

Fornberg, B.; Piret, C. (2008), "On choosing a radial basis function and a shape parameter when solving a convective PDE on a sphere", Journal of Computational Physics, Volume 227, Issue 5, 20, pp 2758-2780

Gan, M.; Peng, H.; Dong, X. (2012), "A hybrid algorithm to optimize RBF network architecture and parameters for nonlinear time series prediction", Applied Mathematical Modelling, Volume 36, Issue 7, Pages 2911-2919.

Gao, M. L.; He, X.H.; Luo, D.S.; Jiang, J.; Teng, Q.Z. (2013), "Object tracking using firefly algorithm," Computer Vision, IET, vol.7, no.4, pp.227-237

Gokberk, B.; Akarun, L.; Alpaydin, E. (2002), "Feature selection for pose invariant face recognition," Pattern Recognition, 2002. Proceedings. 16th International Conference on, vol.4, no., pp.306,309 vol.4

Guo, G.; Dyer, C.R. (2003), "Simultaneous feature selection and classifier training via linear programming: a case study for face expression recognition," Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on, vol.1, no., pp.I-346,I-352 vol.1

Haddadnia, J.; Ahmadi, M.; Faez, K. (2002), "A hybrid learning RBF neural network for human face recognition with pseudo Zernike moment invariant," *Neural Networks, 2002. IJCNN '02. Proceedings of the 2002 International Joint Conference on*, vol.1, no., pp.11-16

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 Recognition, Volume 36, Issue 5, Pages 1187-1202.

Haddadnia, J.; Faez, K.; Moallem, P. (2001), "Neural network based face recognition with moment invariants," *Image Processing, 2001. Proceedings. 2001 International Conference on*, vol.1, no., pp.1018-1021

Han, H.; Chen, Q.; Qiao, J. (2010), "Research on an online self-organizing radial basis function neural network", J Neural Computing and Applications, Vol 19, Springer-Verlag, 2010, pp 667-676.

Harandi, M.T.; Ahmadabadi, M.N.; Araabi, B.N.; Lucas, C. (2004), "Feature selection using genetic algorithm and it's application to face recognition," Cybernetics and Intelligent Systems, 2004 IEEE Conference on, vol.2, no., pp.1368-1373

Hardy, R. L. (1971), "Multiquadric equations of topography and other irregular surfaces," J. geophys. Res. 76, pp 1905-1915

Hardy, R. L. (1990), "Theory and Applications of multiquadric-biharmonic method 20 years of discovery 1968-1988," Computers and mathematics with applications, Vol 19, Issues 8-9, pp 163-208

Harpham, C.; Dawson, C.W. (2006), "The effect of different basis functions on a radial basis function network for time series prediction: A comparative study," Neurocomputing, Vol 69, Issues 16-18, pp 2161-2170

Haykin, S. (1999), "Neural Networks: A Comprehensive Foundation", 2nd Edition, Prentice-Hall, Chapter 5, pp 256-317

Ho, H. T.; Chellappa, R. (2013), "Pose-Invariant Face Recognition Using Markov Random Fields," *Image Processing, IEEE Transactions on*, vol.22, no.4, pp.1573-1584

Holland, J. H.; Reitman, J. S. (1977), "Cognitive systems based on adaptive algorithms",*SIGART Bull.* 63, 49-49

Hruschka, E.R.; Campello, R.J.G.B.; Freitas, A.A.; de Carvalho, A.C.P.L.F. (2009), "A Survey of Evolutionary Algorithms for Clustering," Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, vol.39, no.2, pp.133-155

Hu, H. (2011), "Variable lighting face recognition using discrete wavelet transform", Pattern Recognition Letters, Volume 32, Issue 13, Pages 1526-1534

Hu, X. (2014), "Research for Face Recognition Based on Gabor Wavelet and Sparse Representation," Intelligent Systems Design and Engineering Applications (ISDEA), 2014 Fifth International Conference on, pp.764-767

Huang, C.-S.; Lee, C.-F.; Cheng, A.H.-D. (2007a), "Error estimate, optimal shape factor, and high precision computation of multiquadric collocation method," Engineering analysis with boundary elements, 31, pp 614-623

Huang, D.; Zhao, W. (2005), "Determining the centers of radial basis probabilistic neural networks by recursive orthogonal least square algorithms", Applied Mathematics and Computation, Volume 162, Issue 1, Pages 461-473.

Huang, G.B.; Ramesh, M.; Berg, T.; Learned-Miller, E. (2007b), "Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments". University of Massachusetts, Amherst, Technical Report 07-49, October, 2007.

Huang, S.Y.; Lin, C.J. (2014), "Using Neural Networks with Differential Evolution Learning for Face Recognition," *Computer, Consumer and Control (IS3C), 2014 International Symposium on*, pp.376-379

Izakian, H.; Abraham, A. (2011), "Fuzzy C-means and fuzzy swarm for fuzzy clustering problem", Expert Systems with Applications, Volume 38, Issue 3, Pages 1835-1838.

Jamil, N.; lqbal, S.; Iqbal, N. (2001), "Face recognition using neural networks," Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century. Proceedings. IEEE International, pp.277-281

Jing, X. ; Yao, Y.; Yang, J., Zhang, D. (2008), "A novel face recognition approach based on kernel discriminative common vectors (KDCV) feature extraction and RBF neural network", Neurocomputing, Volume 71, Issues 13–15, August 2008, Pages 3044-3048.

Kanan, H. R.; Faez, K. (2005), "PZMI and wavelet transform features in face recognition system using a new localization method," Industrial Electronics Society, 2005. IECON 2005. 31st Annual Conference of IEEE, pp 2690-2694

Kanan, H. R.; Faez, K.; Hosseinzadeh, M.(2007), "Face Recognition System Using Ant Colony Optimization-Based Selected Features," Computational Intelligence in Security and Defense Applications, 2007. CISDA 2007. IEEE Symposium on, pp.57-62

Kennedy, J.; Eberhart, R. (1995), "Particle swarm optimization," Neural Networks, 1995. Proceedings., IEEE International Conference on, vol.4, no., pp.1942-1948

Kerin, M.A.; Stonham, T.J. (1990), "Face recognition using a digital neural network with self-organising capabilities," *Pattern Recognition, 1990. Proceedings., 10th International Conference on*, vol.1, no., pp.738-741

Koc, M. ; Barkana, A. (2014), "Discriminative common vector approach based feature selection in face recognition", Computers & Electrical Engineering, Volume 40, Issue 8, November 2014, Pages 37-50

Lawrence, S.; Giles, C. L.; Tsoi, A. C.; Back, A. D. (1997), "Face recognition: a convolutional neural-network approach," Neural Networks, IEEE Transactions on, vol.8, no.1, pp.98-113

Lee, P.H.; Hsu, G.S.; Wang, Y.W.; Hung, Y.P. (2012), "Subject-Specific and Pose-Oriented Facial Features for Face Recognition Across Poses," Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, vol.42, no.5, pp.1357-1368

Lei, F.; Lu, Y.; Huang, W.; Yu, L.; Jia, L.(2012), "Fast Static Particle Swarm Optimization Based Feature Selection for Face Detection," Computational Intelligence and Security (CIS), 2012 Eighth International Conference on, pp.401-405

LFW Face Database: http://vis-www.cs.umass.edu/lfw/ (Downloaded in March 2015)

Li, M.; Wu, F.; Liu, X. (2007), "Face Recognition Based on WT, FastICA and RBF Neural Network," in Natural Computation, 2007. ICNC 2007. Third International Conference on, vol.2, no., pp.3-7

Li, X.; Fei, S.; Zhang, T. (2009), "Novel Dimension Reduction Method of Gabor Feature and its Application to Face Recognition," in Image and Signal Processing, 2009. CISP '09. 2nd International Congress on, pp.1-5,

Lin, J.; Ming, J.; Crookes, D. (2011), "Robust face recognition with partial occlusion, illumination variation and limited training data by optimal feature selection," *Computer Vision, IET*, vol.5, no.1, pp.23-32

Liu, C.; Wechsler, H. (2003), "Independent component analysis of Gabor features for face recognition," *Neural Networks, IEEE Transactions on*, vol.14, no.4, pp.919-928

Liu, R.; Feng, W.; Zhu, M. (2013),"Expression and lighting invariant face recognition using fast tree-based matching," *Electronics Letters*, vol.49, no.22, pp.1379-1381

Liu. N.; Wang, H. (2008), "Feature selection in frequency domain and its application to face recognition," Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on, pp.3967-3972

Lu, J. ; Zhao, J.; Cao, F. (2014), "Extended feed forward neural networks with random weights for face recognition," Neurocomputing, Volume 136, Pages 96-102

Lu, J.; Plataniotis, K.N.; Venetsanopoulos, A.N. (2003), "Face recognition using LDA-based algorithms," *Neural Networks, IEEE Transactions on*, vol.14, no.1, pp.195-200

Mao, K.Z. (2002), "RBF neural network center selection based on Fisher ratio class separability measure," Neural Networks, IEEE Transactions on, vol.13, no.5, pp.1211-1217

Martinez, A. M.; Kak, A.C. (2001), "PCA versus LDA," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol.23, no.2, pp.228,233, Feb 2001

Martinez, A.M. ; Benavente, R. (1998)," The AR Face Database. CVC Technical Report #24", June 1998.

