Extended Radial Network For Spectral Neuro Approach In Pattern Recognition

Dr G.Shyama Chandra Prasad1,.Prathibha 2, and A.Pramod Kumar3
[1-3] Associate Professor, Matrusri Engineering College, Hyderabad, T.S, India

ABSTRACT

The problem of pattern recognition in real time applications and its performance improvement in learning system is focused in this paper. The efficiency of learning systems for pattern recognition is improved by the integration of frequency resolution information with the extension of the conventional radial network. The extended neuro modeling with radial network and with the incorporation of frequency feature is observed to provide better accuracy in estimation than the conventional RBF-neuro modeling. A comparative evaluation is carried out for the retrieval accuracy for the developed recognition system and is evaluated for the Precision Recall rate for the pattern recognition system.

Keyword: frequency spectral estimation, extended radial network, neuro modeling, pattern recognition

I. INTRODUCTION

Pattern recognition systems provide important characteristics for surface and object identification from various imaging application such as aerial or satellite images, biomedical images natural images, etc. frequency analysis for pattern recognition is fundamental to many applications such as automated visual inspection, biomedical image processing, Content Based Image Retrieval (CBIR) and remote sensing. Much research work has been done on texture analysis, classification, and segmentation for the last four decades. Despite these efforts, texture analysis is still considered an interesting but difficult problem in image processing. Although the concept of texture was difficult to define, the studies showed that spatial statistics computed on the grey levels of the images were able to give good descriptors of the perceptual observation of texture. Such textural descriptors are more powerful tools for classification tasks or segmentation problems [14]. Over the past decade, the study of texture has been extended to the study of texture in natural images.

Approaches used for gray scale images are adapted to take into account the image information. The texture features are computed taking the correlations between the color bands into account. Descriptors are computed both within and between channels to give information on the whole texture. In the texture features, textural features are computed and then used as the basis of a classifier. Generally stating, for a finite number of classes C_i, i = 1,2,3,. . ., n. with a number of training samples of each classes available, based on the information extracted from the training samples, a decision rule is designed which classifies a given sample of unknown class into one of n classes. To design an effective algorithm for texture classification, it is essential to find a set of texture features with good discriminating power. The wavelet methods offer computational advantages over other methods for texture classification [1,2,10,19]. For almost all previously proposed techniques, the success of object recognition depends on the solution of two problems: representation and matching [1]. The representation of a pattern can be considered as feature extraction in pattern recognition. In [2], image features are divided into four groups: visual features, statistical pixel features, transform coefficient features, and algebraic features. The algebraic features represent intrinsic properties of an image and have good stability. Considering these features object recognition can be carried out based on learning approaches. Basically learning approaches were observed to provide more accuracy in estimation than the distance vector based approaches. In an unsupervised learning approach obtaining high recognition rate is a difficult task. Under conditions where we cannot acquire a large number of object images for every object, utilizing all available samples is very important. This means that not only positive samples but also negative samples need to be learned. A radial basis function (RBF) neural network classifier makes it possible to learn both positive and negative samples. Since the structure of RBF neural networks determines the performance of classification, we should design the network structure to satisfy our requirements. In order to utilize all available samples, a suitable RBF classifier which has learned both "positive" and "negative" samples in advance is needed.

Radial basis function (RBF) neural networks provide good possibilities for solving signal processing and pattern classification problems. Several algorithms have been proposed for choosing the RBF prototypes and training the network. The selection of the RBF prototypes and the network weights can be viewed as a system identification problem. In this paper a extension to the prototype selection is carried out based on the extended recursive learning approach. For the implementation of the suggested approach the extended recursive learning approach in RBF network is trained and tested with wavelet based feature vector resulting in faster computation and higher recognition accuracy.

The remaining paper is outlined in six sections where section II provides a brief outline to the frequency spectral feature coefficients derived for recognition, section III brief out the proposed extension of learing approach to RBF network based on recursive estimation approach, the analysis for the developed system based on the recognition accuracy and system precision is presented in section IV. A conclusion is made in section V.

