

Single Neuron Cascaded Neural Network based Face Recognition System

A. John Dhanaseely*, S. Himavathy and E. Srinivasan

dhanas1998@gmail.com

Abstract

Objectives: This paper analyses the performance of single neuron cascade neural network with existing neural networks such as feed forward and radial basis function neural network for face recognition system. **Methods:** Face recognition system performance is based on the feature extraction and neural network architecture. Principal component analysis method is used for feature extraction and the extracted feature vectors are used to train the network. Using single neuron cascade architecture images are recognized. In the hidden layer single neuron is added one by one till the performance is achieved. Network is trained by set of train image and tested by a new test image. Recognition accuracy is calculated based on the recognized image. **Findings:** An effective classifier is identified for face recognition system. In this paper single neuron cascaded neural network is proposed for classification. In Feed forward Neural network the neurons in a layer get input from the previous layer and feed their output to the next layer. In Cascade neural network architecture the input to any layer includes all the outputs and the inputs from previous layers, which results in a cascaded interconnection between layers leading to more compact structures. Network design by cascading one neuron at a time until the desired performance is obtained can be automated. The proposed method gives systematic approach to design. It combines the advantages of both single layer feed forward neural network and multilayer feed forward neural network. Performance of the network is presented in terms of average recognition accuracy. Number of training image and test images are gradually increased from lower number samples per subject. If the number of training images is more recognition accuracy is improved. Proposed single neuron cascaded neural network out performs the existing network. **Applications:** This proposed method plays vital role in the field of pattern recognition, vision and human computer interactive based applications such as face recognition, surveillance, criminal identification and pass port verification.

Keywords: Artificial Neural Network, Cascade Neural Network, Face Recognition, ORL database, Principal Component Analysis

1. Introduction

Face recognition has attained substantial attention from researcher's computer vision communities and pattern recognition field. Face recognition can be applied in Criminals identification, Passport verification, Electoral identification and Security measure. A formal method of classifying the faces was first proposed by Francis Galton¹. Galton proposed collecting facial profiles as curves, finding their norm, and then classifying other profiles by their deviations from the norm. Bledsoe^{2,3} tried for face recognition with computer system that classified faces based on the marks at the photograph by hand and for

classification normalised distance was used. Two central issues in face recognition are feature extraction and classification. Principal Component Analysis (PCA) method is mostly used for dimension reduction and salient feature extraction in appearance based approach. This technique has been successfully applied in the area of signal processing and pattern recognition. PCA is a statistical approach where the face images are represented as a subset of their Eigen vectors. The PCA algorithm finds optimal linear transformation to reduce the dimensionality of a vector by approximating it with most significant Eigen vectors based on Karhunen – Loeve transform. Sirovich and Kirby^{4,5} applied PCA to represent a face image Turk and Pentland⁶

*Author for correspondence

proposed the Eigen face method for Face Recognition System (FRS). PCA is used in biometric recognition such as face, fingerprint and palm print identification or verification. In face recognition system features are extracted by PCA. Different classifiers like Euclidean distance classifier, nearest neighbour and Bayesian classifier are used for classification. Theoretic literature describes the use of neural networks in face recognition due to its rapid classification capability. The Neural Networks⁷ (NN) are among the most successful classification systems that can be trained to perform complex functions in the field of pattern recognition. Lawrence et al.⁸, proposed Convolutional Neural Network approach for face recognition. Different network like Radial basis function⁹ are employed for face recognition. Eleyan et al.^{10,11}, Shows the face recognition system performance using feed forward neural network. The performance of a NN based classifier^{12,13} largely depends on the network architecture and learning algorithm. In this paper single neuron is cascaded to secure compact structure. In Cascade neural network architecture the input to any layer includes all inputs and the outputs of all previous layers. This results in a cascaded interconnection between layers leading to more compact structures. FRS system is Single Neuron Cascade (SNC) architecture with PCA^{14,15} features. The performance of this network is compared with Radial Basis Function network (RBF) Feed Forward Network (FFNN). ORL¹⁶ database is explored to compare the performance and to draw the conclusion.