Mendel, J. M. and McLaren, R. W. (1970), "Reinforcement learning control and pattern recognition systems", Adaptive, Learning and Pattern Recognition Systems: Theory and Applications (Mendel, J. M. and Fu, K. S., editors), pages 287-318. Academic Press, New York.

Nicholl, P.; Ahmad, A.; Amira, A. (2010), "Optimal discrete wavelet transform (DWT) features for face recognition," Circuits and Systems (APCCAS), 2010 IEEE Asia Pacific Conference on, pp.132-135

Oh, S. ; Yoo, S.; 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 Systems with Applications, Volume 40, Issue 5, Pages 1451-1466.

Oh, S.; Kim, W.; Pedrycz, W.; Seo, K.(2014), "Fuzzy Radial Basis Function Neural Networks with information granulation and its parallel genetic optimization," Fuzzy Sets and Systems, Volume 237, 16 February 2014, Pages 96-117

Oh, S.; Kim, W.;Pedrycz, W.; Joo, S. (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, Volume 78, Issue 1, Pages 121-132.

Ou, F.; Liu, C.; Lee, Y.; Ou, Z. (2007), "Evaluation and Selection of Discriminating Gabor Features for Face Recognition," Automatic Identification Advanced Technologies, 2007 IEEE Workshop on, pp.93-98

Poggio, T.; Girosi, F. (1990)," Networks for approximation and learning", Proceedings of the IEEE, 1990, Vol 78, issue 9, pp 1481-1497.

Qasem, S. N.; Shamsuddin, S. M.; Zain, A. M. (2012), "Multi-objective hybrid evolutionary algorithms for radial basis function neural network design", Knowledge-Based Systems, Volume 27, Pages 475-497

Qiakai, N.; Chao, G.; Jing, Y. (2012), "Research of face image recognition based on probabilistic neural networks," *Control and Decision Conference (CCDC), 2012 24th Chinese*, pp.3885-3888

Radji, N.; Cherifi, D.; Azrar, A. (2013), "Subband selection in Wavelet Packet Decomposition for face recognition," Sciences and Techniques of Automatic Control and Computer Engineering (STA), 2013 14th International Conference on, pp.494-500

Ramadan, R.M.; Abdel-Kader, R.F. (2009), "Particle swarm optimization for human face recognition," Signal Processing and Information Technology (ISSPIT), 2009 IEEE International Symposium on, pp.579-584

Rocha, H. (2009),"On the selection of the most adequate radial basis function", Applied Methematical Modelling, 33, pp 1573-1583

Sarra, S.A. (2006), "Integrated multiquadric radial basis function approximation methods", Computers and Mathematics with applications, 51, pp 1283-1296

Sattiraju, M.; Vikram Manikandan, M.; Manikantan, K.; Ramachandran, S. (2013), "Adaptive BPSO based feature selection and skin detection based background removal for enhanced face recognition," Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), 2013 Fourth National Conference on, pp.1-4

Senthilnath, J. ; Omkar, S.N. ; Mani, V. (2011), "Clustering using firefly algorithm: Performance study, Swarm and Evolutionary Computation," Volume 1, Issue 3, Pages 164-171.

Sing, J. K. ; Thakur, S.; Basu, D. K.; Nasipuri, M.; Kundu, M. (2009), "High Speed Face Recognition using self adaptive radial basis function neural networks", Neural Computing and Applications, Vol. 18, pages 979-990 .

Sing, J. K.; Basu, D. K.; Nasipuri, M.; Kundu, M. (2007), "Face recognition using point symmetry distance-based RBF network, Applied Soft Computing, Volume 7, Issue 1, Pages 58-70.

Slavkovic, M.; Reljin, B.; Gavrovska, A.; Milivojevic, M. (2013), "Face recognition using Gabor filters, PCA and neural networks," *Systems, Signals and Image Processing (IWSSIP), 2013 20th International Conference on,* pp.35-38

Srinivasan, A.; Balamurugan, V. (2013), "A new framework for 3D face reconstruction for self-occluded images", Int. J. of Computational Vision and Robotics, vol. 3, no. 4, pp 308-325

Taffar, M.; Miguet, S.; Benmohammed, M. (2013), "Viewpoint invariant model for face detection", Int. J. of Computational Vision and Robotics, vol. 3, no. 3, pp 182-196

Taylor, W.K. (1967), "Machine learning and recognition of faces", *Electronics Letters,* vol.3, no.9, pp.436-437

Tran, C.; Lee, T.; Chang, L.; Chao, P. (2014), "Face Description with Local Binary Patterns and Local Ternary Patterns: Improving Face Recognition Performance Using Similarity Feature-Based Selection and Classification Algorithm," Computer, Consumer and Control (IS3C), 2014 International Symposium on, pp.520-524

Tsai, C.H.; Kolibal, J.; Li, M. (2010), "The golden search algorithm for finding a good shape parameter for meshless collocation methods," Engineering analysis with boundary elements, 34, pp 738-746

Tsekouras, G. E. ; Tsimikas, J. (2013), "On training RBF neural networks using input–output fuzzy clustering and particle swarm optimization, Fuzzy Sets and Systems", Volume 221, Pages 65-89.

Turk, M.A.; Pentland, A.P. (1991),"Face recognition using eigenfaces," *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR '91., IEEE Computer Society Conference on,* pp.586-591

Utsumi, Y.; Iwai, Y.; Yachida, M. (2006), "Performance Evaluation of Face Recognition in the Wavelet Domain," Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on, pp.3344-3351

Vignolo, L.; Milone, D.; Behaine, C.; Scharcanski, J. (2012), "An evolutionary wrapper for feature selection in face recognition applications," Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on, pp.1286-1290

Wang, B.; Li, W.; Yang, W.; Liao, Q. (2011), "Illumination Normalization Based on Weber's Law With Application to Face Recognition," *Signal Processing Letters, IEEE,* vol.18, no.8, pp.462-465

Wang, D.; Zeng, X.; Keane, J.A. (2012), "A clustering algorithm for radial basis function neural network initialization", Neurocomputing, Volume 77, Issue 1, Pages 144-155.

Wertz, J.; Kansa, E.J.; Ling, L. (2006), "The role of the multiquadric shape parameters in solving elliptic partial differential equations", Computers and Mathematics with Applications, 51, pp 1335-1348.

Wiskott, L.; Fellous, J.-M.; Kuiger, N.; von der Malsburg, C. (1997), "Face recognition by elastic bunch graph matching," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol.19, no.7, pp.775-779.

Wong, Y. W.; Seng, K. P.; Ang, Li-Minn, (2011), "Radial Basis Function Neural Network With Incremental Learning for Face Recognition," Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, vol.41, no.4, pp.940-949.

Xiao, R.; Li, W.; Tian, Y.; Tang, X. (2006), "Joint Boosting Feature Selection for Robust Face Recognition," Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, vol.2, no., pp.1415-1422

Xiao-Dong, L.; Wei, Y. (2012), "Selection algorithm of Gabor Kernel for face recognition," in Control Conference (CCC), 2012 31st Chinese, pp.3839-3843

Xu, T.; Li, B.; Wang, B. (2003), "Face detection and recognition using neural network and hidden Markov models," *Neural Networks and Signal Processing, 2003. Proceedings of the 2003 International Conference on*, vol.1, no., pp.228-231

Yale, Yale Face Database, http://vision.ucsd.edu/datasets/yale_face_dataset_original/yalefaces.zip (Downloaded in 2012)

Yang, X. S. (2010), "Firefly Algorithm, Stochastic Test Functions and Design Optimisation", Int. J. Bio-Inspired Computation, Vol. 2, No. 2, pp.78–84.

Yang, X. S. (2011), "Metaheuristic optimization: algorithm analysis and open problems", in: Proceedings of 10th International Symposium on Experimental Algorithms (SEA 2011) (Eds. P. M. Pardalos and S. Rebennack), Kolimpari, Chania, Greece, May 5-7 (2011), Lecture Notes in Computer Sciences, Vol. 6630, pp. 21-32.

