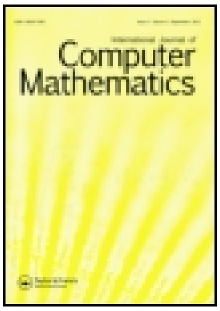

International Journal of Computer Mathematics
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/gcom20

An efficient multiresolution texture classification scheme using neural networks

Anatasios Delopoulos [a] , Levon Sukissian [a] & Stefanos Kollias [a]
[a] Computer Science Division, National Technical University of Athens, Zografou, Athens, 15773, Greece
Published online: 20 Mar 2007.

Intern. J. Computer Math., Vol. 67, pp. 155–168
Reprints available directly from the publisher
Photocopying permitted by license only

AN EFFICIENT MULTIRESOLUTION TEXTURE CLASSIFICATION SCHEME USING NEURAL NETWORKS

ANATASIOS DELOPOULOS, LEVON SUKISSIAN
and STEFANOS KOLLIAS

Computer Science Division, National Technical University of Athens, Zografou 15773, Athens, Greece

(Received 14 April 1997)

An efficient multiresolution texture classification method is proposed in this paper, based on 2-D linear prediction, multiresolution decomposition and artificial neural networks. A multiresolution spectral analysis of textured images is first developed, which permits 2-D AR texture modelling to be performed in multiple resolutions. Recursive estimation algorithms combined witth the Itakura distance measure provide sets of AR model parameters representing different textures at various resolutions. Appropriate neural network banks are constructed and trained being then able to effectively perform classification of textures irrespective of their resolution level. Results are presented using real textured images which illustrate the good performance of the proposed approach.

Keywords: Texture classification; 2D linear prediction; multiresolution decomposition; ANNs

1. INTRODUCTION

Artificial neural networks have been widely adopted in recent years as a powerful tool for providing intelligent solutions in a wide range of problems and applications. In the fields of pattern recognition and signal/image processing, the nonlinear nature of neural networks, combined with their ability to learn from examples, permit the derivation of effective analysis, classification and diagnosis approaches, even in noisy or time and space varying environments. Various applications, as well as commercial products, based on neural networks are being developed, including recognition of handwritten characters, analysis of medical images, speech and face recognition and classification/segmentation of textured images or scenes [7, 8].

Analysis and processing of images include tasks such as classification and segmentation, labeling and interpretation, coding and management of the images. The design and use of neural network architectures in these problems is a subject of on-going research, which involves many other tasks such as image and texture modelling for processing or restoration purposes, as well as morphological operators for segmentation tasks [1, 4].

The basic problem when using neural networks for analysis of real-life images, e.g., aerial or satellite images, is the large size of the images, which causes problems of efficiency of training and of the generalization ability of the networks. The design of efficient versions of learning algorithms [7, 9] and of structured and modular networks [10, 11] is a possible solution for such problems. Such a solution is in accorodance with recent theoretical results on feed-forward network generalization, which refer to pruning or constructive network design, as well as the VC dimension and the requirement for small sized networks [2, 17]. In addition, however, ambiguities regarding the resolution level of the images handled by the network may result in erroneous decisions of the classification and/or segmentation procedures.

Classification of images based on their textural content is a task that emerges in various applications. Automated digital cartography, military supervision, medical imaging and biochemistry are only a few of the areas involving such a methodology. Among the most popular approaches in textural analysis of image data is the parametric modeling of textures accomplished by fitting the coefficients of autoregressive (AR) or autoregressive moving average (ARMA) models to the statistics of the image; the image is in this case approximated by a random field produced by exciting the aforementioned model by a 2-D white noise sequence. Texture classification is then accomplished through classification of the parameter vector containing the estimated coefficients of the linear model.

Various adaptive schemes have been proposed in the literature in order to estimate these coefficients in a computationally attractive way. A variety of conventional classification techniques have been adopted as a means for separating textures in different categories [14].