The paper is organized as follows. Section II describes the face recognition system. Section III details the PCA based feature extraction. Section IV presents the Neural Network Architectures and different algorithm for classification V. Experiments results and comparison are shown in section VI. The major conclusions are drawn in section VII.

2. Face Recognition System

Face Recognition System (FRS) is shown in the Figure 1. This includes preprocessing, feature extraction and classification. Preprocessing is very useful in suppressing irrelevant information. It eliminates undesired distortions

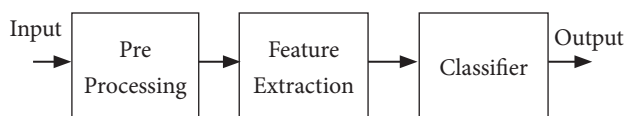


Figure 1. Block diagram of Face recognition system.

and enhances important features for further processing. In face recognition applications, the original input data is usually of high dimension. Feature extraction is a process of transforming the input data to a reduced set of salient features. In face recognition the feature set is classified using different techniques namely Euclidean distance classifier, Bayesian classifier and neural network classifier.

2.1 Preprocessing

Preprocessing is an operation at the lower level of abstraction. Image size normalization, histogram equalization, image sharpening, edge detection, filtering and segmentation are some of the preprocessing methods in image processing. The ORL data base is used in this paper and no preprocessing is performed as the images are noise free and there are no significant changes in illumination level.

2.2 Feature Extraction

In feature extraction image space is transformed into feature space. From feature space most descriptive feature vectors are chosen. Some of the feature extraction techniques are PCA (Principal Component Analysis), LDA (Linear Discriminant Analysis), KPCA (kernel PCA). In this work PCA based feature extraction is employed for dimensionality reduction.

2.3 Classification

A classification problem occurs when an object needs to be assigned into a predefined group or class based on a number of observed attributes related to that object. Conventional statistical classification procedures are built on the Bayesian decision theory. In these procedures, an underlying probability model must be assumed in order to calculate the posterior probability upon which the classification decisions made. One major limitation of the statistical method is that they work well, only when the underlying assumptions are satisfied. Different classifiers like nearest feature line, nearest feature space, Euclidean distance and neural network classifier are used for pattern recognition. In this paper different neural network classifiers are investigated.

3. Principal Component Analysis based Feature Extraction

Principal component analysis transforms several possibly correlated variables in to a smaller number of

uncorrelated principal components. The main process for feature extraction employing PCA can be stated as follows, a face image H in two-dimension with size $N \times N$ can be considered as a vector of dimension N^2 . Images of faces, being similar in overall configuration, will not be randomly distributed in this huge image space and thus can be described by a relatively low dimensional subspace. The aim of the principle component is to find the vectors that best account for the distribution of face images within the entire image space. These vectors define the subspace of face images called "face space". Each of these vectors is a linear combination of the original face images. These vectors are the Eigen vectors of the covariance matrix corresponding to the original face images.

Let the training set of face images be

$$H_1, H_2, \dots, H_M \quad (1)$$

Then the average of the set ϕ is as follows

$$\phi = \frac{1}{M} \sum_{i=1}^M H_i \quad (2)$$

Each face differs from the average by the vector

$$\Phi_i = H_i - \phi \quad (3)$$

This set of large vectors is then subject to PCA, which finds for set of ortho normal vectors U_v .