II. FREQUENCY SPECTRAL REPRESENTATION

The recognition approach suggested is provided with the frequency spectrum information, for the estimation of resolution features description of the give image in pattern recognition. For the evaluation of the spectral information in this work db4 wavelet transformation approach is used. Wavelet transformation basically decomposes the given image into it's fundamental resolution and from the extracted spectral coefficient the resolution mean variation could be predicted and treated as feature information for recognition. In this paper a db4 Wavelet transformation is applied on the image data for extraction of features for training the data and testing for recognition. The DWT architecture developed split the image spectrum in two (equal) parts, a low pass and a high-pass part. The high-pass part contains the smallest details that are interested in and could stop here. However, the low-pass part still contains some details and therefore it can be split again. And again, until a satisfactory number of bands are have created. In this way an iterated filter bank can be created.

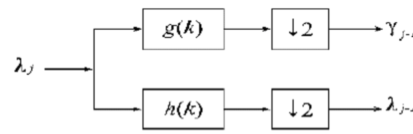


Figure 1: Implementation of one stage iterated filter banks

Usually the number of bands is limited by for instance the amount of data or computation power available. The process of splitting the spectrum is shown in figure 1. The advantage of this scheme is to design only two filters whereas the disadvantage is; only image spectrum coverage is fixed. Wavelet transform is capable of providing the time and frequency information simultaneously. Hence it gives a time-frequency representation of the signal. When one is interested in knowing what spectral component exists at any given instant of time, to know the particular spectral component at that instant. In these cases it may be very beneficial to know the time intervals these particular spectral components occur. Wavelets (small waves) are functions defined over a finite interval and having an average value of zero. The basic idea of the wavelet transform is to represent any arbitrary function $f(t)$ as a superposition of a set of such wavelets or basis functions. These basis functions are obtained from a single wave, by dilations or contractions (scaling) and translations (shifts). wavelet has two functions "wavelet "and "scaling function". They are such that there are half the frequencies between them. They act like a low pass filter and a high pass filter. Figure 2-6 shows a typical decomposition scheme. The decomposition of the signal into different frequency bands is simply obtained by successive high pass and low pass filtering of the time domain signal. This filter pair is called the analysis filter pair. First, the low pass filter is applied for each row of data, thereby getting the low frequency components of the row. But since the low pass filter is a half band filter, the output data contains frequencies only in the first half of the original frequency range. By Shannon's Sampling Theorem, they can be sub-sampled by two, so that the output data now contains only half the original number of samples. Now, the high 8 pass filter is applied for the same row of data, and similarly the high pass components are separated.

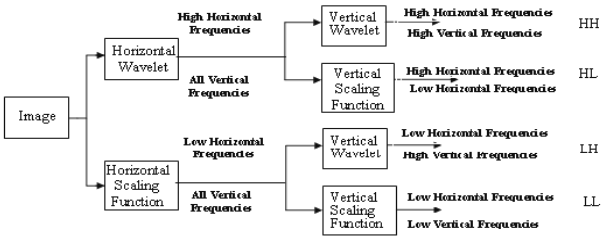
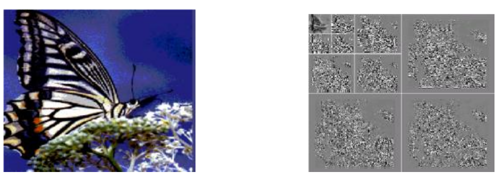


Figure 2: Pyramidal Decomposition of an image

This is a non-uniform band splitting method that decomposes the lower frequency part into narrower bands and the high-pass output at each level is left without any further decomposition. This procedure is done for all rows. Next, the filtering is done for each column of the intermediate data. The resulting two-dimensional array of coefficients contains four bands of data, each labeled as LL (low-low), HL (high-low), LH (low-high) and HH (high-high).