In the present work we propose a classification/segmentation algorithm based on neural networks and linear prediction modelling which is applied to scenarios that involve images appearing in multiple resolutions. These scenarios include satellite surveillance from varying heights, segmentation of medical, biochemical and textile microscopical images at varying magnification factors, categorization of texture pictures shot from varying azimuth. Multiple resolution images are met, for example, in LANDSAT imaginery, where new satellites are launched providing images of five times higher resolution than the existing ones; images taken from geographical information systems are also available at varying resolution levels [18]. More specifically, the goal is to exploit information regarding the autocorrelation structure and the associated model coefficients acquired in multiple resolutions; the multiresolution versions of the images, in analog form, are assumed to be related through a wavelet based representation [12], which, in digital form, corresponds to 2-D subband analysis. The obtained AR feature vectors at multiple resolutions are processed for classfication purposes by appropriate feedforward neural networks that exploit the multiresolution information in both the training and testing stages.

Section II of the paper describes the application of multiresolution analysis to 2-D images, deriving appropriate relations between autocorrelation structures of images in successive resolutions. In Section III we exploit the results of Section II in order to compute 2-D AR image models at various resolutions. Batch and space recursive estimation algorithms are considered. The use of neural networks in view of the obtained results are considered in Section IV for the implementation of multiresolutions, multiscale classification tasks. Section V contains experimental results based on synthetic and real textured images.

2. MULTIRESOLUTION ANALYSIS OF TEXTURED IMAGES

Representation of signals at many resolution levels has gained much popularity especially with the introduction of the discrete wavelet transform, implemented in a straightforward manner by filter banks using appropriate low- and high-pass filters [3, 16]. In image processing the above are equivalent to subband filtering [19]. Multiresolution decomposition results in approximation images of low resolution that contain coarser information of the image content and in a set of detail images which contain more information as resolution is gradually decreasing.

For an image at resolution level i its corresponding lower resolution version is

$$x_{i-1}(m,n) = \sum_{k=1}^{N_i} \sum_{l=1}^{N_i} h(2m-k, 2n-l) x_i(k,l), \tag{1}$$

where N_i is the dimension of the image at level i, $h(.,.)$ is the blurring mechanism that relates adjacent resolution versions and decimation by 2 indicates down scaling by a factor 2 when moving from one resolution to the lower one. Arbitrary (rational) down-scaling factor L/M can be achieved by appropriately cascading decimators and imterpolators,

$$x_{i-1}(m,n) = \sum_{k=1}^{N_i}\sum_{l=1}^{N_i} h(Mm - Lk, Mn - Ll)x_i(k,l). \tag{2}$$

For the ease of presentation, the subsequent analysis follows the paradigm of Eq. (1) rather than that of Eq. (2).

In the rest of this section we establish a useful relation between the autocorrelation of images that appear in different resolutions. This relation in next used in order to obtain relations between the coefficients of 2-D AR structures fitted to different resolutions of the same textured image.

Assuming that $x_i(m, n)$ is a 2D random field (texture) with zero mean $(E\{x_i(m,n)\} = 0)$ we denote by

$$r_i(s,t) \triangleq E\{x_i(m,n)x_i(m+s,n+t)\}, \tag{3}$$

the corresponding 2D autocorrelation function. In view of Eqs. (1) and (3) $r_{i-1}(s,t)$ and $r_i(s, t)$ are related by

$$\begin{aligned} r_{i-1}(s,t) &= \sum_{k_1,l_1}\sum_{k_2,l_2} h(2m-k_1, 2n-l_1) \\ &\quad h(2m+2s-k_2, 2n+2t-l_2)E\{x(k_1,l_1)x(k_2,l_2)\} \\ &= \sum_k\sum_l h_2(2s-k, 2t-l)r_i(k,l), \end{aligned} \tag{4}$$

where

$$h_2(s,t) \triangleq \sum_k\sum_l h(k,l)h(k+s,l+t). \tag{5}$$

Equations (4) and (5) imply that the 2D autocorrelation of the decimated signal can by computed by decimating the 2D autocorrelation of the higher level using $h_2(s,t)$, which is the autocorrelation of $h(m,n)$, as decimating filter.

Similar relations exist between the autocorrelation functions of rationally decimated signals. In [5] such relations are established between cumulants of any order for rationally related resolution levels.