It describes the distribution of the data. The k^{th} vector U_k is chosen in such a way that

$$\lambda_k = \frac{1}{M} \sum_{i=1}^M (U_k^T \Phi_i)^2 \quad (4)$$

Is maximum subject to

$$U_i^T U_k = \delta_{ik} = \begin{cases} 1, & \text{if } i = k \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

The vectors u_k and the corresponding scalars λ_k are the Eigen vectors and Eigen values of the covariance matrix C

$$C = \frac{1}{M} \sum_{i=1}^M (\Phi_i \Phi_i^T) = A A^T \quad (6)$$

Where the matrix

$$A = [\Phi_1, \Phi_2, \dots, \Phi_M] \quad (7)$$

Since the covariance matrix C is a $N^2 \times N^2$ real symmetric matrix. To calculate the N^2 Eigen vectors and Eigen values for typical image sizes the following steps are done. Consider the Eigen vectors E_v of $A^T A$

Such that

$$A^T A E_v = \mu_i A E_v \quad (8)$$

Pre-multiplying both sides by matrix A ,

$$A^T A A E_v = \mu_i A E_v \quad (9)$$

Obviously, E_v are the Eigen vectors and μ_i are the corresponding Eigen values of covariance matrix C . We can then construct the $M \times M$ matrix $L = A^T A$,

$$\text{Where } L_{ij} = \Phi_i^T \Phi_j \quad (10)$$

And obtain the M Eigen vectors of L . These vectors determine linear combinations of the M training set face images to form the Eigen face

$$U_I = \sum_{k=1}^M \nu_{Ik} \Phi_k, \quad I = 1, 2, \dots, M \quad (11)$$

The calculations will be thus greatly reduced from the order of the number of pixels in the images (N^2) to the order of the number of images in the training set (M). The calculations become quite manageable since the training set of face images is relatively small ($M \ll N^2$) in practical applications. The associated Eigen values allow ranking the Eigen vectors according to their usefulness in characterizing the variation among the images.

The Eigen face images calculated from the Eigen vectors of L span a basis set that can be used to describe face images. Smaller number of Eigen face is adequate for face identification since accurate reconstruction of the image is not necessary. In the task of face recognition the Eigen faces span an n ($n < M$) dimensional subspace of the original N^2 image space are sufficient for reliable representation of the faces and the n significant Eigen vectors of the L matrix are chosen as those with the largest associated Eigen values. After that any new input face image (H_{new}) can be transformed into its Eigen face components as follows

$$\omega_k = u_k^T (H_{\text{new}} - \phi) \quad k = 1, 2, \dots, n \quad (12)$$

The weights from a projection vectors

$$\ell^T = [\omega_1, \omega_2, \omega_3, \dots, \omega_n] \quad (13)$$

This describes the contribution of each Eigen face in representing the input face image and treats the Eigen faces as a basis set for face images. These Eigen vectors are used for training the neural network

4. Neural Network Architectures and Different Algorithm for Classification

Artificial Neural Network (ANN) is an interconnected group of artificial neurons that uses a mathematical method for processing the information. Neural networks have been widely applied in pattern recognition for the reason that neural networks based classifiers can incorporate both statistical and structural information and can be trained to achieve better performance than the simple minimum distance classifiers. The classification ability of neural network to a large extent depends on its architecture. Most popularly used architectures namely Feed Forward (FFNN), Radial Basis Function (RBF) network and Cascade Neural Network (CNN) are investigated in this paper. The input to the network is an R-dimensional feature vector. The output of the network is the total number of subjects.

The learning algorithms are used to obtain the optimum parameters (weights and biases) of the network by minimizing the performance index defined in terms of output mean square error function. There are different types of learning algorithms reported in the literature to train neural network. These are mainly grouped into two types, first order approach algorithms (based on steep descent method) and second order approach algorithms. The popularly used first order approach algorithms are Back Propagation Algorithm with Momentum (BPM) and Variable Learning Rate Back Propagation Algorithm (VLBP). The popularly used second order approach algorithm is Levenberg-Marquardt (LM) algorithm.

