International Journal of Soft Computing 3 (3): 248-253, 2008
ISSN: 1816-9503

BImplementation of High Speed Face Recognition Based on Karhunen Loeve Transform and Fisher's Discriminant, Radial Basis Function of Echo State Neural Network

[1]Srinivasa Rao Madane, [2]Wahidha Banu and [3]Purushothaman Srinivasan, S. Chandra Rao Madane
[1]Vinayaka Missions University, Salem, India
[2]Government of College of Engineering, Salem, India
[3]MIT, Kundapura, Karnataka State, India

Abstract: Recently proposed approach to recognitize facial expressions have been proposed Jager with the so called Echo State Neural Network (ESNN). The ESSN approach assumes a sort of "block box" operability of the network and clients a broad applicability to several different problems using same principle, here we proposes a simplified version of ESNN which we call a simple echo state network which exhibits good results in memory capacity and facial matching and recognition which allows a better understating of the capability and restriction of ESNN. ESSN gives promising result when the input are distorted. Simulation results show that a proposed system (ESNN) achieves a excellent performance with high training and recognition speed.

Key words: ESNN, RBF, RNN, ligh speed, face recognition

INTRODUCTION

Face recognition has been recently the most favored subject due to its most non-intrusive nature and wide range of applications. Its applications are in crime investigation, security, etc. But the whole process depends on the images, so image quality plays an important role and is influenced by some fundamental characteristics like lighting, camara position and emotional expressions.

Many Face Detection Researchers have used the idea that facial images can be chracteristized directly in terms of pixel intensities. these images can be characteritized by probabilistic models of the set of face images (Antonio and Thomas, 1997; Baback and Alex, 1995; Alex *et al*., 1994) or implicitly by neural network of other mechanisms (Pawan, 1994; Kah-kay, 1996; Vaillant *et al*., 1994; Gaungzheng and Thomas, 1994; Kin and Roberto, 1996; Gilles and Dominique, 1994; Lin *et al*., 1997; Meng *et al*., 2005).

Training a neural network for the face detection task is challenging because of the difficulties in characteritizing prototypical non face images. Unlike face recogintion, in which the classes to be discriminated are different faces, the two classes to be discriminated inn face recognition are images containing faces and images not containing faces.

Generally speaking, research on face recognition can be grouped into two categories, namely, feature based and holistic (also called template matching) approaches (Chellapa *et al*., 1995; Brunelli and Poggio, 1993) are based on the shapes and geometrical relationships of individual facial features including eyes, mouth, nose and chin on the other hand, holistic approaches handle the input face images globally and extract important facial features based on the high dimensional intensity values of face images automatically. Although, feature based schemes are more robust against rotation, scale and illumination variations, they greatly rely on the accuracy of facial feature detection methods (Meng *et al*., 2005) and it has been argued that existing feature based techniques are not reliable enough for extracting individual facial features. Holistic face recognition has attached more attention since the well known statistical method, the principal component analysis was applied in face recognition (Kirupy and Sirvoich, 1990; Turk and Pentaland, 1997). Another well-Known approaches is the Fisher faces in which the Fisher's linear discriminate is employed after the PCA is used for dimensionality reduction (Belhumeur *et al*., 1997), compared with the Eigenface, the fisher face approach is more sensitive and statistical method for feature extraction, the choice of training samples will affect its performance. In Martinez and Kak (2001), the author indicate that the FLD works efficiently only the number of training samples is large and facial expression and variation in lighting direction and facial expression. More recently, some variations of FLD have been developed for face recognition such as F-LDA

Corresponding Author: Professor Srinivasa Rao Madane, Vinayaka Missions University, Salem, India

(Lotlikar and Kothari, 2000), D-LDA (Yu and Yang, 2001), FD-LDA (Lu *et al.*, 2003) and KDDA (Lu *et al.*, 2003), etc.. However, the computation requirements of these approaches are greatly related to dimensionality of the original data and number of training samples. The Echostate neural network has been employed in face recognition.

Problem definition: The problem is to find out a better method to identify a person even with distorted information of the face, under poor lighting, under different unusual posture. Different methods have been evolved during the past research work that include the varieties of intelligent methods. The proposed work involves estimation of the person's identity from the given facial photograph using ESNN. The network learns the facial feature obtained from fishers linear discriminant plane. The work involves in comparison of the performance of the Radial Basis Function (RBF) with ESNN.