Yang, X. S. (2013), "Swarm intelligence based algorithms: a critical analysis", Evolutionary Intelligence, Springer Berlin Heidelberg, pp 1864-5909

Yang, X. S.(2008), "Nature-Inspired Metaheuristic Algorithms", Luniver Press,, Chapter 10

Yang, X.S.; Deb, S.; Loomes, M.; Karamanoglu, M. (2013), "A framework for self-tuning optimization algorithm, Neural Computing and Applications", vol. 23, no. 7-8, pp 2051-2057.

Yu, C.; Jin, B.; Lu, Y.; Chen, X.; Yi, Z.; Zhang, K.; Wang, S. (2013), "Multi-threshold Image Segmentation Based on Firefly Algorithm," Intelligent Information Hiding and Multimedia Signal Processing, 2013 Ninth International Conference on, pp.415-419

Yu, J.; Duan, H. (2013), "Artificial Bee Colony approach to information granulation-based fuzzy radial basis function neural networks for image fusion", Optik - International Journal for Light and Electron Optics, Volume 124, Issue 17, Pages 3103-3111.

Yu, M.; Yan, G.; Zhu, Q. W. (2006), "New Face Recognition Method Based on DWT/DCT Combined Feature Selection," Machine Learning and Cybernetics, 2006 International Conference on, pp.3233-3236

Yujie, H.; Jie, L.; Shi, Y. (2014), "A multi-condition relighting with optimal feature selection to robust face recognition with illumination variation," Communications, China, vol.11, no.6, pp.99-107

Zhang, B.; Zhang, H.; Ge, S. (2004), "Face recognition by applying wavelet subband representation and kernel associative memory," Neural Networks, IEEE Transactions on, vol.15, no.1, pp.166-177

Zhang, D.; Zuo, W. (2007), "Computational Intelligence-Based Biometric Technologies," Computational Intelligence Magazine, IEEE, vol.2, no.2, pp.26 - 36

Zhang, G.P. (2000), "Neural networks for classification: a survey," Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, vol.30, no.4, pp.451-462

Zhang, H.; Kiranyaz, S.; Gabbouj, M. (2014), "Cardinal sparse partial least square feature selection and its application in face recognition," Signal Processing Conference (EUSIPCO), 2014 Proceedings of the 22nd European, pp.785-789

Zhang, H.; Zhang, B.; Huang, W.; Tian, Q. (2005), "Gabor wavelet associative memory for face recognition," *Neural Networks, IEEE Transactions on*, vol.16, no.1, pp.275-278

Zhang, J.; Yan, Y.; Lades, M. (1997), "Face recognition: eigenface, elastic matching, and neural nets," *Proceedings of the IEEE*, vol.85, no.9, pp.1423-1435

Zhao, L.; Cai, Y.; Li, J.; Xu, X. (2005), "Face Recognition Based on Discrete Cosine Transform and Support Vector Machine," *Neural Networks and Brain, 2005. ICNN&B '05. International Conference on*, vol.2, no., pp.1248-1252

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

Zhenhua Chai; Zhenan Sun; Mendez-Vazquez, H.; Ran He; Tieniu Tan (2014), "Gabor Ordinal Measures for Face Recognition," Information Forensics and Security, IEEE Transactions on, vol.9, no.1, pp.14-26

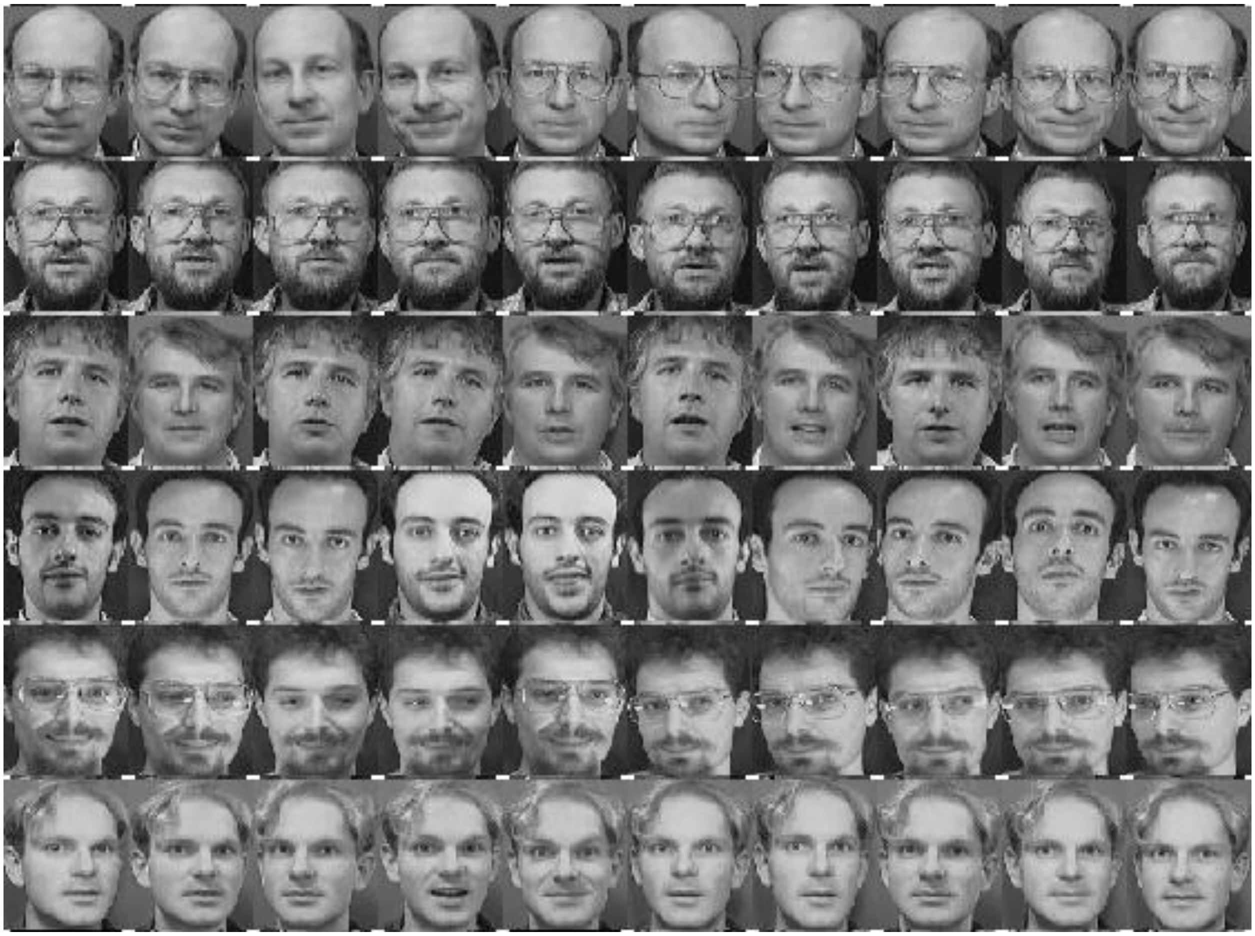
Zhifeng Li; Unsang Park; Jain, A.K. (2011), "A Discriminative Model for Age Invariant Face Recognition," *Information Forensics and Security, IEEE Transactions on*, vol.6, no.3, pp.1028-1037