4.1 Cascade Neural Network

Cascade Neural Network (CASNN) an architecture where layers of neurons are cascaded. Each neuron in the architecture includes weights, bias and a non-linear activation function. The weights of interconnections to the previous layer are called as “input weights” and the weights of the interconnections between the layers are called “link weights”. The commonly used tan-sigmoid / log-sigmoid activation function is used for all hidden layers while pure-linear function is used for output layer. To create a multilayer structure similar to MLFF-NN, hidden layers are added one by one and the whole network is trained repeatedly using the concept of moving weights so as to obtain compact network. This process continues, till the performance index is reached. A typical Cascaded architecture is

shown in Figure 2. The cascade architecture can have one neuron in each layer it is called single neuron cascaded architecture and is used in this paper. The use of SNC for face recognition has not been reported in literature.

p - Input vector, $p = [1, 2, \dots, R]$.

S^m - Number of neurons in the layer ‘ m ’ where $m = [1, 2, \dots, M]$ and $s^0 = p$.

$w_{i,R}^m$ - Input weight of neuron ‘ i ’ of layer ‘ m ’ for external input ‘ p ’.

$w_{i,j}^{m,k}$ - Link weight of neuron ‘ i ’ of layer ‘ m ’ for input from neuron ‘ j ’ of layer ‘ k ’.

b_i^m - Bias for neuron ‘ i ’ of layer ‘ m ’

f^m - Activation functions of all neurons in a layer ‘ m ’

a_i^m - Output of neuron ‘ i ’ of layer ‘ m ’

$$P_c (\text{Total parameters}) = \sum_{m=1}^M \sum_{q=0}^{m-1} S^m S^q + \sum_{m=1}^M S^m \quad (14)$$

Weights *Biases*

4.2 Feed Forward Network

Feed forward network provides a general frame work for representing nonlinear functional mapping between a set of input variables and set of output variables. The structure of the multi layer feed forward architecture is shown in Figure 3. The network consists of a number of layers and each layer output behaves as an input to the next layer. The hidden layers use tan sigmoid activation function and output layer uses linear activation function. The number of hidden layers and the number of neurons in each layer have to be identified with repeated trials.

$$\sum_{m=1}^M S^{m-1} S^m + \sum_{m=1}^M S^m \quad (15)$$

weights *biases*

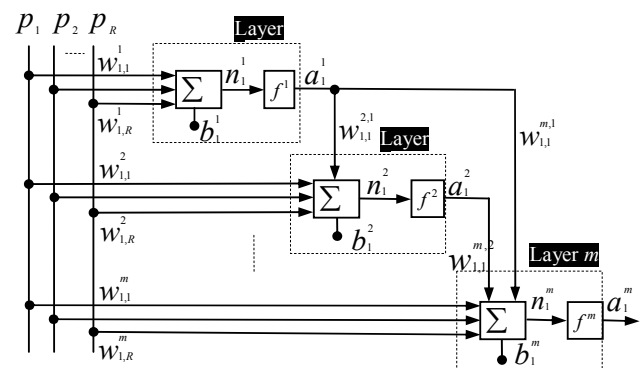


Figure 2. Cascade Architecture.

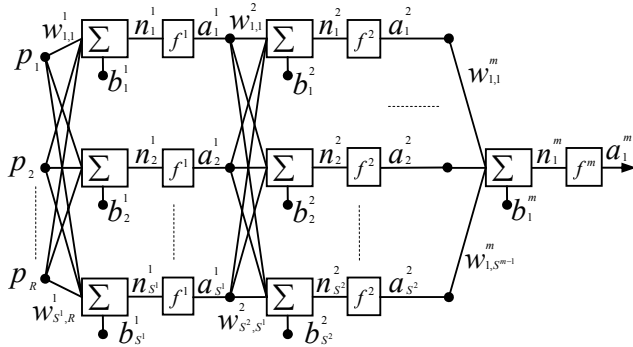


Figure 3. Multi-layer Feed Forward Architecture.

