

Human Face Recognition Using A Hybrid Learning RBF Neural Network with Pseudo Zernike Moment Invariant

A.A. Yusuff¹, B. Osofisan², M. O. Okelola¹ & I. A. Adeyemo¹

¹Electronic and Electrical Engineering Department, Faculty of Engineering and Technology
Ladoke Akintola University of Technology, P.M.B. 4000, Ogbomoso Oyo-state Nigeria

²Electrical Engineering Department, University of Lagos, Akoka, Nigeria.

ABSTRACT

In this paper a new method for the recognition of human face in 2-Dimensional digital images using a new Hybrid Learning Algorithm (HLA) for Radial Basis function Neural Network as classifier and Pseudo Zernike moment invariant (PZMI) as a face feature is proposed.

INTRODUCTION

Human face recognition is still a challenging and difficult task for Computer system. The difficulties are due to large variations in facial appearance, orientation, and changes in environmental conditions. In the recent years there has been a growing interest in face recognition systems [1]-[2] due to the commercial application in areas such as film processing, personal identification, access control system, e.t.c.

A complete face recognition system normally includes two stages. The first stage is used in detecting the location of face and extraction of pertinent features from the localized image. The second stage involves classification of facial images based on the feature vector obtained in the first stage. Neural networks have been employed and compared to conventional classifiers for a number of classification problems. The results have shown that the accuracy of the neural network approaches equivalent to or slightly better than other methods.

Radial Basis Functions (RBF) neural networks are found to be very attractive for many engineering problems because (1) they are universal approximator, (2) they have compact topology and (3) their learning speed is very fast because of their locally tuned neurons [7]-[8]. RBF neural network form a unifying link between many different research fields such as function approximation, regularization, noisy, interpolation and pattern recognition. Therefore RBF neural networks serve as an excellent candidate for pattern applications and attempts have been carried out to make the learning process in this type of classification faster than normally required for the multiplayer feed forward neural network [9]. In this paper we have used RBF neural network classifier with a new Hybrid Learning Algorithm (HLA) for human face recognition. PZMI was used to generate the feature vector, which yielded the best results for human face recognition in comparison with other moments [10].

Section 2 presents RBF neural network structure. Section 3 introduces classifier design based on RBF neural network. Feature extraction is given in section 4 and section 5 describes the experimental results and conclusion is described in section 6

RBF NEURAL NETWORK STRUCTURE

An RBF neural network structure is shown in Fig.(1). It has architecture similar to a traditional three layer feed forward neural network. It has three different layers with feed forward architecture. The input layer of this network is a set of n units, which accept the elements of an n dimensional input feature vector. The input units are fully connected to the hidden layer with r hidden units. Connections between the input and hidden layers have unit weights and, as a result, do not have to be trained. The goal of the hidden layer is to cluster the data and reduce the dimensionality. In this structure hidden layer is named RBF units. The RBF units are fully connected to the output layer. The output layer supplies the response of neural network to the activation pattern applied to the input layer. The transformation from the input space to the RBF-unit space is nonlinear.

The RBF neural network is a class of neural networks, where the activation function of the hidden units is determined by the distance between the input vector and a prototype vector. The activation function of the RBF units is expressed as follow [8]

$$R_i(x) = R_i \left(\frac{\|x - c_i\|}{\sigma_i} \right), \quad i = 1, 2, \dots, r \quad (1)$$

where x is an n -dimensional input feature vector, c_i is a n -dimensional vector called the center of the RBF unit, σ_i is the width of RBF unit and r is the number of the RBF units. Typically the activation function of the RBF units is chosen as a Gaussian function with mean vector c_i and variance σ_i as follows:

$$R_i(x) = \exp\left(-\frac{\|x - c_i\|^2}{\sigma_i^2}\right) \quad (2)$$

Note that σ_i^2 represent the diagonal entries of covariance matrix of Gaussian function. The output units are linear and therefore the response of the j-th output unit for x is given as:

$$y_j(x) = b(j) + \sum_{i=1}^r R_i(x)w_2(i, j) \quad (3)$$

where $w_2(i, j)$ is the connection weight of the I-th RBF units to the j-th output node and $b(j)$ is the bias of the j-th output. The bias is omitted in the network to reduce complexity.