3. MULTIRESOLUTION 2-D AR IMAGE MODELLING

A common means of characterizing texture images is that of fitting Autoregressive (AR) structures that represent the inter-pixel relation of the form,

$$x_i(m,n) = \mathbf{c}_i^T \mathbf{x}_i(m,n) + u(m,n), \tag{6}$$

where the pixel value at (m, n) is approximated by a linear combination of pixels contained in the vector $\mathbf{x}_i(m,n)$ corresponding to a neighborhood $\mathbf{N}(m, n)$ and $u(m, n)$ represents the approximation error ordinarily modeled as a white noise process. The associated AR coefficients collected in the column vector $\mathbf{c}_i$ constitute a finite dimension feature vector characterizing the statistics of $x_i(m, n)$. The choice of the pixels included in $\mathbf{N}(m, n)$ varies depending on the nature of the images. The non-symmetric half plane (NSHP) model was used by the authors in the past yielding satisfactory results to a wide variety of natural textured images [1, 14].

Given $x_i(m, n)$ the optimal estimate $\hat{\mathbf{c}}_i$ of $\mathbf{c}_i$ is obtained solving the normal equations,

$$\mathbf{R}_i \hat{\mathbf{c}}_i = \mathbf{r}_i, \tag{7}$$

where for each resolution level i both $\mathbf{R}_i \underline{\underline{\Delta}}\ E\{\mathbf{x}_i(m,n)\mathbf{x}_i(m,n)^T\}$ and $\mathbf{r}_i \underline{\underline{\Delta}}\ E\{x_i(m,n)\mathbf{x}_i(m,n)\}$ contain appropriate lags of the autocorrelation structure $r_i(s, t)$.

Conversely, multiplying both sides of Eq. (6) by $x_i(m+s, n+t)$ and taking expectation yields,

$$r_i(s,t) = \mathbf{c}_i^T \mathbf{r}_i(s,t), \tag{8}$$

under the causality assumption, i.e., $E\{x_i(m+s,\ n+t)\ u(m,\ n)\} = 0$. In Eq. (8) the $\mathbf{r}_i(s, t)$ vector contains autocorrelation lags depending on the specific neighborhood $\mathbf{N}(m, n)$. Eq. (8) indicates that $r_i(s, t)$ is itself a self driven autoregressive function which is fully characterized by $\mathbf{c}_i$ apart of a scaling ambiguity.

As a consequence of the preceding analysis, a multiresolution feature extraction scheme can be implemented according to the following algorithmic steps.

ALGORITHM 1

STEP 1 Estimate the autocorrelation structure as some high resolution level $i=0$.

STEP 2 Solve for $\mathbf{c}_0$ using Eq. (7) with $i=0$.

STEP 3 Compute $r_{-1}(s, t)$ from $r_0(s, t)$ employing Eq. (4).

STEP 4 Use $r_{-1}(s, t)$ to formulate $\mathbf{R}_{-1}$, $\mathbf{r}_{-1}$ and compute $\hat{\mathbf{c}}_{-1}$using Eq. (7).

STEP 5 Repeat steps 3 and 4 for $i=-2, -3, \ldots$.

The algorithm above produces a bank of AR coefficient vector corresponding to successive resolution levels $i=0, -1, -2, \ldots$ while only a single autocorrelation estimation procedure is performed at the highest level.

A slight modification of Algorithm 1 allows the transition between AR coefficient vectors corresponding to successive resolution levels.

ALGORITHM 2

Assuming that $\mathbf{c}_0$ is given (for $i=0$)

STEP 1 Use Eq. (8) to obtain an approximation of $r_0(s, t)$.

STEP 2 Follow steps 3 to 5 of Algorithm 1.

Solution of Eq. (7) at any level i can be performed using space recursive algorithms exploiting the specific shape of the neighborhood $\mathbf{N}(m, n)$. Such a recursive implementation for the extended NSHP model is described in [1]. The application of such recursive algorithms produces pixel dependent AR estimates $\hat{c}_i(m,n)$ of $\mathbf{c}_i$, thus handling possible nonstationarities of the textured images.

The additional advantage of the aforementioned recursive implementation is its ability to produce more than one AR feature vectors for a single class of textures rather than a single representative which results from the batch solution of Eq. (7). A detailed description of this procedure is presented in [8] where the Itakura distance

$$d = \log \frac{c_i^T(m,n)\mathbf{R}_i(m,n+1)\mathbf{c}_i(m,n)}{\mathbf{c}_i^T(m,n+1)\mathbf{R}_i(m,n+1)\mathbf{c}_i(m,n+1)}, \tag{9}$$

in conjunction to appropriate thresholds is used for texture classification and segmentation. The outcome of the procedure used therein is a collection of AR vectors for each class of different textures. The major importance of this approach relies on the handling of local nonstationarities in the interior of a single texture field.