These resolution features are averaged over the mean to process the query image for recognition. For a given query image



(a) (b)

Figure 3: (a) Original Image, (b) multi scaled image using db4 wavelet coefficient

In the proposed feature extraction methodology each Image I, is decomposed into three color channels Ci, where i = 1, 2, 3. Each channel is raster scanned with a fixed size sliding square window. On each window a K-level 2D-Discrete Wavelet Transform (DWT) is applied. The Daubechies-4 wavelet bases were used due to their orthonormal properties, which are important for the preservation of the textural structure along the different scales of the transform. This transform results in a new representation of the original window, which consists of B=3K+1 sub-windows, corresponding to different wavelet bands. Each band is denoted as $B_j(k)$, where k is the current level of the transform and j = 0, 1, 2, 3 for k = K, or j = 1, 2, 3 for k < K. $B_0(k)$ corresponds to the low frequency band. The textural information contained in each window is captured with the use of co occurrence matrices. Co occurrence matrices encode the gray level spatial dependence based on the estimation of the 2nd order joint conditional probability density function f(i, j, d, a), which is computed by counting all pairs of pixels at distance d having gray levels i and j at a given direction a. The angular displacement of d = 1 is included in the range of the a-values {0, π/4, π/2, 3π/4}. Four Haralick's measures, namely the angular second moment (f1), the correlation (f2), the inverse difference moment (f3) and the entropy (f4). These four features provide high discrimination accuracy which can only be marginally increased by adding more features in the feature vector . The features f1- f4 are estimated over each sub-window $B_j(k)$, j ≠ 0, k = 1, 2, … K, of the color channels Ci, i = 1, 2, 3 of the frame and they are noted as:

$$F_{C_i}^{B_j(k)}(a),$$

j ≠ 0, k = 1, 2, … K, where F Є { f1, f2, f3, f4 } and a corresponds to the angle considered in the estimation of the cooccurrence matrices, a Є {0, π/4, π/2, 3π/4}. We define Color Wavelet Covariance of a feature F Є { f1, f2, f3, f4 } at wavelet band $B_j(k)$, j ≠ 0, k = 1, 2, … K, between two color channels C_l and C_m as:

$$CWC^{B_j(k)}(C_l, C_m) = Cov\left(F_{C_l}^{B_j(k)}, F_{C_m}^{B_j(k)}\right)$$

estimated over the different angles a. For K=1, thecorresponding feature vectors consist of 72 CWC features((3 variances + 3 covariances) x 4 cooccurrence matrices x3 wavelet bands). These features are passed to the learning method based on extended recursive RBF (E-RBF) architecture for recognition.

III. EXTENDED RADIAL NETWORK

A radial basis function (RBF) neural network is trained to perform a mapping from an m-dimensional input space to an n-dimensional output space. RBFs can be used for discrete pattern classification, function approximation, image processing, control, or any other application which requires a mapping from an input to an output.

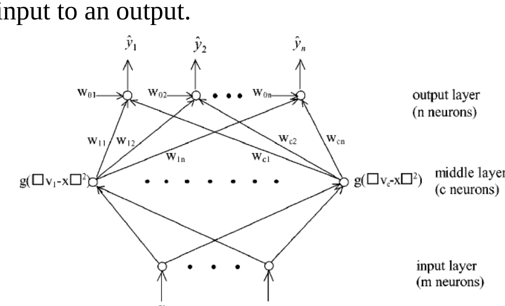


Figure 4: RBF network architecture

An RBF consists of the m-dimensional input ‘x’ passed directly to a hidden layer. Suppose there are ‘c’ neurons in the hidden layer. Each of the ‘c’ neurons in the hidden layer applies an activation function which is a function of the Euclidean distance between the input and an m-dimensional prototype vector. Each hidden neuron contains its own prototype vector as a parameter. The output of each hidden neuron is then weighted and passed to the output layer. The outputs of the network consist of sums of the weighted hidden layer neurons. Figure 4 shows a schematic of a generic RBF network. It can be seen that the design of an RBF requires several decisions, which includes;

1) How many hidden units will reside in the hidden layer (i.e., what is the value of the integer c);

2) What are the values of the prototypes (i.e., what are the values of the v vectors);

3) What function will be used at the hidden units (i.e., what is the function g);

4) What weights will be applied between the hidden layer and the output layer (i.e., what are the values of the w weights).