Appendix

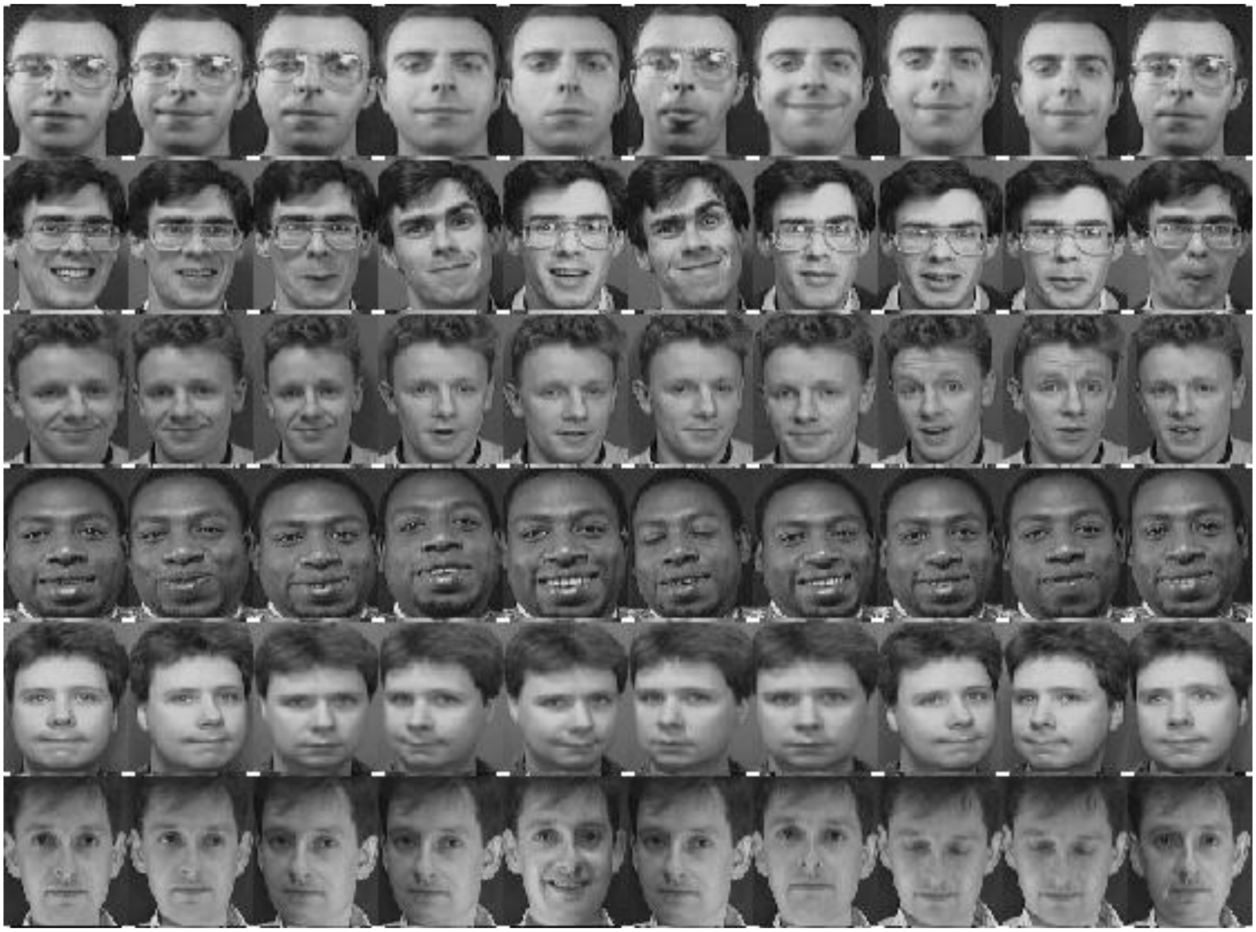
ORL Face Database

ORL Face Database

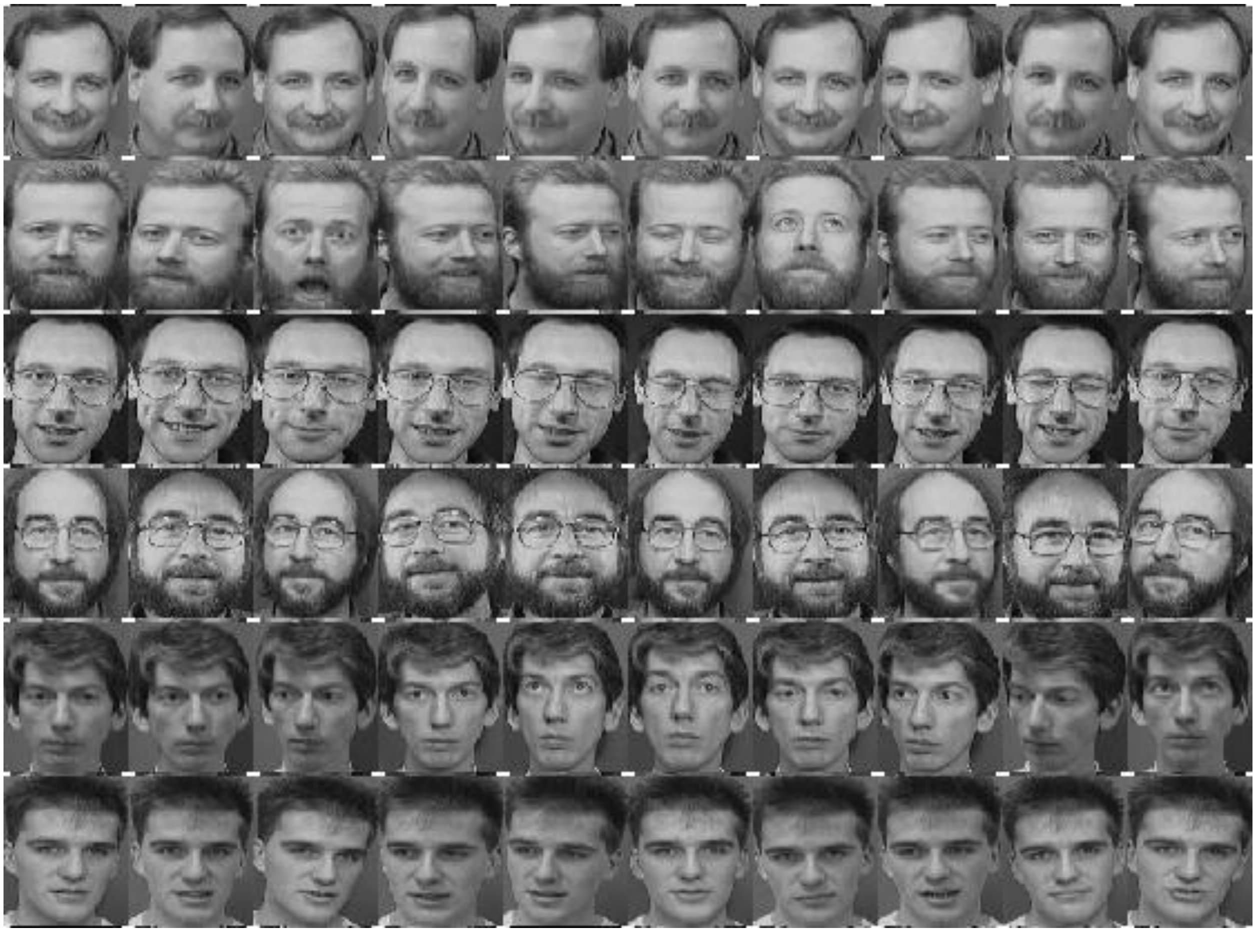
ORL Face Database



ORL Face Database

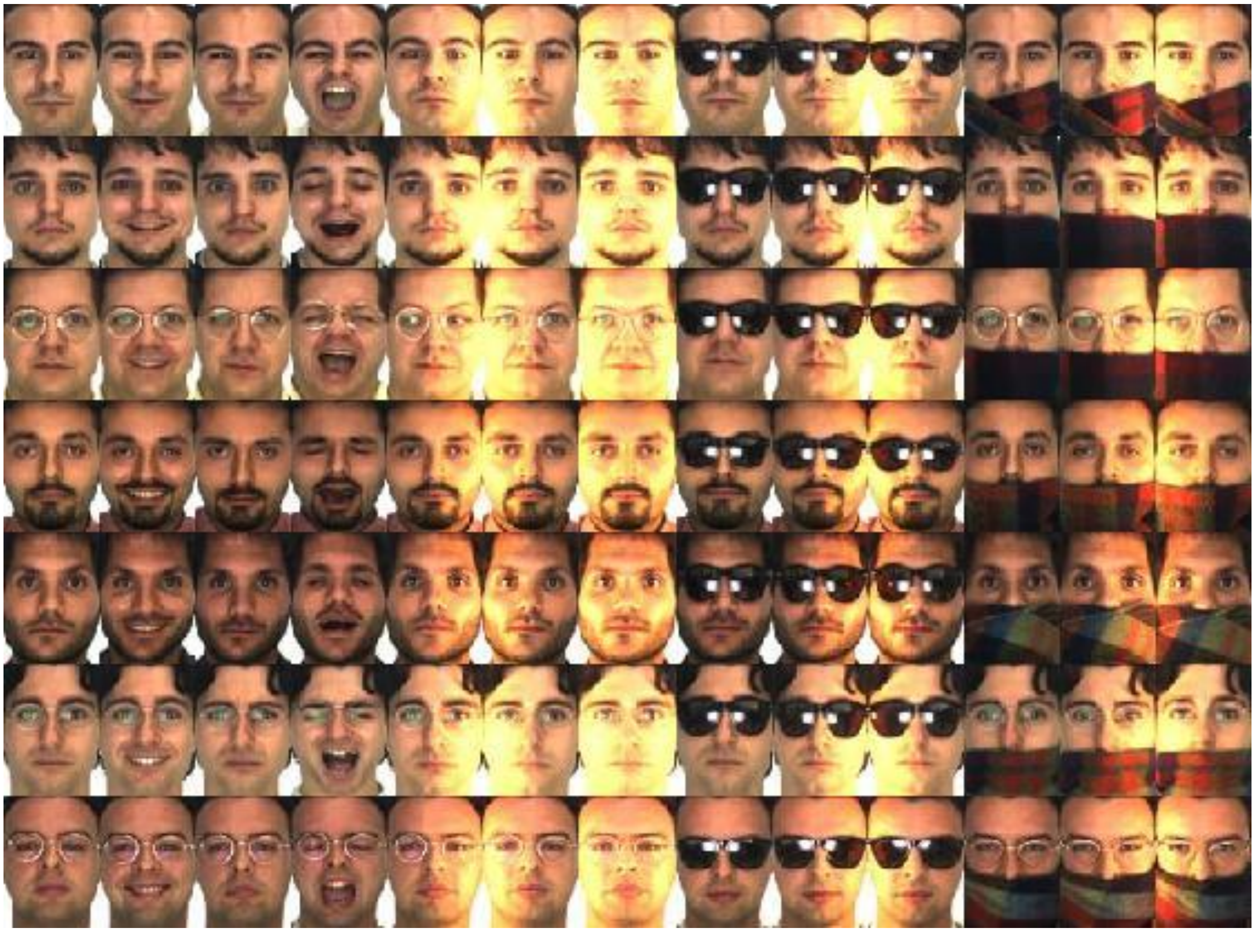


ORL Face Database

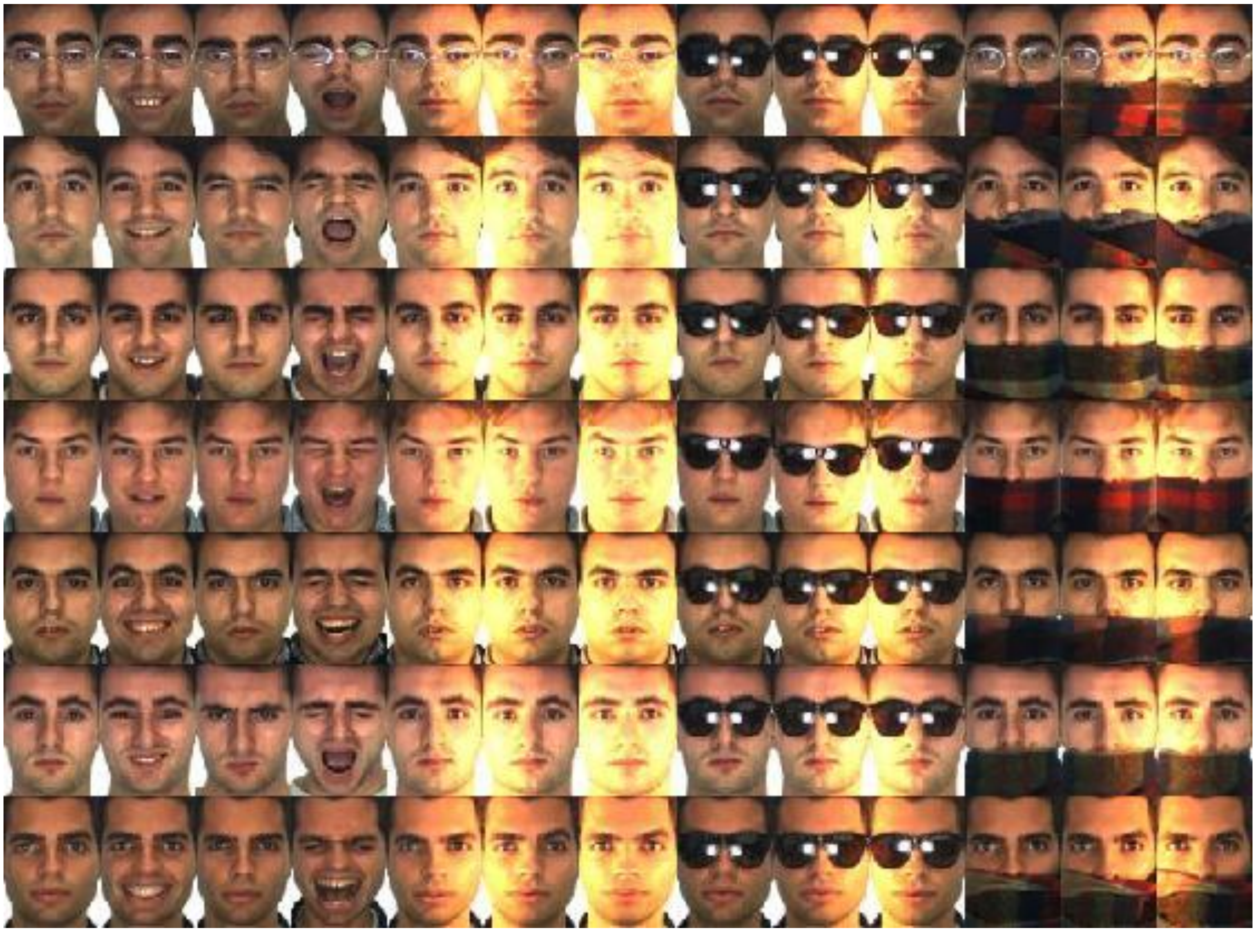


ORL Face Database

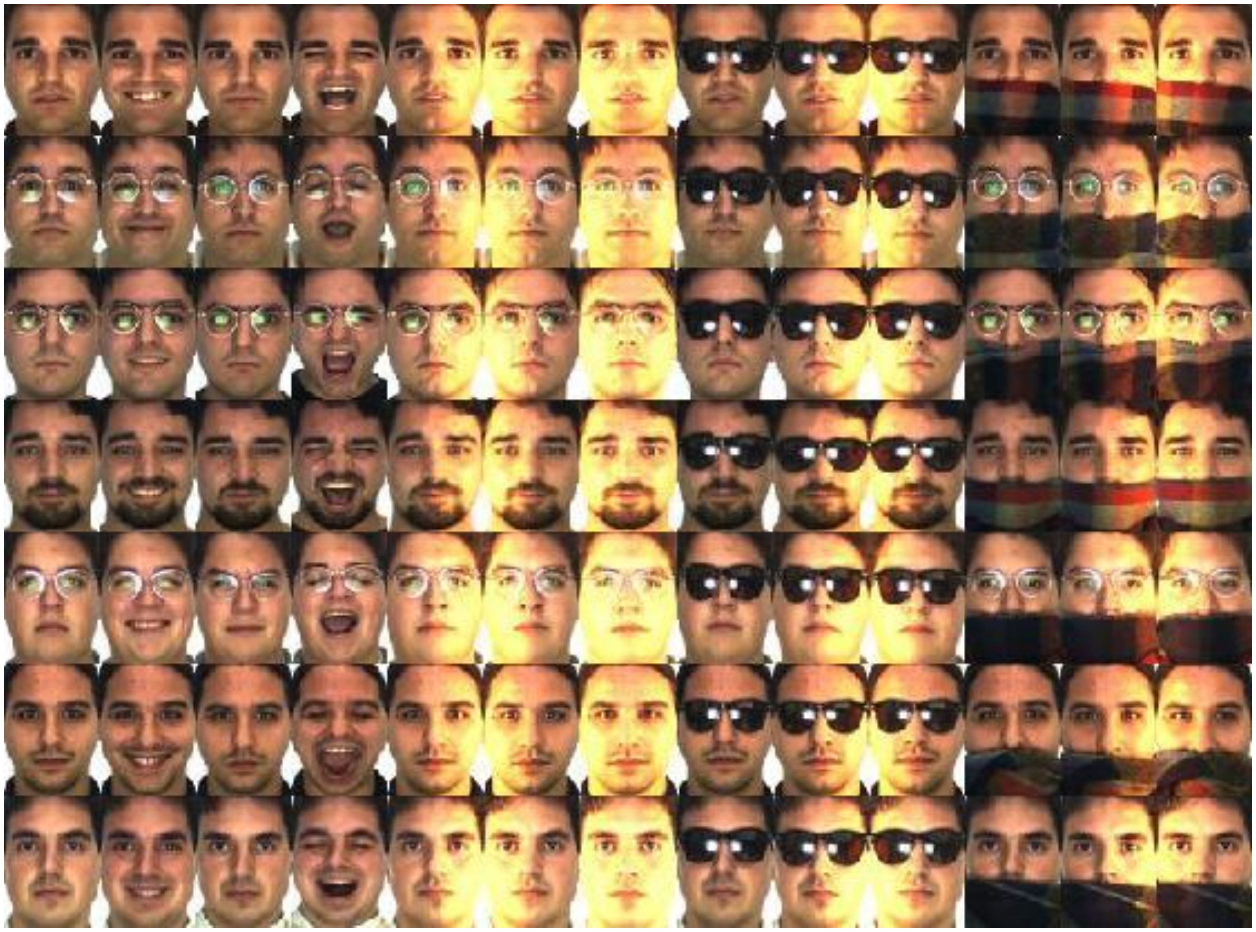
AR Face Database



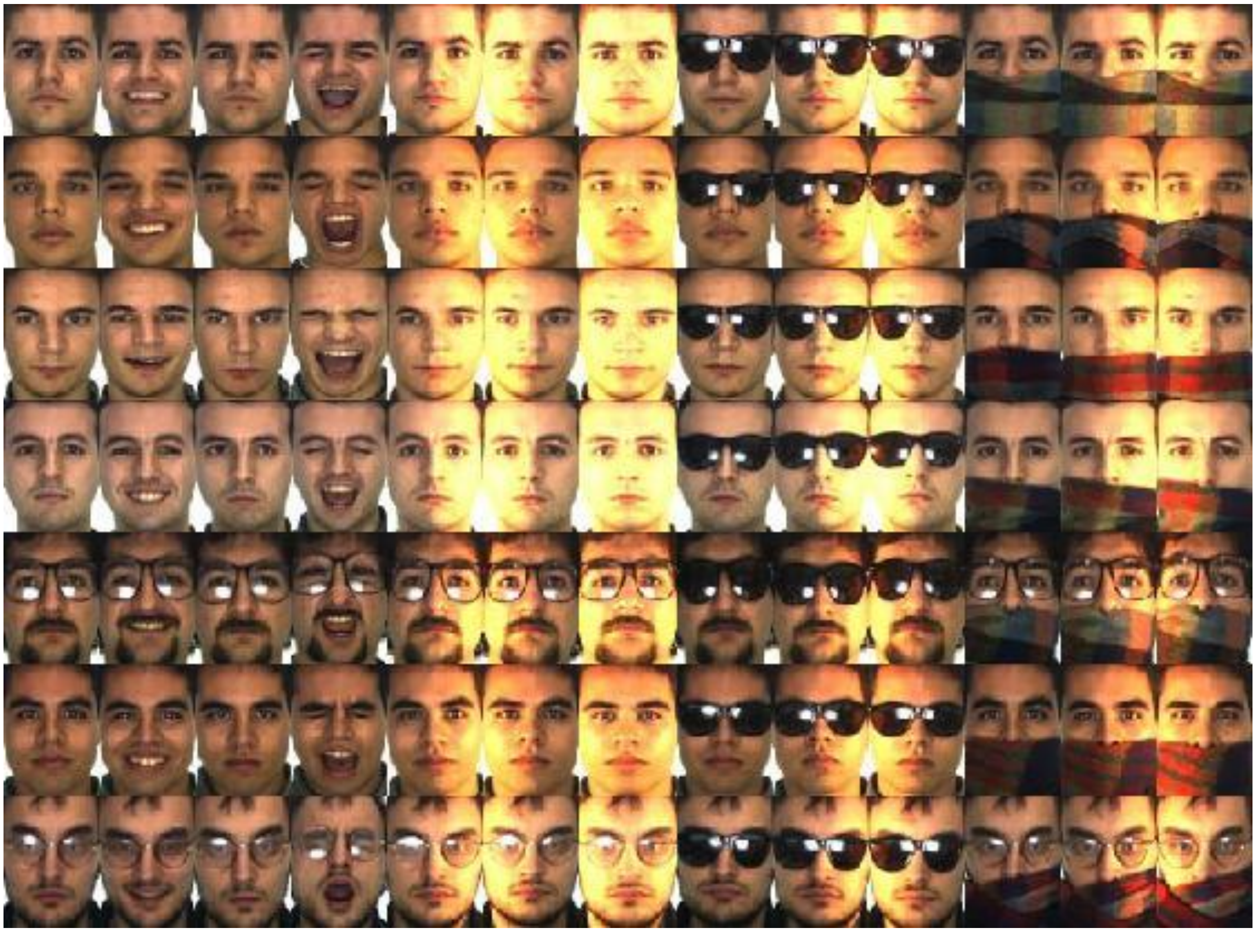
AR Face Database



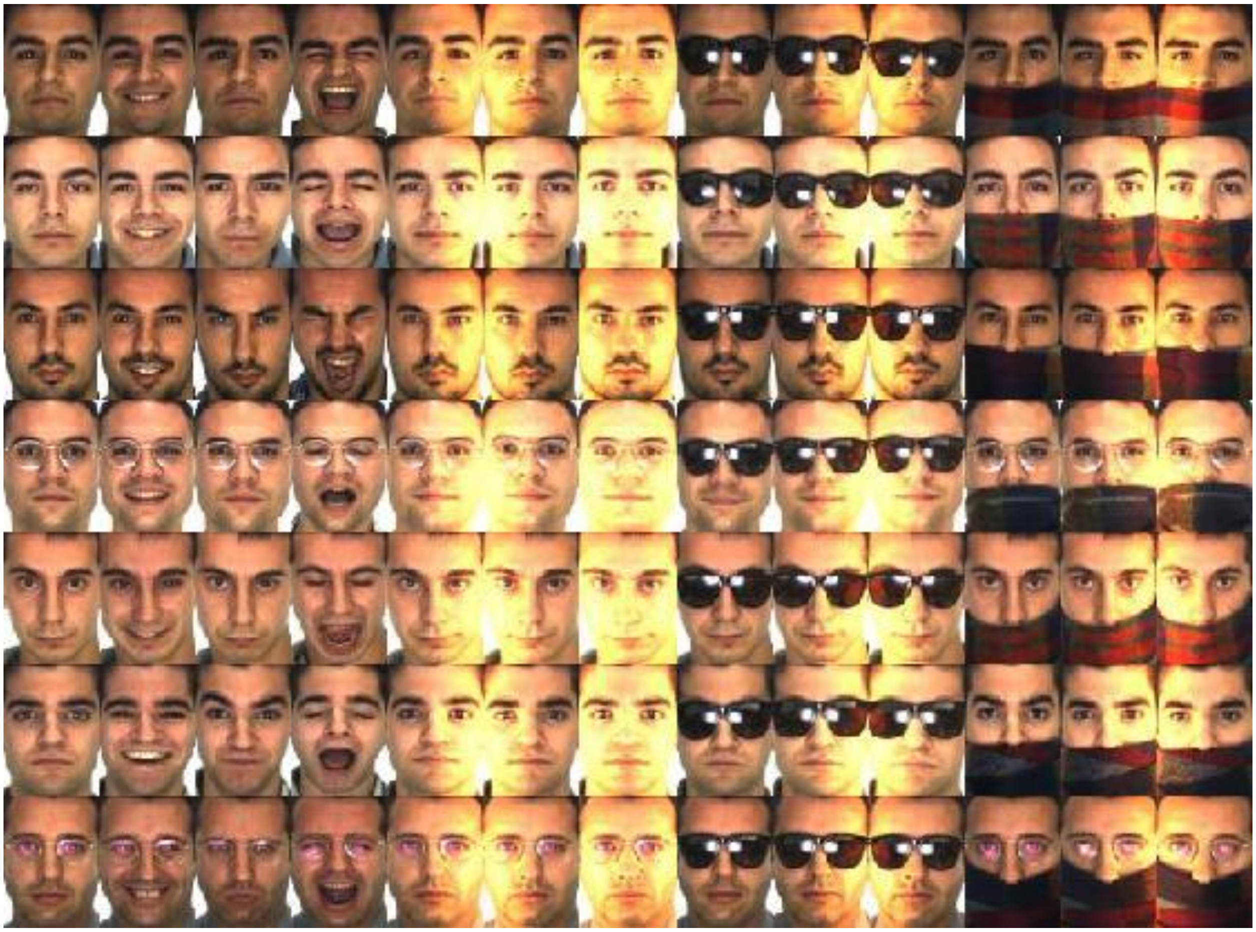