4.3 Radial Basis Function Network

Radial basis function network is a class of single hidden layer feed forward neural network. The schematic diagram of a RBF neural network is shown in Figure 4. The input nodes pass the input directly and the first layer connections are not weighted. The transfer functions in the hidden nodes are similar to the multivariate Gaussian density function,

$$h_i(x) = \exp \left[-\frac{\|x - \mu_j\|^2}{\sigma_j^2} \right] \quad (16)$$

Where σ is the width parameter, μ is the vector determining the center of basis function h and x is the dimensional input vector. Where x is the input vector, μ_j center and σ_j the spread of the corresponding Gaussian function. Each RBF unit has a significant activation over a specific region, thus each RBF represents a unique local neighborhood in the input space. The connections in the second layer are weighted and the output nodes are linear summation units. The training in RBF networks is done in three sequential stages as against the single optimization procedure followed for back propagation network. The first stage of the learning consists of determining the unit centers by the K-means clustering algorithm and unit width σ_j using a heuristic approach that ensures the smoothness and continuity of the fitted function. The width of any hidden node is taken as the maximum Euclidean distance between the identified centers. Finally, the weights of the second layer connections are determined by linear regression using a least-squares objective function. While RBF networks exhibit the same properties as back propagation networks such as generalization ability and robustness, they also have the additional advantage of fast learning and ability to detect outliers during estimation.

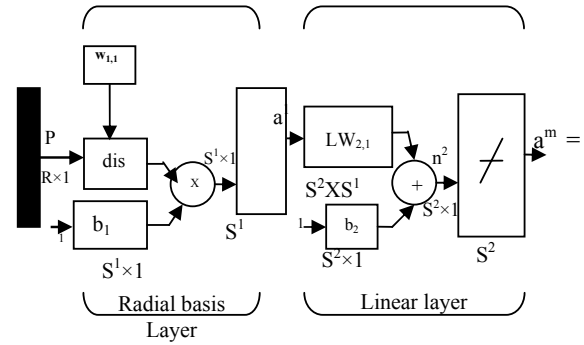


Figure 4. RBF Architecture.

4.4 Levenberg-Marquardt Algorithm (LM)

LM algorithm is a second approach algorithm that was designed especially for minimizing sums of squares functions. This is well suited to neural network training where the performance index is defined in terms of mean square error. It gives a nice compromise between the speed of the Gauss-Newton algorithm and the guaranteed convergence of the steepest descent method. When the scalar μ is decreased to zero, the algorithm becomes Gauss Newton. When μ is increased, it approaches the steep descent with a small learning rate. Gauss Newton's method is faster and more accurate near an error minimum, so if μ is decreased after each successful step (if there is a reduction in MSE) and is increased only when there is an increase in MSE

4.5 Back Propagation Algorithm with Momentum (BPM)

The learning rate employed in the Back Propagation (BP) algorithm determines the step size. Larger value of learning rate means faster learning of NN. This in turn would result in oscillations in the trajectory, close to convergence leading to instability. To overcome this difficulty and reduce oscillations in the trajectory, a momentum term is added to smoothen out the oscillations. The momentum factor allows larger learning rate and accelerates convergence. BPM is first order algorithm allowing faster convergence as compared to BP Algorithm. The update equations are represented in the following manner.

$$W^m(k+1) = W^m(k) + \eta \Delta W^m(k-1) - (1-\eta) \alpha S^m(a^{m-1})^T \quad (17)$$

$$b^m(k+1) = b^m(k) + \eta \Delta b^m(k-1) - (1-\eta) \alpha S^m \quad (18)$$

γ = Momentum Factor.

α = Learning rate.

S^m = Sensitivity factor of m^{th} layer.

4.6 Variable Learning Rate Back Propagation (VLBP)

Variable learning rate helps to speed up the convergence. Learning rate is increased on flat surfaces and decreased when the slope increases. This algorithm accelerates the rate of convergence with adaptable learning rate. This algorithm depends on several parameters such as ξ , ρ and η . The update equations are similar to BPM.