4. THE MULTIRESOLUTION NEURAL NETWORK CLASSIFICATION BANK

Artificial Neural Network (ANN) structures have shown good performance in texture classification based on pre-extracted AR feature vectors [8, 15]. In the present work we examine the performance of neural network architectures in classifying textured images at multiple resolution levels. The vehicle towards this goal is the analysis of Section III and in particular Algorithms 1 and 2 which are employed both in the training and testing phases.

Training

The training procedure in multiresolution setup targets the production of a bank of ANNs, one for every possible resolution level i, that gather the necessary information for classification of AR features computed in any level. Let NN_i denote the network classifier at level i. If the multiresolution procedure proposed in Section III was not followed, neural network training would require availability of the textures to be classified at all resolution levels of interest. An AR coefficient feature extraction procedure (c.f. Eq. (7)) in batch or recursive mode would be used next followed by presentation of the obtained feature vectors to the networks at each level. The drawback of this approach is twofold:

(1) The prototype textures must be available at (or should be converted to, using. Eq. (4)) all levels of interest
(2) The computationally demanding estimation of $\mathbf{R}_i$, $\mathbf{r}_i$, directly from $x_i(m, n)$, should be done at each level i.

Both (1) and (2) introduce serious memory and complexity constrains that may turn the overall task practically impossible. In the present work we propose the use of Algorithm 1 that allows estimation of the AR coefficients $\mathbf{c}_i$ at all levels i based on the autocorrelation estimates computed at the single highest resolution level ($i=0$). Clearly, the computational complexity of steps 3−5 is much lower than the re-estimation of the autocorrlelation function directly from the textured image samples at each level. At the same time the memory requirements are substantially reduced. The proposed scheme is even more effective since more than one AR feature vectors are associated to each class of textures. In this scenario the Itakura distance based procedure is employed at the highest level ($i = 0$) producing multiple representatives $\mathbf{c}_{0,k}(k = 1, 2, \ldots)$ for each class k while Algorithms 1 or 2 is used for computing the corresponding AR feature vectors $\mathbf{c}_{i,k}$ for all lower resolution levels $i(i = -1, -2, \ldots)$.

The above techniques refer to the pre-processing stage required in the construction of feature vector sets appropriate for NN classifiers operating at multiple resolutions. It should be emphasized that, if a conventional classifier based on pattern matching was employed, the extraction and use of a probably large number of AR feature vectors per texture class would increase the computational load during the operation (testing) phase. The use of neural networks, however, for this task does not inherit this drawback, provided that good generalization is achieved since the patterns have been embedded in the fixed number of network weights. Moreover, to obtain good generalization of the network we propose the following procedure for constructing it during training. Let us consider first a two texture classification problem and train a network using, e.g., a back-propagation variant [9, 13] and a set of reference patterns extracted from the textures as described in Section III. In order to add a new texture class to the networks, we follow a constructive approach, inspired by the cascade correlation algorithm [6] we first freeze the interconnection weights of the already trained network and expand the network architecture by adding a new output unit and one or more hidden units. We then train the non-fixed part of the network, consisting of the thresholds of all three output units and of the weights corresponding to connections due to the added unit(s). Using this approach a small number of free parameters is estimated during each network training step, generally resulting in small sized networks with good generalization abilities.

Testing

The NN structure employed for testing consists of a bank of independent networks, each operating on a specific resolution level i. In practice, a pre-defined depth of levels, say $i = 0, -1, \ldots, -q$ is assumed to be of interest. The procedure to follow depends on whether the resolution level of the incoming pattern is a-priori known or not.

If it is known, a standard AR parameter estimator is used (c.f. Eq. (7)) in order to extract the feature vector $\mathbf{c}_i$. The vector is next fed to the i-th neural network (NN_i) that performs classification at resolution level i. In this case the use of the preceding multiresolution analysis of Section III is limited to the training of the networks as described in the previous subsection.