AR Face Database



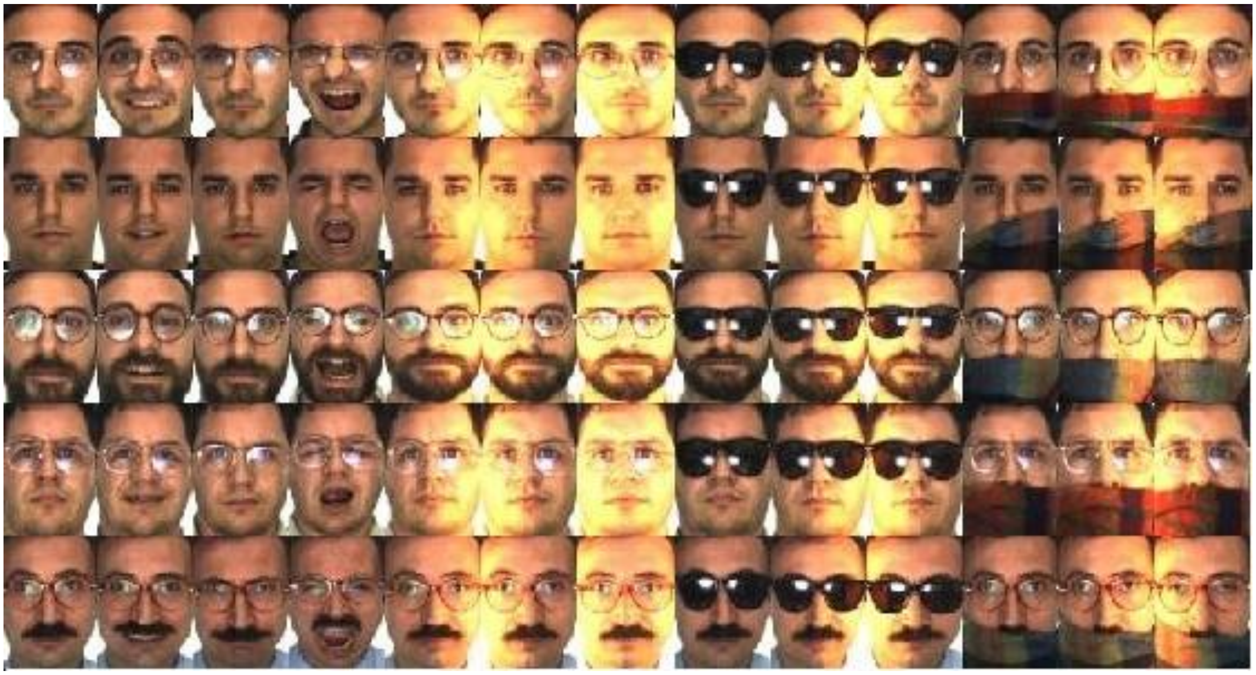
AR Face Database



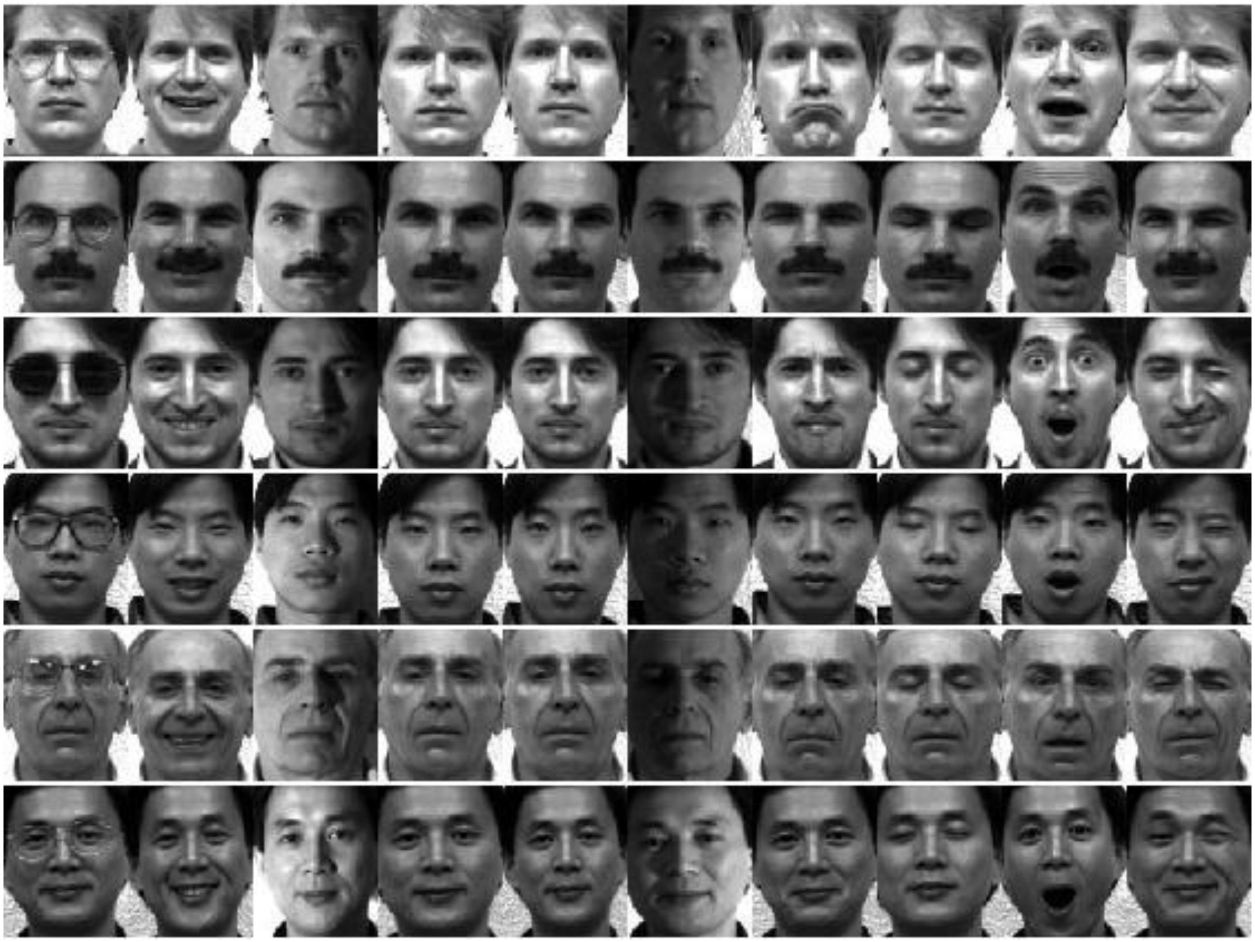
AR Face Database



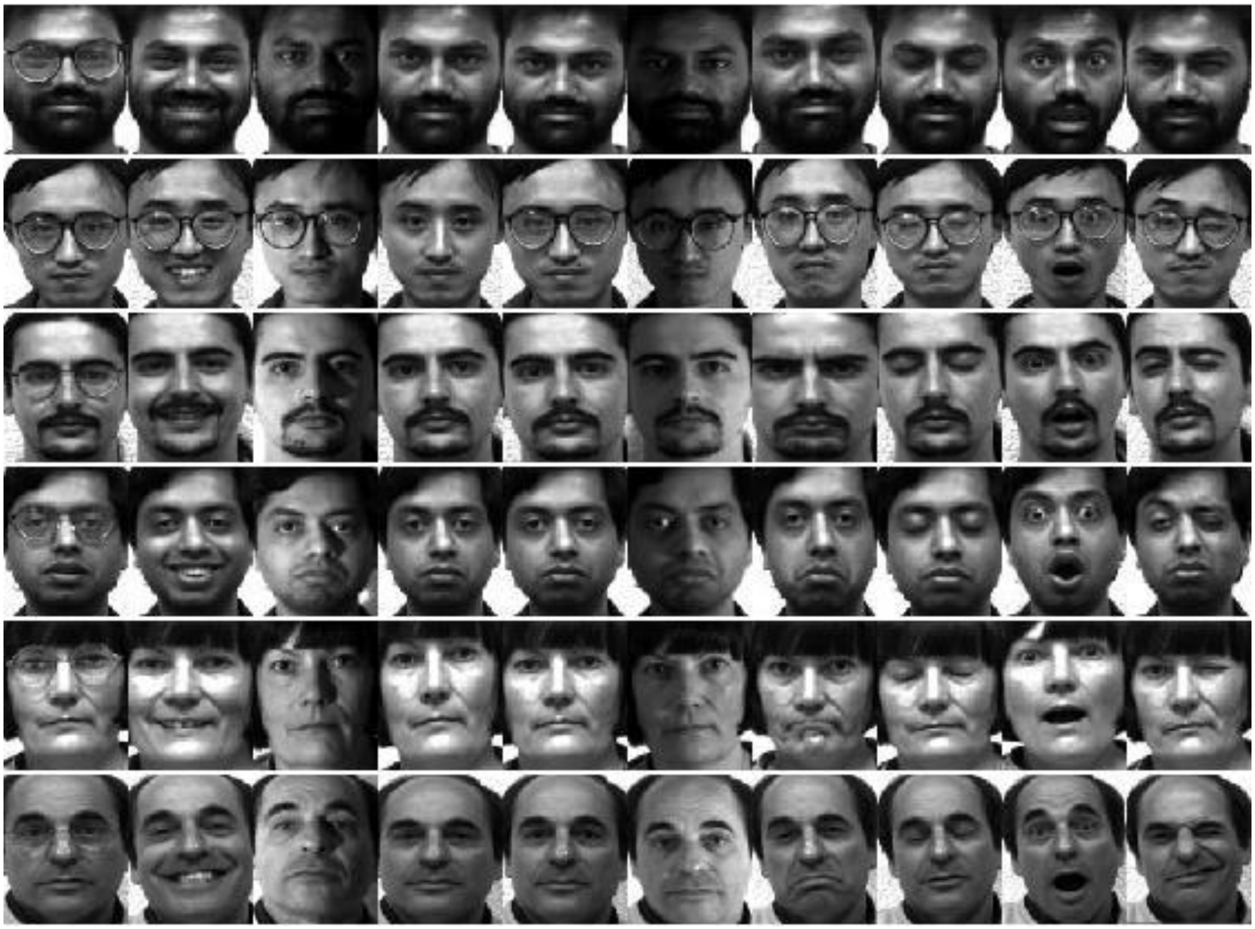
AR Face Database



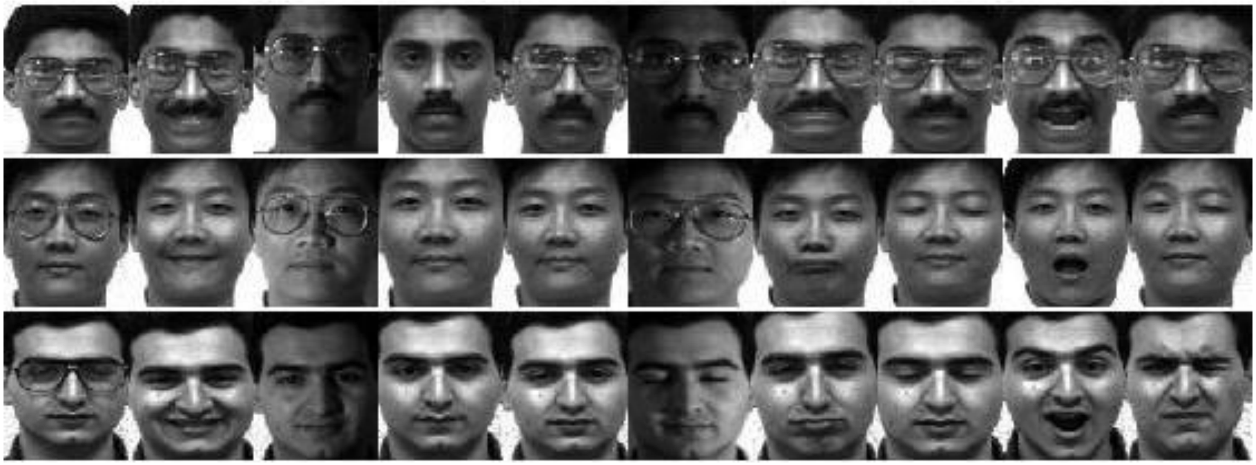
Yale Face Database



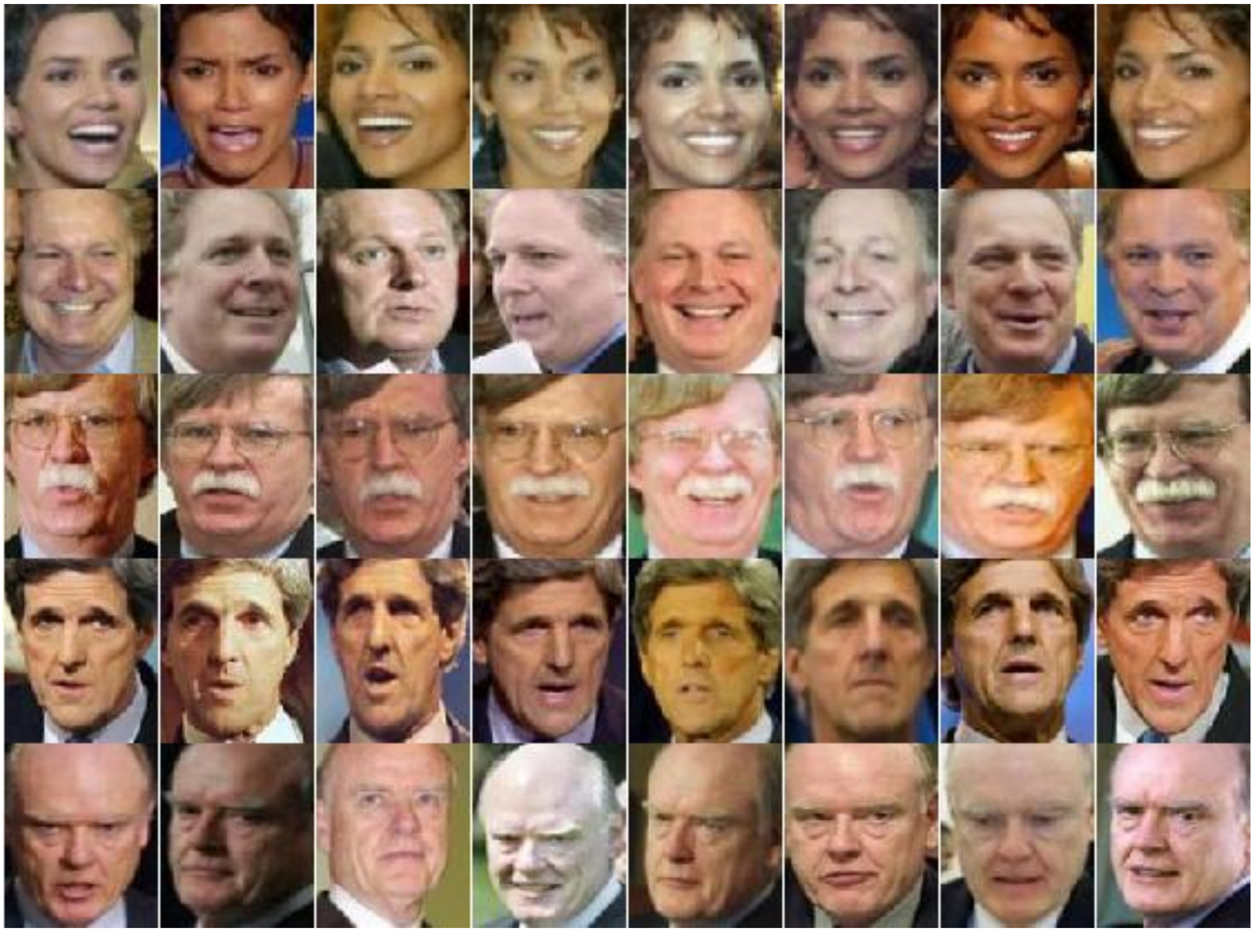
Yale Face Database



Yale Face Database



LFW Face Database



LFW Face Database

Biography

Biography of the Supervisor