$$y_j(x) = \sum_{i=1}^r R_i(x)w_2(i, j) \quad (4)$$

RBF BASED CLASSIFIER DESIGN

RBF neural network classifier can be viewed as a function mapping interpolant that tries to construct hypersurfaces for each class by taking a linear combination of the RBF units. These hypersurfaces can be viewed as discriminant functions, where the surface has a high value for the class it represents and low value for all others. An unknown input feature vector is classified as belonging to class associated with the hypersurface with the largest output at that point. In this case the RBF units serve as component in a finite expansion of the desired hypersurface where the component coefficients (the weights) have to be trained [11]

In designing a classifier based on RBF neural network, we have set the number of input nodes in the input layer of neural network equal to the number of feature vector elements. The number of nodes in the output layer is set to the number of image classes. The RBF units are selected using the following steps:

Step1: Set RBF units number to the number of outputs.

Step 2: For each class k , ($k=1,2,\dots,s$) the center of RBF units is selected as the mean value of the sample feature belonging to the same class, i.e

$$C^k = \frac{\sum_{i=1}^{N^k} p^k(n, i)}{N^k} \quad (5)$$

where $p^k(n, i)$ is the i-th sample with n as the number of features, belonging to the class k , and N^k is the number of images in the class.

Step3: For any class k , compute the distance d_k from the mean C^k to the furthest point p_j^k belonging to class k :

$$d_k = \left| p_j^k - C_k \right| \quad (6)$$

Step4: For any class k , compute the distance $dc(k, j)$ between the mean of class k and the mean of other classes as follows:

$$dc(k, j) = \left| C^k - C^l \right| \quad (7)$$

where $j = 1, 2, \dots, s$ and $j \neq k$

Then find $d_{\min}(k, l) = \min(dc(k, j))$ and check the relationship between $d_{\min}(k, l)$, d_k , and d_l . If

$d_k + d_l \leq d_{\min}(k, l)$ then the class k is not overlapping with other classes. Otherwise the class k is overlapping with other classes and misclassifications may occur in this case.

Step5: For all the training data within the same class, check how they are classified.

Step6: The mean values of the classes are selected as the centers of RBF units.

A. Hybrid Learning Algorithm

Training RBF neural network can be made faster than the methods used to train multi-layer neural networks.

This is based on the properties of the RBF units, and lead to a two stage training procedure. The first stage of training involves the determination of the output connection weights, which requires the solution of a linear problem, which is fast. In the second stage, the parameters governing the basis function (corresponding to the RBF units) are determined using unsupervised method that is also relatively fast.

Training of the RBF neural network involves estimating output connection weights, centers and widths of RBF units. Dimensionality of the input vector and the number of classes set the number of input and output units respectively. A Hybrid Learning Algorithm (HLA), which combines the gradient method and the Linear Least Squared (LLS) method is proposed for training neural network in this paper. This is done in two steps. In the first step, the neural network connection weights in the output of the RBF units $w_2(i, j)$ are adjusted that is shown in Fig. (1) with this assuming that the centers and widths of the RBF units are known. In the second

step, the centers and widths (c and σ) of the RBF units are estimated.

1) *Computing Connection Weights*: Let r and s be the number of inputs and outputs respectively, and we assume that the number of u RBF outputs are generated for all training face patterns. For any input

$P_i(p_{1i}, p_{2i}, \dots, p_{ri})$, the j-th output y_j of the RBF neural network in equation (4) can be calculated in more compact form as follows:

$$W_2 \times R = Y \quad (8)$$

Our objective is the followings:

Given $R \in \mathfrak{R}^{u \times N}$ that contains c and σ of RBF units and

$T = (t_1, t_2, \dots, t_s)^T \in \mathfrak{R}^{s \times N}$, where N is the total number of sample face patterns and T is the target matrix consisting of 1's and 0's with each column having only one nonzero element. This matrix

identifies the processing pattern to which the given exemplar belongs. Furthermore, given the following relationship for error:

$$E = \|T - Y\| \quad (9)$$

We find an optimal coefficient matrix

$W'_2 \in \mathfrak{R}^{s \times N}$ such that the $E^T E$ to be minimized. This is done by the well-known LLS method [7] as follows:

$$W'_2 \times R = Y \quad (10)$$

The optimal W'_2 is given by:

$$W'_2 = T \times R^+ \quad (11)$$

Where R^+ is the pseudo inverse of R and is given by:

$$R^+ = (R^T R)^{-1} R^T \quad (12)$$

We can compute the connection weights using equation (11) and (12) by knowing R matrix as follows:

$$W'_2 = T(R^T R)^{-1} R^T \quad (13)$$

2) *Defining Center and Width of RBF Units:* Here, the center and width of RBF units (R matrix elements) are adjusted by taking the negative gradient of the error function, E^n , which is given by [7]:

$$E^n = \frac{1}{2} \sum_{k=1}^s (t_k^n - y_k^n)^2, \quad n = 1, 2, \dots, N \quad (14)$$

Where t_k^n and y_k^n represent the k-th target output and real output of the n-th face pattern respectively. For the RBF units with the center C and the width σ , the error rate of the center $\Delta C^n(i, j)$ can be derived from equation (14) by the chain rule as follows:

$$\Delta C^n(i, j) = -\xi \frac{\partial E^n}{\partial C^n(i, j)} \quad (15)$$

$$\Delta C^n(i, j) = 2\xi \sum_{k=1}^s y_k^n \cdot w_k^n(k, j) \cdot R_j^n \cdot (P_i^n - C^n(i, j)) / (\sigma_j^n)^2$$

and the error rate of the width $\Delta \sigma_j^n$ are computed as follow:

$$\Delta \sigma_j^n = -\xi \frac{\partial E^n}{\partial \sigma_j^n} \quad (16)$$

$$\Delta \sigma_j^n = 2\xi \sum_{k=1}^s y_k^n \cdot w_k^n(k, j) \cdot R_j^n \cdot (P_i^n - C^n(i, j)) / (\sigma_j^n)^3$$

Where $i=1, 2, \dots, r$ and $j=1, 2, \dots, u$ and also P_i^n is the i-th input variable of the n-th face pattern and ξ is the learning rate. $\Delta C^n(i, j)$ is the error rate of the center of the i-th input variable of the j-th prototype of the n-th training pattern. $\Delta \sigma_j^n$ is the error rate of the width of the j-th prototype at the face pattern. Now we can combine the linear least square and the gradient method to update the parameters in RBF neural network. Each epoch of this hybrid learning

algorithm is composed of forward pass and a backward pass. In the

forward pass, we calculate the W'_2 matrix. In the backward pass, the RBF parameters are obtained.

FEATURE EXTRACTION

Statistical-based approaches for feature extraction such as moment invariants [12] have received considerable attention in recent years for their invariance property. The term invariant denotes an image feature remains unchanged if that image undergoes one or a combination of the changes such as: change of size (scale), change of position (translation), change of orientation (rotation), and reflection. The invariant properties of moments are utilized as pattern sensitive features in classification and recognition applications [13].

A. Pseudo Zernike Moment Invariant

Pseudo Zernike polynomials are well known and widely used in the analysis of optical systems. Pseudo Zernike polynomials are orthogonal set of complex valued polynomials defined as [12]:

$$v_{nm}(x, y) = R_{nm}(x, y) \exp(jm \tan^{-1}(\frac{y}{x})) \quad (17)$$

where $x^2 + y^2 \leq 1, n \geq 0, |m| \leq n$ is even and

Radial

polynomials R_{nm} are defined as:

$$R_{nm}(x, y) = \sum_{s=0}^{n-|m|} D_{n, |m|, s} (x^2 + y^2)^{\frac{n-s}{2}} \quad (18)$$

where:

$$D_{n, |m|, s} = (-1)^s \frac{(2n+1-s)!}{s!(n-|m|-s)!(n-|m|-s+1)!} \quad (19)$$

The PZMI can be computed using the scale invariant central moments and the radial geometric moments that is defined in reference [12] as follows:

$$PZMI_{min}^n = \frac{m!}{\pi} \sum_{(n-m-s) \text{ even}}^{n-|m|} D_{n, |m|, s} \sum_{i=0}^k \sum_{b=0}^m \binom{k}{a} \binom{m}{b} (-j)^b S_{n, |m|, s} CM_{k+m-2i-b, 2i+b}^{n-|m|} + \frac{m!}{\pi} \sum_{(n-m-s) \text{ odd}}^{n-|m|} D_{n, |m|, s} \sum_{a=0}^d \sum_{b=0}^m \binom{d}{a} \binom{m}{b} (-j)^b S_{n, |m|, s} RM_{2i+m-2a-b, 2i+b}^{n-|m|} \quad (20)$$

Where

$$k = (n - s - m) / 2, \quad d = (n - s - m + 1) / 2, \quad S_{n, |m|, s}, \quad CM_{i, j}$$

is the Central moments and j, i RM is the Radial moments are as follow:

$$S_{n, |m|, s} = (-1)^s \frac{(n-s)!}{s! \left(\frac{n+|m|}{2} - s\right)! \left(\frac{n-|m|}{2} - s\right)!} \quad (21)$$

$$CM_{pq} = \frac{\mu_{pq}}{M_{00}^{(p+q+2)/2}} \quad (22)$$

$$RM_{pq} = \frac{\sum_x \sum_y f(x, y) ((\hat{x}^2 + \hat{y}^2)^{p/2} \hat{x}^q \hat{y}^q)}{M_{00}^{(p+q+2)/2}} \quad (23)$$

$\hat{x} = x - x_0, \hat{y} = y - y_0$ and x_0, y_0, μ_{pq} and M_{pq} are defined as follows

$$M_{pq} = \sum_x \sum_y f(x,y) x^p y^q \quad (24)$$

$$\mu_{pq} = \sum_x \sum_y f(x,y) \hat{x}^p \hat{y}^q \quad (25)$$

$$x_0 = \frac{M_{10}}{M_{00}} \quad (26)$$

$$y_0 = \frac{M_{01}}{M_{00}} \quad (27)$$

B. Feature Vector Creation

The feature extraction is done in two steps. In the first step, considering the elliptical shape of a face in general, it is convenient to search for connected component of region using a region-growing algorithm and fit an ellipse to every connected component of nearly elliptical shape [3]. After that we create a subimage, which contains information needed for recognition algorithm. The subimage encloses the pertinent information around the face in the best-fit ellipse while pixel value outside the ellipse is set to zero. This is a basic preprocessing step, because it disregards irrelevant data to facial portion such as hair, neck, shoulders and background and therefore the speed of computing PZMI is increased due to smaller pixels content of the subimages [3]. Figure 2 shows sample of selecting of face location and creating of subimage in feature extracting respectively.



Face location Subimage
Figure 2: Creating subimages from face images

In the second step, the feature vector is obtained through calculating of the PZMI of the derived subimage. For selecting PZMI as face feature we define four categories of feature vectors based on



the order (n) of the PZMI. In the first category that is n=1,2,..6, all moments of PZMI are considered as feature vectors elements. The number of the feature vector elements in this category is

26. In the second category, n=4,5,6,7 is chosen. All moments of each order included in this category are summed up to create a feature vectors of size 26. In the third category, n=6,7,8 is considered. The feature vectors for this category has 24 element. Finally in the last category that n=9,10 is considered with 21 feature element. These elements are shown in table 1. Our experimental study indicates that this categorizing process allows the feature extractor to have a lower dimensional vector while maintaining a good discrimination capability. Also the lower dimensionality of the feature vector lowers computational burden of the recognition algorithm [10].