Radial basis function: A function is a Radial Basis Function (RBF) if its output depends on the distance of the input sample (vector) from another stored vector, referred to as the center for the RBF. An RBF Network (RBFN) is a feedforward neural network with one hidden layers, with an RBF node function at each hidden node; the weight vector from input layer to a hidden node is identical to the location of the center of the RBF for that node.

RBFN development algorithms use the following steps:

- The training set is "compressed" into a smaller set of RBF node centers ($\mu_1, \mu_2, \ldots\ldots, \mu_N$) and associated values ($\sigma_1, \ldots\ldots \sigma_N$) describing the range of applicability of each node.
- The function values ($f(\mu_1), \ldots\ldots, f(\mu_N)$) at all these centers are estimated from the training set.
- For any x, the value of f(x) is estimated by computing a weighted average of ($f(\mu_1), \ldots\ldots, f(\mu_N)$), where each $f(\mu_K)$ is weighted by a quantity proportional to $\rho_k|x-\mu_k|$, the corresponding value of the RBF. In other words, the output of the RBFN is

$$\sum_{K=1}^{N} f(\mu_k)\ \varphi k\ \varphi k\ (|x-\mu_k|)$$

Where φ_k denotes the connection weight from the kth hidden (RBF) node to the output (Summation) node.

ECHO STATE NEURAL NETWORK (ESNN)

Echo state neural networks are special recurrent neural networks, with the following properties.

- A large, sparsely connected, RNN is used as a "reservoir" of dynamics (recurrent interconnected perceptions).
- This dynamic reservoir can be excited by inputs and/or feedback of the outputs.
- Connection weights of the reservoir are not changed by training.
- Only weights from the reservoir to the output units are adapted, so training becomes linear regression task.

We consider discrete time neural network with K input units internal network units and L output units (Fig. 1). Activations of input units at time step n are $u(n) = (u_1\ (n), \ldots, u_k\ (n))$, of internal units are $x(n) = (x_1\ (n), \ldots\ldots, x_N\ (n))$ and of output units $y(n) = (y_1\ (n), \ldots, y_L\ (n))$ real-valued connection weights are collected in a N×K weight matrix $W^{in} = (w^{in}_{ij})$ for the input weights, in an N×N matrix $W = (w_{ij})$ for the internal connections, in an L×(K+N+L) matrix $Wout = (w_{ij}^{out})$ for the connections to the output units and in a NxL Matrix $W^{back} = (w_{ij}^{back})$ for the connections that project back from the output to the internal units. Note that connections directly from the input to the output units and connections between output units are allowed. No further conditions on the network topology induced by the internal weights W are imposed. We will also not formally require, but generally intend that the internal connections W induce recurrent pathways between internal units. Without further mention, we will always assume real-valued inputs, weights and activations.

The activation of internal units is updated according to $x(n+1) = f\,(W^{in}\,u\,(n+1) + Wx\,(n) + W^{back}\,y\,(n))$,

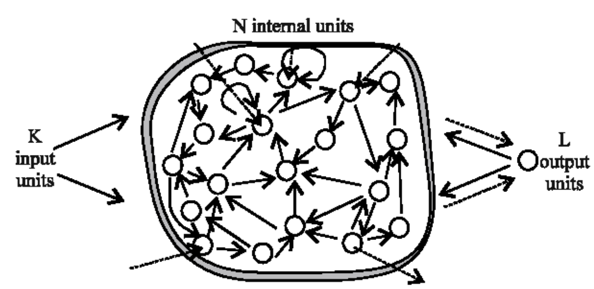

Fig. 1: Basic network architecture

Where f = (f1,…, fN) are the internal units output functions. The output is computed according to $y(n+1) = f^{out}(W^{out}(u(n+1), x(n+1), y(n))$, where $f^{out} = f_1^{out},\ldots, f_L^{out})$ are the output units output functions and $(u(n+1), x(n+1), y(n))$ is the concatenation of the input, internal and previous activation vectors.