The performance of an RBF network depends on the number and location (in the input space) of the centers, the shape of the RBF functions at the hidden units, and the method used for determining the network weights. Some researchers have trained RBF networks by selecting the centers randomly from the training data [4]. Others have used unsupervised procedures (such as the k-means algorithm) for selecting the RBF centers [11]. Still others have used supervised procedures for selecting the RBF centers [9]. Several training methods separate the tasks of prototype determination and weight optimization. This trend probably arose because of the quick training that could result from the separation of the two tasks. In fact, one of the primary contributors to the popularity of RBF networks was probably their fast training times as compared to gradient descent training include back propagation. it can be seen in figure 4 that once the prototypes are fixed and the hidden layer function 'g' is known, the network is linear in the weight parameters '*w*'. At that point training the network becomes a quick and easy task that can be solved via linear least squares. Training methods that separate the tasks of prototype determination and weight optimization often do not use the input–output data from the training set for the selection of the prototypes. For instance, the random selection method and the k-means algorithm result in prototypes that are completely independent of the input–output data from the training set. Although this results in fast training, it clearly does not take full advantage of the information contained in the training set. Gradient descent training of RBF networks has proven to be much more effective than more conventional methods [9]. However, gradient descent training can be computationally expensive. To minimize the computational effort a training method for RBF network based on Recursive estimation approach. This new method proves to be quicker than gradient descent training while providing performance at the same level of accuracy. Training a neural network is, in general, a challenging nonlinear optimization problem. Various derivative-based methods have been used to train neural networks, including gradient descent [9], and the well-known back propagation approach [7]. Derivative-free methods, including genetic algorithms [6], learning automata [12], and simulated annealing [10], have also been used to train neural networks. Derivative-free methods have the advantage that they do not require the derivative of the objective function with respect to the neural network parameters. They are more robust than derivative-based methods with respect to finding a global minimum and with respect to their applicability to a wide range of objective functions and neural network architectures. However, they typically tend to converge more slowly than derivative-based methods. Derivative-based methods have the advantage of fast convergence, but they tend to converge to local minima. In addition, due to their dependence on analytical derivatives, they are limited to specific objective functions and specific types of neural network architectures.

In this paper the recursive estimation filters have been used extensively with neural networks. They have been used to train multilayer perceptrons [17,20,21] and recurrent networks [13]. They have also been used to train RBF networks, but so far their application has been restricted to single-output networks with exponential functions at The hidden layer [3]. In this paper we extend the use of Recursive estimation filters to the training of general multi-input, multi-output RBF networks.

Considering a system is trained by a set of M desired input–output responses {xi; yi} (i=1; : : : ; M).There have been a number of popular choices for the g(.) function at the hidden layer of RBFs. The most common choice is a Gaussian function of the form

$$g(v) = \exp\,(-v/\beta^2)$$

Where β is a real constant. Other hidden layer functions that have often been used are the thin plate spline function

$$g(v) = v log\sqrt{v}$$

The response of system function can be written as

$$\hat{y} = \begin{bmatrix} w_{10} & w_{11} & \cdots & w_{1c} \\ w_{20} & w_{21} & \cdots & w_{2c} \\ \vdots & \vdots & \vdots & \vdots \\ w_{n0} & w_{n1} & \cdots & w_{nc} \end{bmatrix} \begin{bmatrix} 1 \\ g(||x - v_1||^2) \\ \vdots \\ g(||x - v_c||^2) \end{bmatrix}.$$