5. Experimental Results and Comparisons

In this section selection feature vectors, suitable learning algorithm, recognition rate for different combination of poses and overall recognition rate is discussed. All neural architectures use tangent hyperbolic function as activation function in the hidden layer and linear function in the output layer. The ORL data base is used. The ORL database contains 400 face images from 40 individuals. Some of the sample of the ORL database is shown in Figure 5. For each image the PCA features are obtained. To select the number of feature vectors most suitable for image recognition, an investigation is carried out. The 50% of the selected images are used for training and the rest is used for testing. At the beginning five vectors are taken to train the network the recognition rate is observed then number PCA vectors gradually increased and the recognition rate is plotted. It is observed if the number of vectors is increased more than 40 there is no improvement in the recognition accuracy. So the vector size is fixed to forty for all the networks.



Figure 5. Sample images of ORL database.

From the results the recognition accuracy is high when the length of the vectors is set to 40 and further increase in vectors does not improve the recognition accuracy. Hence it is concluded that 40 PCA vectors will be used for all the architectures. Investigation on Neural learning algorithm is investigated in the next section.

For performance comparison, all the three NN architectures are trained with same 40 feature vector, same learning algorithms, and same target MSE of 1×10^{-4} . The test MSE achieved for all the NN models is tabulated in Table 1.

From the Table 1, it is observed that all the three NN architectures trained with LM algorithm performed well as compared to BPM, VLR trained NN architectures. The network has number of inputs are forty, hidden neurons thirty and the output is forty, Hence, it is concluded that LM algorithm is most suitable for FRS. Three different FRS systems are built with the three architectures namely Single Neuron Cascaded Architecture (SNC), Feed Forward Architectures (FFNN) and Radial Basis Function (RBF) with tan sigmoid activation function. The same training and testing data are used to compare all the architectures. The output of the FRS is equal to the number of persons in the data base. Each person is assigned a number. Let y_i be the output of the network and $i = 1, 2, 3, \dots, n$ where n is the number of persons in the database. For a given i^{th} person the output $y_i = 1$, where as the rest of the outputs are zero. The output stage is a competitive network. The input output data thus created is used to train the network. In testing the new image which is to be recognized is given as input and the output is obtained. The

Table 1. Recognition Performance of SNC, RBF, FFNN architecture for different learning algorithms

Database	NN Architecture [input: hidden layers :output]	Test MSE (300epochs)	Training Algorithm	Recognition Rate (%)
ORL	SNC 40:1[30layers]: 40	0.00025	LM	87
		0.0166	VLBP	76.5
		0.3613	BPM	61
	RBF 40:30:40	0.00012	Gaussian	78
	FFNN 40:15:15:40	0.00031	LM	65
		0.0123	VLBP	55.5
		0.2674	BPM	52

maximum value of the output is declared the winner and used to identify the person. Input is selected randomly.

Different number of training samples per subject is taken. Some of the poses taken for testing and training and their recognition are rate shown in Table 2 for three combinations. To consider overall recognition rate number of ten different runs is taken. The average recognition rate with standard deviation is shown in Table 3.

For same number of input, hidden neuron and output SNC outperforms than the RBF and FFNN. The SNC based model is the most compact in structure and less complex in computation that enables easy implementation and faster execution. The performance graph is shown in Figure 7.