On the contrary, if the resolution level of the incoming textured image is not known, the proposed bank of networks can simultaneously identify the correct class of the texture and its resolution. To this end Algorithm 1 or 2 is used to produce a series $\mathbf{c}_k$, $\mathbf{c}_{k-1}, \ldots, \mathbf{c}_{k-0}$ of AR coefficients corresponding to successive resolutions of the texture to be classified. The subscript k represents the unknown resolution of the given texture. At the next step hypotheses $H_i = \{k = i\}$ for $i = 0, \ldots, -(q-p)$ are iteratively tested using the following rule:

"*H_i is true if and only if the decisions of neural networks NN_j for all $j = i$, $i-1, \ldots, p$ coincide regarding the classification of the given pattern*".

The procedure terminates upon some i_0 for which H_{i_0} is true. In this case it is indicated that the resolution level $k = i_0$ and the correct class of the texture is given by the common decision of the employed neural networks. In general, the above procedure will provide correct classification of textures, irrespectively of their resolution level, assuming that no texture, at some resolution k is statistically identical to a different texture at another resolution l.

The operation diagram of Figure 1 illustrates the simultaneous classification and resolution identification procedures included in the proposed multiresolution neural network bank. The instance shown in this figure corresponds to the i-th iteration where the hypothesis $H_i : k = i$ is tested. A sequential operation is assumed, where the currently examined feature vector $\mathbf{c}_k$ and the computed vectors $\mathbf{c}_{k-j}$, $j = 1, \ldots, p$ are sequentially presented at the input of a subset of $p+1$ corresponding neural networks in the bank. This operation can be viewed as following a sliding window procedure which sequentially uses a subset of the network classifiers at resolutions $i, i-1, \ldots i-p$ for $i = 0, -1, \ldots, -(q-p)$. To avoid the sequential, in the form of sliding window, operation, and favor parallel implementation, one may consider keeping p copies of each network and perform the above operation in parallel.

Despite the fact that a possibly large bank of neural networks may be required for this task, by taking advantage of the massive parallelism of the networks [10, 11], a parallel implementation of the bank is feasible in high performance parallel computing environments both in training and in testing.

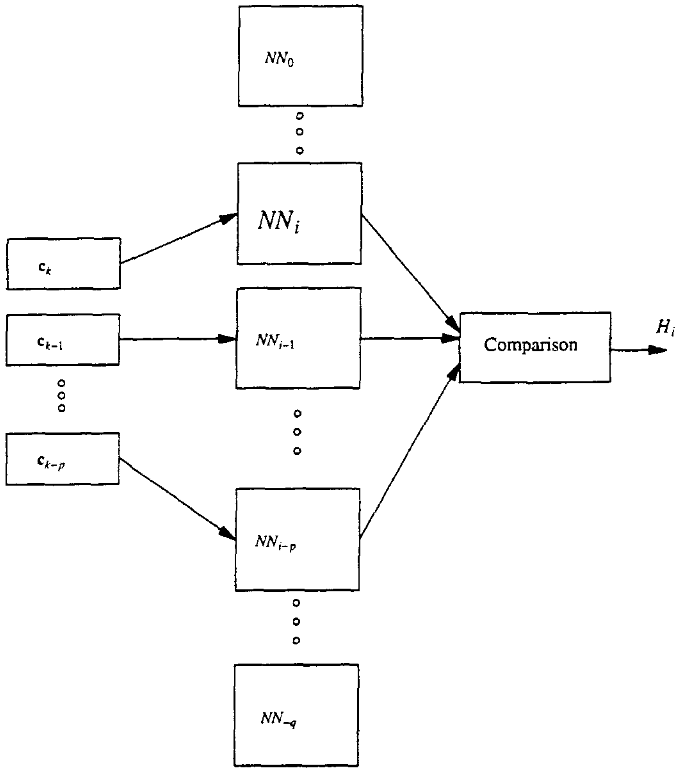

FIGURE 1 Operation diagram of the neural network bank for simultaneous classification and resolution identification of feature vector $\mathbf{c}_k$ (i-th iteration).

5. SIMULATION RESULTS

Test Case #1

In the first experiment we used three different textures, p.1.1.1, wall and sand, taken from the Brodatz album with sizes 512×512, 256×256 and 256×256 respectively. Parts of these images of size 100×100 pixels were used to extract characteristic NSHP model coefficient vectors at, what was considered as, high resolution level. Seven (7) characteristic vectors were extracted from the first picture, eight (8) from the second and five (5) from the third one. We used the extracted $\hat{\mathbf{c}}_0$ vectors to train the two hidden layer network classifier and then tested its performance over the other parts of the images. The good network performance is indicated from the results shown in Table I.