SYSTEM DESIGN FOR FACIAL RECOGNITION USING ESNN

In this research, much concentration is done for the best recognition of face by implementing an ESNN. Figure 2 illustrates the sequence of steps involved in intelligent recognition of facial expression.

Implementation
Training:

- Decide number of persons.
- Take three facial expression of each person.
- Calculate the Principal Component Vector by

$$Z = Z * Z^T$$

Where
Z = Intensities of image.

- Find Eigen Vector of the Z matrix..
- Calculate the Phi_ And Phi_2 Vectors as follows.

For Discriminating various persons.

$$Phi\_1 = eigenvector(S_b * S_w^{-1})$$

$$S_b = \Sigma (PCV_i - M_0) (PCV_i - M_0)^T / N$$

Where,
PCV_i (i = 1,2,3,…n)
PCV_1 = Principal Component Vectors of Person each person.
M_0 = Average of ($PCV_1 + PCV_2 + PCV_3$).
S_w = $\Sigma (PCV_i - M_i) / N (PCV_i - M_i)^T$.

Where
M_i = 1 , 2 , 3,….n
M_i = Average of PCV_i

- Calculate Phi_2 Vector.
  $Phi\_2 = eigenvector (Q\, S_b S_w^{-1})$
  $Q = I - ((Phi\_1 * Phi\_1^{-1} * S_w^{-1}) / (Phi\_1^t * S_w^{-1} * Phi\_1))$

- Transfer for M,N Dimensional Vector into Two Dimensional Vector.

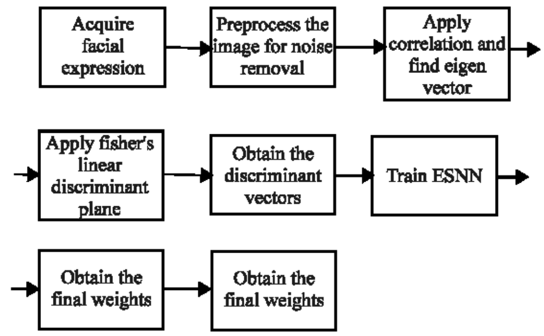

Fig. 2: Schematic diagram of the ESNN facial recognition

$U = Phi\_1 * PCV_{i\,(1,2,3,\ldots n)}$
$V = Phi\_2 * PCV_{i\,(1,2,3,\ldots n)}$

- Apply Echostate Neural network as follows.

Testing:
- Read test Image.
- Calculate the Principal Component Vector b.

$$Z = Z * Z^T$$

Where Z = Intensities of image

- Find Eigen Vector of the Z matrix by applying Eigen process.
- Connect to the Database.
- Update the table respective to the Person's image.
- Display the Person's Name which is get updated in the Database.

Echostate training of facial features: Decide the input features of the registered image
Fix the target values
Set no. of inputs=2;
Set no of reservoir = 20;
Set no. of output = 1
Create weight matrix(no of reservoirs,no.of inputs)= random numbers -0.5
Create weight backup matrix(no.of outputs, no of reservoirs) = (random numbers -0.5)/2
Create weight not (w0)(no.of reservoirs, no of reservoirs)= (random numbers -0.5)
Create temp matrix (te)(no.of reservoirs, no of reservoirs)= random numbers
Calculate w0 = w0.* (te <0.3)
Calculate w0 = w0.* (w0 <0.3)
Follow the heuristics
v = eig(w0)

```
lamda = max(abs(v))
w1= w0/lamda
w = .9 * w1
Create network training dynamics
state = zeros(no_reservoir,1)
desired = 0;
for loop
      input = x(i:i+nipp-1)
      F = wt_input* input'
      TT = w*state
      TH = wt_back' * desired
      next_state = tanh( F+TT + TH)
      state = next_state
      desired = x(i+nipp-1)
      desired_1 = desired
end
```

ESNN network testing:

```
input   =   x(i:i+nipp-1);
   F      =   wt_input* input';
   TTH  =   wt_back' * output_d;
next_state = tanh(F + w*state + TTH);
state = next_state;
output(i) = (wout' *state);
```

RESULTS AND DISCUSSION

Different orientations of persons have been considered to implement the ESNN for the facial recognition. The various postures have been shown in Fig. 3.