In the right hand side the first matrix will represents the weight matrix this can shorted as w. The above equation can be written as

$$\hat{y} = WH$$

$$h_{ok} = 1 \quad (k=1,\ldots\ldots,M),$$
$$h_{jk} = g(//x_k - v_j//^2)(k=1,\ldots,M),(j=1,\ldots c)$$

$$\begin{pmatrix} h_{01} & \ldots & h_{0M} \\ h_{11} & \ldots & h_{1M} \\ . & . & . \\ . & . & . \\ h_{c1} & \ldots & h_{cM} \end{pmatrix} = [\, h_1, h_2 \ldots\ldots\ldots\ldots h_M \,] = H$$

taking the response of the system at time k is given by

$$y_k = h(\theta_K) + v_k$$

and $$\theta_{K+1} = f(\theta_K) + \omega_k$$

where the vector θ_K is the state of the system at time k, y_k is the observˆation vector, v_k is the observation noise, and f(.) and h(.) are nonlinear vector functions of the state.
For the efficient performance of system is done by applying recursive updation of the state by

$$\theta_K = f(\theta_{K-1}) + K_k[y_k - h(\theta_{K-1})],$$

$$K_k = P_k H_k (R + H_k^T P_k H_K)^{-1}$$

$$P_{k+1} = F_k (P_k - k_K H_k^T P_k) F_k^T + Q$$

K_k is known as the gain.

IV. Simulation Observation

The system architecture of the proposed system is shown below

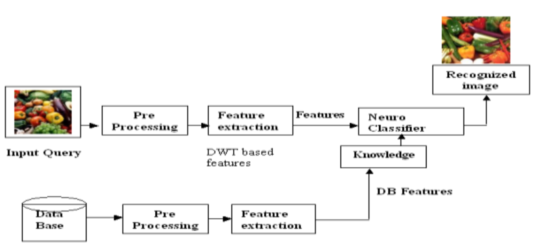


Figure 5: system architecture

The suggested approach is implemented using MATLAB tool. The developed system is tested for different types of images.The results are generated based on the above suggested approach.

The data base images used for system is shown below.

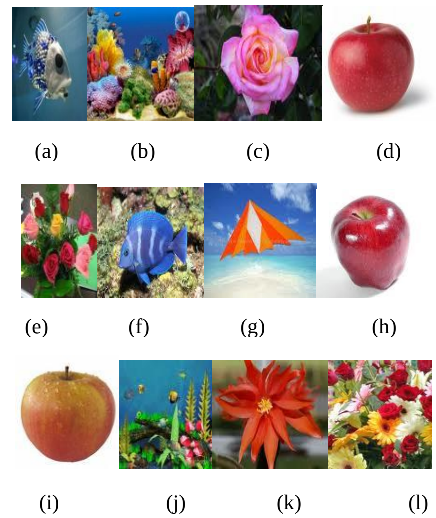


Figure 6: Data base images

for the developed system given query image as input as shown in below.

Figure 7.input query image

The result obtained from the system is shown below

Figure 8.output image of the system

Table 1.specified parameters and performance

Feature dimension	Training phase			Testing phase
	RBF units	β	θ	NOM*
12	50	0.9~0.95	0.9~2.2	10
23	50	0.9~0.93	1.2~2.0	8
28	50	0.8~0.65	0.99~1.4	4
35	50	0.55~0.7	0.88~1.2	4
40	50	0.55~0.65	1.0~1.1	6
45	50	0.5~0.7	1.0~1.1	6

* NOM—Number of Misclassifications

After the structure of extended RBF neural networks and parameters of prototypes are selected, the suggested algorithm is trained and the one run of the recognition results is shown in above.

According to observations, if the information is sufficient (feature dimension is larger than 23), the results are stable in each case for different choice of initial parameters and in terms of the number of misclassifications.and results in higher performance.

The reason may be that high dimension will lead to complexity in structure and increase difficulty in learning. Moreover, the addition of some unimportant information may become noise and degrade the performance.The best results are achieved when the dimensionis 28–35.

Along with the increase in the feature dimension, the training patterns have more overlapping, and a small should be selected.