TABLE 1: FEATURE VECTORS ELEMENTS BASED ON PZMI

Category No	PZMI feature elements
1	n=1 m=0,1 n=2 m=0,1,2 n=3 m=0,1,2,3 n=4 m=0,1,2,3,4 n=5 m=0,1,2,3,4,5 n=6 m=0,1,2,3,4,5,6
2	n=4 m=0,1,2,3,4 n=5 m=0,1,2,3,4,5 n=6 m=0,1,2,3,4,5,6 n=7 m=0,1,2,3,4,5,6,7
3	n=6 m=0,1,2,3,4,5,6 n=7 m=0,1,2,3,4,5,6,7 n=8 m=0,1,2,3,4,5,6,7,8
4	n=9 m=0,1,2,3,4,5,6,7,8,9 n=10 m=0,1,2,3,4,5,6,7,8,9,10

EXPERIMENTAL RESULTS

Experimental studies are carried out on the database images of Final year students of Electronic and Electrical Engineering 2003/2004 set. 500 face images from 50 individuals in different states from the database have been used to evaluate the performance of the proposed method. None of the 10 samples are identical to each other. They vary in position, rotation, scale and expression. In this database each person has changed his face expression in each of 10 samples (open/close eye, smiling/not smiling). For some individuals, the images were taken at different times, varying facial details (glasses/no glasses). Samples of database used are shown in Fig. (3).



Figure 3: Sample of face database

A total of 250 images were used to train and another 250 were used to test. Each training set consists of 5 randomly chosen images from the same class in the training stage. Simulation has been done in four steps based on the order (n) of the PZMI as described in section III.B.

Number of elements in the feature vectors in each step depends on the number of moments in each category as was shown in table 1. Neural network classifier is trained in each category based on training images. The outcome of experimental results is shown in the table 2.

TABLE 2: EXPERIMENTAL RESULTS

Features vectors		Training Phase		Testing Phase	
Category	NFE ¹	No. of Epochs	RMSE	NM ²	ER ³
n=1,2,...,6	27	80 ~ 100	0.09~0.06	15	7.5%
n=4,5,6,7	26	60 ~ 80	0.06 ~ 0.04	9	4.5%
n=6,7,8	24	45 ~ 60	0.04 ~ 0.02	6	3%
n=9,10	21	30 ~ 45	0.04 ~ 0.01	3	1.3%

1: Number of Feature Element

2: Number of Misclassification

3: Error Rate = NM/ Number of total testing images

This table indicates that in the training phase of the RBF neural network classifier, number of epochs decrease when the moments orders increase. On the other hand, training phase in RBF neural network with HLA learning method has converged faster when higher orders of PZMI are used in comparison with the lower orders of PZMI. Also these tables indicate that HLA method in training phase has a lower error (RMSE) with good capability to discriminate classes. Table (2) shows the recognition rate has increased in spite of the decrease in the number of feature elements of the feature vector. This observation is interesting since the improvement in recognition rate is obtained while the number of moment calculation is decreased. To compare the HLA learning algorithm with other learning algorithms, we develop the k-mean clustering algorithm for training RBF neural network that very popular method for training RBF neural network [14].

We have applied it to our database with same feature extraction technique. Table 3 shows the outcome of this comparative study.

TABLE 3: COMPARISON BETWEEN TWO METHOD

Feature category	K-mean clustering		HLA method	
	No. of Epochs	RMSE	No. of Epochs	RMSE
1	135 ~ 120	0.12~0.09	80 ~ 100	0.09~0.06
2	120 ~ 95	0.09 ~ 0.06	60 ~ 80	0.06 ~ 0.04
3	95 ~ 70	0.06 ~ 0.04	45 ~ 60	0.04 ~ 0.02
4	70 ~ 50	0.04 ~ 0.02	30 ~ 45	0.02 ~ 0.01

As table 3 indicates the HLA method has converged faster than the k-mean clustering. Also RMSE during the training phase for HLA method is smaller than k-mean clustering learning algorithm. Also to compare the effectiveness of our proposed method in comparison with k-mean clustering in overall human face recognition system, we have chosen the PZMI of orders 9 and 10 for feature extraction, which yielded the best result in our study. The feature vector in this case has 21 elements. Also

we consider RBF neural network as classifier with HLA and k-mean clustering techniques as learning algorithm. In this study an average error rate is defined as:

$$E_{av} = \frac{\sum_{i=0}^m E_{NM}(i)}{m \cdot N_i} \quad (28)$$

Where m is the number of experimental runs, each being performed on random partition of the database into sets, $E_{NM}(i)$ is the number of misclassification for the i th run, and N_i is the number of total testing images for each runs. Table 4 shows the result of this comparative.