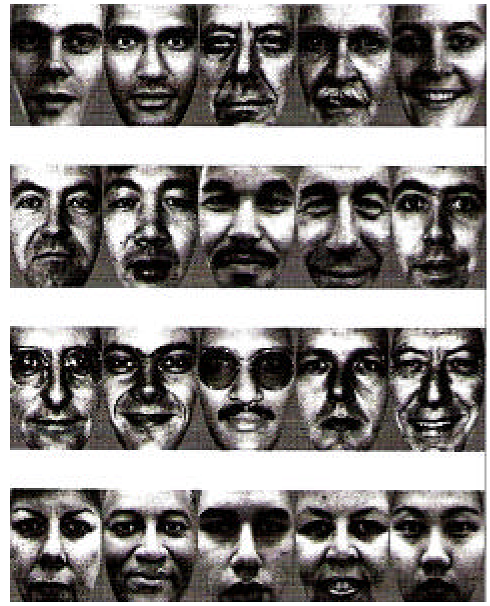

Fig. 3: Facial expressions of different persons

Sample image pixel matrix

147	148	153	155	152	154	159	161	163	163
149	149	156	158	158	163	162	162	64	164
151	152	157	157	161	161	163	165	166	169
155	157	159	159	160	163	164	167	167	168
153	158	161	161	162	162	164	168	170	170
156	160	163	162	164	165	166	167	172	170
158	162	161	162	166	166	168	170	172	170
160	161	164	164	168	165	167	171	175	172
159	161	164	167	170	168	170	175	175	179
161	162	165	167	167	168	174	173	171	174

Correlation Matrix

3078745	2773595	2555539	2367792	2258210	2139709	2113195	2103348	2036330	193086
2773595	2682993	2500483	2331402	2232712	2114552	2083707	2048782	1990069	188762
2555539	2500483	2442523	2311269	2216606	2100021	2064694	2019620	1974854	188231
2367792	2331402	2311269	2272284	2189300	2074773	2037447	1998668	1960587	187250
2258210	2232712	2216606	2189300	2165827	2066828	2026552	1992738	1956568	187226
2139709	2114552	2100021	2074773	2066828	2046545	2015895	1982751	1949807	186665
2113195	2083707	2064694	2037447	2026552	2015895	2011248	1985045	1953431	186956
2103348	2048782	2019620	1998668	1992738	1982751	1985045	1990493	1955644	187084
2036330	1990069	1974854	1960587	1956568	1949807	1953431	1955644	1953017	187617
1930868	1887621	1882311	1872507	1872265	1866657	1869566	1870840	1876170	185118

Eigenvector Matrix

0.0831	0.0580	-0.1477	-0.0771	-0.1397	-0.0354	0.0134	-0.0301	0.0765	0.0845
0.0735	0.0447	-0.0103	0.1179	0.0601	0.1065	0.0247	0.1457	0.0718	0.0403
-0.1618	0.0294	-0.0263	0.0555	0.0842	-0.0805	0.1065	-0.1532	0.0446	0.0845
0.1353	0.1675	0.1356	0.1062	-0.1016	-0.0707	0.1735	0.0843	0.1651	0.0349
-0.0148	0.0642	-0.0026	-0.1232	0.0386	0.0328	-0.3047	0.3148	-0.2422	-0.0870
-0.0318	0.0194	-0.1203	-0.0771	-0.0975	-0.0215	0.0115	-0.1866	-0.0387	-0.1402
-0.0292	-0.1180	0.2274	-0.0882	-0.0971	0.2704	-0.0232	-0.1394	0.1302	0.0491
0.0538	-0.1386	-0.1496	0.0230	-0.0165	-0.1151	-0.0613	0.1083	-0.1367	0.1269
-0.0501	0.1916	0.1572	-0.0139	0.1230	0.0571	0.0267	-0.0678	-0.1290	-0.29 60
0.0660	-0.0328	0.0466	0.0596	0.1765	0.0078	-0.0520	-0.1454	0.0269	0.2466

Eigenvalue Matrix

0.4407	0	0	0	0	0	0	0	0	0
0	0.8801	0	0	0	0	0	0	0	0
0	0	2.2042	0	0	0	0	0	0	0
0	0	0	3.7260	0	0	0	0	0	0
0	0	0	0	5.9275	0	0	0	0	0
0	0	0	0	0	9.2362	0	0	0	0
0	0	0	0	0	0	10.2134	0	0	0
0	0	0	0	0	0	0	12.6410	0	0
0	0	0	0	0	0	0	0	20.4903	0
0	0	0	0	0	0	0	0	0	30.3254