Surekha Bhanot received B.E.(Hons), Mechanical Engineering and M.Phil. (Instrumentation) degrees from BITS Pilani, and Ph.D. degree from Indian Institute of Technology (IIT) Roorkee, in 1979, 1983 and 1995, respectively. In 1979, she joined BITS Pilani as Teaching Assistant. In 1983, she joined Thapar Institute of Engineering and Technology (TIET) Patiala, in 2002, she joined BITS Pilani, Pilani campus. Her research area of interest includes artificial intelligence applications in sensors, process modeling, control, medical instrumentation. She has written a book on Process control under Oxford University Publication.

Biography of the Candidate

Vandana Agarwal received her Master of Science (M.Sc.) degree in Mathematics from Indian Institute of Technology, Kanpur in 1989 and Master of Technology (M.Tech.) in Computer Applications from Indian Institute of Technology, New Delhi in 1991. She possesses an experience of about seventeen years in teaching and research. She worked as Scientist at Indian Institute of Remote Sensing(IIRS), Dehradun, as Lecturer at Indian Institute of Information Technology (IIIT), Allahabad and has been working as Lecturer at Birla Institute of Technology and Science (BITS), Pilani since the year 2008. Her research interests include Pattern Recognition, Neural Networks and Evolutionary Algorithms.