TABLE I Test Case #1: Classification of real life textured images at different (known) resultion levels

Image	*Success Rate* (%) *Resolution* (0)	*Resolution* (−1)
p.1.1.1.	95.0%	93.6%
wall	93.4%	90.8%
sand	94.7%	91.1%

We then used the analysis (decimating) separable filter $h(m,n) = h_c(m)h_r(n)$ with

$$h_c(m) = h_r(n) = [0.0094, -0.0707, 0.0694, 0.4900, 0.4900, 0.0694, 0.0707, 0.0094]^T \tag{10}$$

in order to generate, via Eq. (1), lower resolution versions of the three textures.

We employed next Algorithm 1 to produce the corresponding low resolution **NSHP** model coefficient sets $\hat{\mathbf{c}}_{-1}$ and generate a two-resolution reference data base. After training a similar network classifier using the model coefficient vectors from the images at low resolution level, we tested its generalization performance, which was also very good as shown in Table I; the above verifies that, assuming the resolution level a-priori known, the recursive classification scheme performs well at both resolution levels.

Test Case #2

In the second experiment we examined the proposed classification scheme when the resolution of the image to be classified is not a-priori known. We used textures p.1.1.1, p.1.1.3 and p.1.1.7 which are all of dimensions 512×512 pixels. Through the proposed procedure we extracted characteristic vectors at the higher resolution level and used them to derive corresponding vectors at two lower resolutions (images sizes of 512×512, 256×256 and 128×128 pixels respectively); Figures 2a and 2b show texture p.1.1.1, at resolutions $i=-1$ and $i=-2$ respectively considered in the experiment. As a consequence, we generated a three-resolution **NSHP** model reference vector data bank, including 8 vectors per resolution for the first image, 7 for the second and 9 for the third one; in total the data base consisted of 72 data vectors. Using this database we trained three separate

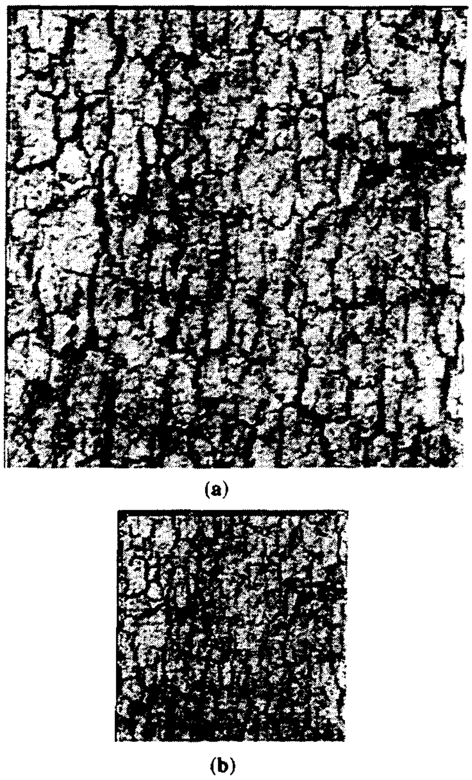

FIGURE 2 Multiresolution versions of texture p.1.1.1 of the Brodatz album.

networks, following the procedure described in Section IV, each of which had i) 6 nodes in its input layer, corresponding to the AR model coefficients ii) two hidden layers composed of 5 and 4 nodes respectively, iii) an output layer with three nodes corresponding to the three texture categories; each network classified the images at a single resolution level. After training, we presented the recursively estimated NSHP coefficient vectors from the images to the corresponding resolution network input and obtained the classification ratios shown in Table II.

We then presented all the above vectors to the networks without any information about their resolution level and tested the procedure proposed earlier in section IV of the paper. The classification ratios obtained for each texture class are also shown in Table II; these ratios are approximately equal to the ratios obtained at the lower resolution level per class, indicating that the proposed procedure is very effective for classifying textures without a-priori knowledge about their resolution level.