Diagonal Values of Eigenvalue Matrix

Columns 1 through 12

0.4407	0.8801	2.2042	3.7260	5.9275	9.2362	10.2134	12.6412	20.4903	30.3254	34.6926

Columns 13 through 20

5205256	606881	779464	97.0914	129.1984	141.5460	156.3125	171.4565

Sorted Value

Columns 1 through 12

0.4407	0.8801	2.2042	3.7260	5.9275	9.2362	10.2134	12.6412	20.4903	30.3254	34.6926

Columns 13 through 20

52.5256	60.6881	77.9464	97.0914	129.1984	141.5460	156.3125	171.4565

Column Chosen100 Eigenvector chosen

Columns 1 through 12

0.1138	0.1036	0.0978	0.0927	0.0903	0.0882	0.0880	0.0884	0.0870	0.0842	0.0832

Columns 13 through 20

0.0822	0.0840	0.0842	0.0872	0.0890	0.0913	0.0916	0.0929

Projection Vector phi_1 and phi_2

K>> phi_1(1:30)'

ans =

Columns 1 through 12

0.0964	-0.0742	-0.0649	-0.0524	-0.0474	-0.0380	-0.0282	-0.0234	-0.0123	-0.0071	-0.0024

Columns 13 through 24

0.00091	0.0154	0.0223	0.0229	0.0322	0.0397	0.0396	0.0406	0.0375	0.0387	0.0501

Columns 25 through 30

0.0590	0.0654	0.0657	0.0685	0.0562	0.0568

K>> phi_2(1:30)

ans =

Columns 1 through 12

-0.2251	0.0768	0.0707	0.0744	0.0689	0.0699	0.0689	0.0632	0.0638	0.0711	0.0770

Columns 13 through 24

0.1006	0.1004	0.1023	0.0994	0.0879	0.0863	0.0831	0.0761	0.0729	0.0608	0.0571

Columns 25 through 30

0.0306	0.0231	0.0108	0.0068	0.0056	0.0060

Two Dimensional Vectors for Three Persons and Three Expressions

k>> two_1		
two_1 =		
	0.1323	-0.0459
	0.0414	-0.0645
K>> two_2	-0.1317	-0.1193
two_2 =		
	0.1362	0.0102
	0.1678	0.0039
	0.1678	0.0039
K>> two_3		
two_3 =		
	-0.1325	0.0981
	-0.0625	0.0586
	0.0249	-0.0319

The outputs of two_1 , two_2 and two_3 are plotted in Fig. 4 for understanding if there is any overlapping of expressions of different persons in order to get better classification of given expression of a particular person.

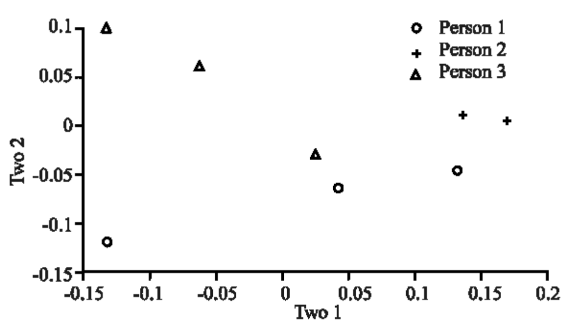

Fig. 4: Facial expressions

Distribution of the facial expressions

Testing the image randomly

Testing with the trained inputs: For testing the inputs of trained images or with image that was not used for

Table 1: Classification of the expressions of three persons

	Classification			
	Expression 1	Expression 2	Expression 3	Total classification
Person 1	CL	CL	CL	4
Person 2	CL	CL	CL	4
Person 3	CL	CL	CL	4
Person 4	CL	CL	NCL	4
Person 5	CL	CL	CL	3
				19/20 = 95%

training most of the steps of training have to be followed followed by using the final weights for final classification of the expressions.

The classification done by the program developed is given in Table 1.