Table 2. Average recognition rate for SNC, FFNN and RBF architecture for three runs

Training Images		Test Images		Neural Network		
Number of Poses for training	Training Poses	Number of Poses for test	Test poses	FFNN	RBF	SNC
3	1,2,3	7	4,5,6,7,8,9,10	34.6	64.6	75.4
	1,3,5		1,2,3,4,5,6,7	40.7	57.9	73.6
	8,9,10		2,4,6,7,8,9,10	41.1	64.7	73.2
4	1,2,3,4	6	5,6,7,8,9,10	49.2	68.8	80
	1,3,5,7		1,2,3,4,5,6	55.5	69.2	81.7
	7,8,9,10		2,4,6,8,9,10	53.3	55.8	70.8
5	1,2,3,4,5	5	6,7,8,9,10	55.5	75	82
	1,3,5,7,9		2,4,6,8,10	50	74.5	88.5
	6,7,8,9,10		1,2,3,4,5	49	78	81
6	1,2,3,4,5,6	4	7,8,9,10	56.3	73.1	86.9
	6,7,8,9,10		1,2,3,4	54.4	69.4	88.1
	1,2,3,5,7,9		4,6,8,10	57.5	67.5	84.3
7	4,5,6,7,8,9,10	3	1,2,3	59.2	79.2	87.5
	1,2,3,4,5,6,7		8,9,10	61.7	70.8	85.8
	1,2,3,5,7,9,10		4,6,8	55.8	76.7	87.5
8	1,2,3,4,5,6,7,8	2	9,10	62.5	78.6	88.8
	3,4,5,6,7,8,9,10		1,2	62.5	70	86.3
	1,2,4,6,7,8,9,10		3,5	57.5	78.6	90

Table 3. Comparison of recognition rate for different approach on ORL database

Training Sample per class	Average recognition rate (%) with standard deviation		
	FFNN	RBF	SNC
2	23.3±6.3	61.5±4.2	58.0±4
3	37.3±3.3	64.1±3.7	74.1±3.6
4	47.8±7.1	65.2±6.9	78.1±4.4
5	47.8±6.0	69.1±5.7	83.2±3.4
6	53.4±5.2	68.5±2.9	87.1±1.9
7	56.8±3.5	72.6±5.0	88.2±2.2
8	59.3±3.3	73.2±5.8	88.5±1.6
9	62.5±2.5	73±2.1	92±4.1

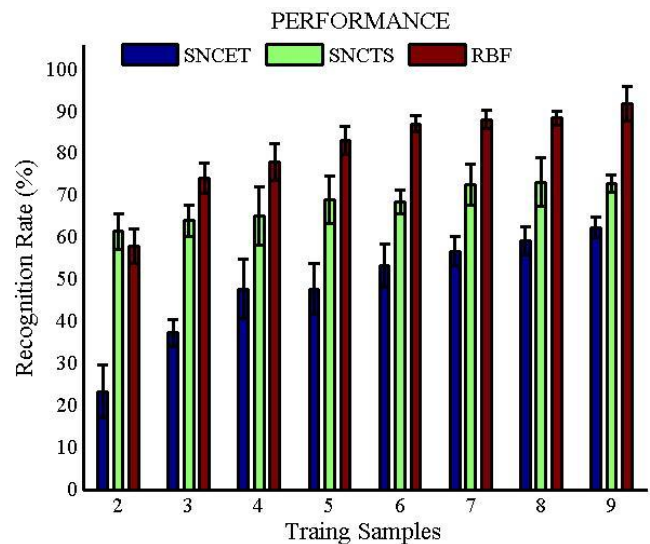


Figure 7. Comparison of proposed method with existing methods for ORL.

6. Conclusion

Neural architecture based FRS systems are investigated in this paper. The standard ORL data base is used to facilitate comparison with related works. The PCA based feature extraction is used. To determine the optimal number of PCA vectors an investigation is carried out. Based on the results it is concluded that 40 PCA vectors are suitable for ORL data base.

Three neural architectures FFNN, RBF and SNC are used to build the FRS. To identify the best possible learning algorithm the networks are trained using three different learning algorithms namely BPM, VLR and LM. For comparison, all the three NN models are trained

with the same forty feature vector, using same learning algorithms, for the same target MSE using 50% of images for training the rest of the 50% testing. For all the three architectures the LM algorithm converged faster and its recognition rate was superior to BPM and VLR trained architectures. Hence it is concluded that LM learning algorithm is most suitable for FRS. Using 40 PCA features as input and LM learning algorithm three FRS systems are built using FFNN, RBF and SNC architectures. The three FRS systems are extensively trained and tested using different combinations of training and testing images. With all possible permutations and combinations the average recognition rate is obtained along with the standard deviation. From the analysis it is inferred that recognition rate of SNC is superior as compared to FFNN and RBF.