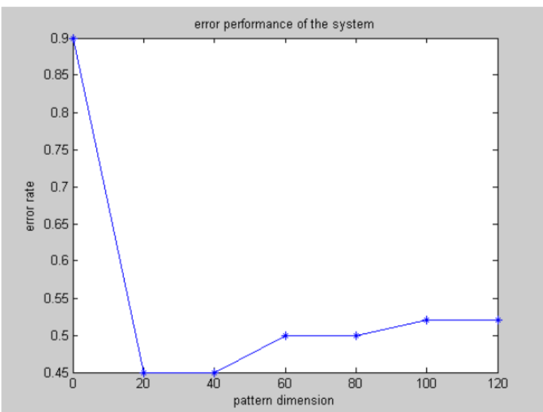


Figure 9:Error performance of the system

The above figure shows error performance of the system with patter dimensions the system performance is higher the patter dimension in the range from 20 to 40

V. Conclusion

The neural network architecture depends heavily on the availability of effective learning algorithms. The theoretical strength of the Recursive estimation filter has led to its use in hundreds of technologies, and this paper demonstrates that RBF network training is yet another fruitful application of Recursive estimation approach. The experiments reported in this paper verify that Recursive estimation filter training provides about the same performance as gradient descent training, but with only a fraction of the computational effort. In addition, it has been shown that the decoupled Recursive estimation filter provides performance on par with the standard Recursive estimation filter while further decreasing the computational effort for large problems.

VI. References

[1] B. Anderson, J. Moore, Optimal Filtering, Prentice-Hall, Englewood Clis, NJ, 1979.
[2] J. Bezdek, J. Keller, R. Krishnapuram, L. Kuncheva, H. Pal, Will the real Iris data please stand Up" IEEE Trans. Fuzzy Systems 7 (1999) 368–369.
[3] M. Birgmeier, A fully Kalman-trained radial basis function network for nonlinear speech modeling, IEEE International Conference on Neural Networks, Perth, Western Australia, 1995, pp. 259–26ff.
[4] D. Broomhead, D. Lowe, Multivariable functional interpolation and adaptive networks, Complex Systems 2 (1988) 321–355.
[5] S. Chen, C. Cowan, P. Grant, Orthogonal least squares learning algorithm for radial basis function networks, IEEE Trans. Neural Networks 2 (1991) 302–309.
[6] S. Chen, Y. Wu, B. Luk, Combined genetic algorithm optimization and regularized orthogonal least squares learning for radial basis function networks, IEEE Trans. Neural Networks 10 (1999) 1239–12ff3.
[7] R. Duro, J. Reyes, Discrete-time backpropagation for training synaptic delay-based artificial neural networks, IEEE Trans. Neural Networks 10 (1999) 779–789.
[8] A. Gelb, Applied Optimal Estimation, MIT Press, Cambridge, MA, 197ff.
[9] N. Karayiannis, Reformulated radial basis neural networks trained by gradient descent, IEEE Trans. Neural Networks 3 (1999) 657–671.
[10] S. Kirkpatrick, Cl. Gelatt, M. Vecchi, Optimization by simulated annealing, Science 220 (1983) 671–680.
[11] J. Moody, C. Darken, Fast learning in networks of locally-tuned processing units, Neural Computation. (1989) 289–303.
[12] K. Narendra, M. Thathachar, Learning Automata—An Introduction, Prentice-Hall, Englewood Cliffs, NJ, 1989.
[13] D. Obradovic, On-line training of recurrent neural networks with continuous topology adaptation, IEEE Trans. Neural Networks 7 (1996) 222–228.
[14] G. Puskorius, L. Feldkamp, Neurocontrol of nonlinear dynamical systems with Kalman filter trained recurrent networks, IEEE Trans. Neural Networks 5 (199ff) 279–297.
[15] V.D. SaUnchez, A. (Ed.), Special Issue on RBF Networks, Part I, Neuro computing 19 (1998).
[16] V.D. SaUanchez, A. (Ed.), Special Issue on RBF Networks, Part II, Neuro computing 20 (1998).
[17] S. Shah, F. Palmieri, M. Datum, Optimal filtering algorithms for fast learning in feed forward neural networks, Neural Networks 5 (1992) 779–787.
[18] D. Simon, Distributed fault tolerance in optimal interpolative nets, submitted for publication.
[19] S. Sin, R. DeFigueiredo, Efficient learning procedures for optimal interpolative nets, Neural Networks 6 (1993) 99–113.