In practice all the nine expressions of three persons should be correctly classified. However, there is one misclassification. This may be due to noise in the expression.

CONCLUSION

The faces of five persons with different orientations are considered for the project. Each orientation was trained using PCA, FLD and ESNN. Testing was done with the final weights obtained during training. A set of final weights was obtained. These weights are used for testing the existing face and detecting new face. The accuracy with which the project works is 95%.

REFERENCES

Alex Pentland, Baback Moghaddam and Thad Starner, 1994. View-based modular eigenspaces for face recognition. In Computer Vision and Pattern Recognition, pp: 84-91.

Antonio J. Colmenarez and Thomas S. Huang, 1997. Face Detection with information-Based Maximum Discrimination. In Computer Vision and Pattern Recognition, pp: 782-787.

Baback Moghaddam and Alex Pentland, 1995. Probalistic visual learning for object detection. In 5th international conference on computer vision, Cambridge, Massachusetts. IEEE. Computer Society Press, pp: 786-793.

Belhumeur, P.N., J.P. Hespanha and D.J Kriegman, 1997. Eigenfaces versus fiserfaces: Recognition using class specific linear projection. IEEE. Trans. Pattern Anal. Mach. Intell., 19: 711-720.

Brunelli, R. and T. Poggio, 1993. Face Recognition: Features versus templates. IEEE. Trans. Pattern Machine Intell., 15: 1042-1053.

Chellapa, R., C.L. Wilson and S. Sirohey, 1995. Human and machine recognition of faces: A survey. Proc. IEEE., 83: 705-740.

Gaungzheng Yang and Thomas S. Huang, 1994. Human face detection in a complex background. Pattern Recognition, 27: 53-63.

Gilles Burel and Dominique Carel. Detection and localization of faces on digital images. Pattern Recog. Lett., 15: 963-967.

Kah-Kay Sung, 1996. Learning and Example selection for object and pattern detection. PhD Thesis, M IT AI Lab, Available as AI Technical Report 1572.

Kin Choong Yow and Roberto Cippola, 1996. Feature-based human face detection. Technical report CUED/F-INFENG/TR 249, Department of Engineering, University of Cambridge, England.

Kirupy, M. and L. Sirvoich, 1990. Application of the Karhunen- Loveve procedure for the characterization of human faces. IEEE Trans. Pattern Anal. Mach. Intell., 12: 103-108.

Lin, S.H., S.Y. Kung and L.J. Lin. Face, 1997. Recognition/detection by probabilistic decision-based neural network. IEEE. Trans. Neural Networks, Special Issue on Artificial Neural Networks and Pattern Recog., pp: 8.

Lotlikar, R. and R. Kothari, 2000. Fractional step dimensionality reduction. IEEE Trans. Pa and T. Poggio, Face Recognition: Features versus templates, IEEE. Trans. Pattern Anal. Mach. Intell., 22: 623-627.

Lu, J., K.N. Plataniotis and A.N. Venetsanopoulos, 2003. Face recognition using Kernel direct discriminate analysis algorithm. IEEE. Trans. Neural Netw., 14: 195-200.

Lu, J., K.N. Plataniotis and A.N. Venetsanopoulos, 2003. Face recognition using LDA-based algorithms. IEEE. Trans. Neural Netw., 14: 195-200.

Martinez, A. and A. Kak, 2001. PCA versus LDA. IEEE. Trans. Pattern Anly. Machine Intell., 23: 228-233.

Meng Joo Er, Weilong Chen and Shiqian Wu, 2005. High-speed face recognition based on discrete cosine transform and RBF Neural network. IEEE. Trans. Neural Network, 16: 679-691.

Pawan Sinha, 1994. Object recognition via image invariants: A case study. Investigative Ophthalmol. Visual Sci., pp: 35.

Turk, M.A. and A.P. Pentland, 1991. Eigenfaces for recognition, J. Cogn. Neurrosci., 3: 71-86.

Vaillant, R., C. Monrocq and Y. Le Cun, 1994. Original approach for the localization of objects in Images. IEEE. Proceddings Vision, Image and Signal Proc., pp: 141.

Yu, H. and J. Yang, 2001. Adirect LDA algorithm for high dimensional data-with application to face recognition. Pattern Recog., 34: 2067-2070.