TABLE 4: ERROR RATE IN DIFFERENT METHOD

Learning Methods	No. of Experimental (m)	$E_{av}\%$
k-mean clustering [14]	6	1.761
HLA	6	1.311

CONCLUSION

This paper presents a novel technique for the RBF neural network learning algorithm in recognition of human facial images. This algorithm improves convergence of the learning phase of the RBF neural network classifier. The algorithm also substantially decreases the dimensions of the search space in the gradient method, this is crucial in optimization of high-dimensional problem such as human face recognition. A comparative study with k-mean clustering learning techniques demonstrates the superiority of the proposed algorithm. We have shown in this paper, the relationship between the order of the PZMI and convergence speed of RBF neural network with HLA technique. It is shown that, higher orders PZMI contain more information about facial image and hence improves the recognition rate the speed of convergence in learning phase is increased.

REFERENCES

- M. A. Grudin, "On Internal Representation in face Recognition Systems", *Pattern Recognition*, Vol. 33:7, pp. 1161-1177, 2000
- J. Daugman, "Face Detection: A Survey", *Computer Vision and Image Understanding*, Vol. 83:3, pp. 236-274, September 2001
- J. Haddadnia, K. faez, "Human Face Recognition Based on Shape Information and Pseudo Zernike Moment", 5th Int. Fall Workshop Vision, Modeling and Visualization, Saarbrucken, Germany, pp. 113-118, Nov. 22-24, 2000
- K. Sobotta and I. Pitas, "Face Localization and Facial Feature Extraction Based On Shape and Color Information", *International Conference on Image Processing*. Vol. 3, pp. 483-486, 1996
- J. Wang and T. Tan, "A New Face Detection Method Based on Shape Information", *Pattern Recognition Letter*, Vol. 21, pp. 463-471, 2000
- W. Zhou, "Verification of the nonparametric characteristics of backpropagation neural networks for image classification", *IEEE Transaction On Geoscience and Remote Sensing*, Vol. 37:2, pp. 771-779, March 1999
- J-S. R. Jang, "ANFIS: Adaptive-Network-Based Fuzzy Inference System", *IEEE Trans. Syst. Man. Cybern.*, Vol. 23:3, pp. 665-684, 1993
- J. Haddadnia, K. Faez, "Human Face Recognition Using Radial Basis Function Neural Network", 3rd Int. Conf. On Human and Computer, Aizu, Japan, pp. 137-142, Sep. 6-9, 2000
- C. M. Bishop, *Neural Network for Pattern Recognition*, Oxford University Press, New York, U.S.A., 1995
- J. Haddadnia, K. Faez, "Human Face Recognition with Moments Invariant", *The IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing*, Baltimore, Maryland, USA, June 3-6, 2001
- "MIT Tech. Rep. 894", Lincoln Labs., 1991 C. H. The and R. T. Chin, "On Image Analysis by the Methods of Moments", *IEEE Trans. On Pattern Analysis And Machine Intelligence*, Vol. 10:4, pp. 496-513, 1988.
- S.O. Belkasim, M. Shridhar and M. Ahmadi, "Pattern Recognition with Moment Invariants: A Comparative Study and New Results", *IEEE Trans. On Pattern Recognition*, Vol. 24:12, pp. 1117-1138, 1991
- S. Gutta, J. R. J. Huang, P. Jonathon and H. Wechsler, "Mixture of Experts for Classification of Gender, Ethnic Origin, and Pose of Human Faces", *IEEE Trans. On Neural Network*, Vol. 11:4, pp. 949-960, July 2000
- J. Haddadnia, M. Ahmadi and K. Faez, "A Hybrid Learning RBF Neural Network For Human Face Recognition With Pseudo Zernike Moment Invariant", *Proceedings of the International Joint Conference on Neural Networks*, Honolulu, Hawaii, May 12-17 2002