The SNC-NN model resulted in structurally compact, computationally less complex model as compared to FFNN and RBF model. The potential of the SNC network has been identified through this investigation. The SNC network is also computationally less complex. It can be self-organized which greatly aids design automation. The SNC network performance is also shown to be superior to other results published in literature. The research investigations conclude that SNC neural network architecture is most suitable for FRS systems. It exhibits high recognition rate, are less complex, more compact and is easy to design. Hence SNC based system is a promising neural architecture for face recognition systems.

7. References

1. Sir Francis Galton. Personal identification and description-II. 1888 Jun 88. p. 201–3.
2. Bledsoe WW. The Model Method in Facial Recognition. Panoramic Research Inc Palo Alto. CA Rep. PRI: 1966; 15.
3. Bledsoe WW. Man-Machine Facial Recognition. Panoramic Research Inc Palo Alto.
4. CA Rep. PRI: 22; 1966.
5. Kirby M, Sirovich L. Application of the Karhunen-Loeve procedure for the characterization of human faces. *IEEE Trans. Patt Anal Mach Intell.* 1990; 12(1):103–7.
6. Sirovich L, Kirby M. Low-dimensional Procedure for the Characterization of Human Faces. *Journal of the Optical Society of America.* 1987 Mar; 4(3): 519–24.
7. Turk M, Pentland A. Face recognition using Eigen faces. *Proceedings of IEEE CVPR. Hawaii;* 1991 Jun. p. 586–91.
8. Martin Hagan T, Howard Demuth T, Mark Beale H. *Neural Network Design.* Cengage Learning (Thompson); 2008.
9. Lawrence S, Giles CL, Tsoi AC, Back AD. Face Recognition: A Convolutional Neural-Network Approach. *IEEE Trans. Neural Networks.* 1997 Jan; 8(1).
10. Musavi MT, Ahmed W, Chan KH, Faris KB, Hummels DM. On the training of radial basis classifier. *Neural Networks.* 1992; 5:595–603.
11. Eleyan A, Demirel H. Face Recognition System based on PCA and Feed Forward Neural Networks: In *Proc Lecture Notes in Computer Science.* 2005; 3512:935–42.
12. Hybrid Feature based Face Verification and Recognition System using Principal Component Analysis and Artificial Neural Network. Usha Mary Sharma. *Indian Journal of Science and Technology.* 2015 Jan; 8(S1):115–120.
13. Danti D, Danti A. Multiple Hierarchical Technique to Predict the Gender of a Person based on 3 Sigma Control Limits on Neural Network. *Indian Journal of Science and Technology.* 2015 Jul; 8(14): IPL 022.
14. Sujatha P, Sudha KK. Performance Analysis of Different Edge Detection Techniques for Image Segmentation. *Indian Journal of Science and Technology.* 2015 Jul; 8(14):IPL026.
15. A Hybrid Face Detection Approach in Colour Images with Complex Background Esmaeil Kheirkhah and Zahra Sadri Tabatabaie. *Indian Journal of Science and Technology.* 2015 Jan; 8(1): 49–60.
16. Improving Face Recognition Rate with Image Pre processing. Krishna Dharavath, Fazal Ahmed Talukdar and Rabul Hussain Laskar. *Indian Journal of Science and Technology.* 2014 Aug; 7(8):1170–75.
17. Olivetti and Oracle Research Laboratory. The Olivetti and Oracle Research Laboratory Face Database of Faces. Available from: <http://www.camorl.co.uk/facedatabase.html>