TABLE II Test Case #2: Classification of real life textured images at different (known and unknown) resolution levels

Image	*Success Rate*(%)			
	known resoltuion level			*unknown resolution level*
	Resolution (0)	*Resolution* (−1)	*Resolution* (−2)	
p.1.1.1	95.2%	93.7%	91.3%	90.9%
p.1.1.3	93.6%	91.9%	90.4%	90.1%
p.1.1.7	92.4%	90.8%	89.6%	89.3%

6. CONCLUSIONS

Multiresolution analysis of 2-D autoregressive image models has been combined with artificial neural networks for effective classification of textures. A scheme has been proposed which permits texture calssification to be performed irrespective of the resolution level of the presented image, based on the efficient generation of a multiresolution bank of artificial neural networks trained by appropriately extracted model parameter vectors. The presented results are quite promising. We are now examining the application of the approach to combined classification and segmentation problems, for image analysis and coding at low bit rates.

References

[1] Boutalis, Y., Kollias, S. and Carayannis, G. "A Fast Multichannel Approach to Adaptive Image Estimation", *IEEE Trans. On ASSP*, vol. ASSP-**37**(7), July 1989.

[2] Cheng, B. and Titterington, D. M. (1994). "Neural Networks: A Review from a Statistical Perspective", *Statistical Sci.*, **9**(1), pp. 2–54.

[3] Daubechies, I. *et al.* (1992). "Image Coding using the Wavelet Transform", *IEEE Transactions on Image Processing*, **1**, pp. 205–220.

[4] Delopoulos, A., Tirakis, A. and Kollias, S. "Invariant Image Classification using Cumulant based Neural Networks", *IEEE Trans. on Neural Networks*, **5**, pp. 392–408, May 1994.

[5] Delopoulos, A. and Rangoussi, M. Cumulants of a Multidimensional Process Observed at Rationally Related Resolutions", *Proc. of the 1997 Intl. Workshop on Sampling Theory and Applications*, pp. 205–210, Aveiro, Portugal, June 1997.

[6] Fahlman, S. and Lebiere, C. (1990). "The Cascade-Correlation Learning Architecture", *Tech Report CMU-CS-90-100*, Carnegie Mellon University.

[7] Haykin, S. (1994). "Neural Networks: A Comprehensive Formulation", *McMillan College Publishing Company, Inc.*, New York, NY 10022.

[8] Hush, D. R. and Horne, B. G. "Progress in Supervised Neural Networks", *IEEE Signal Processing Magazine*, pp. 8–39, January 1993.

[9] Kollias, S. and Anastassiou, D. "An Adaptive Least Squares Algorithm for the Efficient Training of Artificial Neural Networks", *IEEE Trans. on Circuits and Systems*, **36**, August 1989.

[10] Kollias, S. and Stafylopatis, A. (1993). "Parallel Implementations of the Backpropagation Learning Algorithm Based on Network Topology", in *Parallel Algorithms and Architectures for Digital Image Processing, Computer Vision and Neural Networks*, John Wiley and Sons.

[11] Kontoravdis, D., Stafylopatis, A. and Kollias, S. (1992). "Parallel Implementation of Structured Feedforward Neural Networks for Image Recongnition", *Intl. Journal of Neural Networks*, **2**, pp. 91–99.

[12] Mallat, S. "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation", *IEEE Trans. on PAMI*, **11**(7), July 1989.

[13] Rumelhart, D., Hinton, G. and Williams, R. (1986). "Learning internal representations by error propagation", In Rumelhart, D. and McClelland, J. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, MIT Press.

[14] Sukissian, L., Kollias, S. and Boutalis, Y. (1994). " Adaptive Classification of Textured Images Using Linear Prediction and Neural Networks", *Signal Processing*, **36**(2), pp. 209–232.

[15] Tirakis, A., Kaloyeras, D. and Kollias, S. (1992). "A Multimodel Approach to Image Classification Using Neural Networks", *Neural Network World*, **3**, pp. 269–287.

[16] Vaidyanathan, P. P. "Multirate Digital Filters, Filter Banks, Polyphase Networks and Applications: A Tutorial", *Proceedings of the IEEE*, **78**(1), Januray 1990.

[17] Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*, NewYork, Springer-Verlag.

[18] Wilkinson, G. G. "Open Questions in Neurocomputing for Earth Observation", *Proc. of 1st COMPARES Workshop*, York, UK, 17–19, July 1996.

[19] Woods, J. W. and O'Neil, S. D. "Subband coding of Images", *IEEE Trans. of Acoustics, Speech and Signal Processing*, **34**, pp. 1278–1288, October 1986.