Neural Recognition in a Pyramidal Structure

Virginio Cantoni and Alfredo Petrosino

***Abstract*—In recent years, there have been several proposals for the realization of models inspired to biological solutions for pattern recognition. In this work we propose a new approach, based on a hierarchical modular structure, to realize a system capable to learn by examples and recognize objects in digital images. The adopted techniques are based on multiresolution image analysis and neural networks. Performance on two different data sets and experimental timings on a single instruction multiple data (SIMD) machine are also reported.**

***Index Terms*—Multiresolution, pyramidal architecture, shape recognition, single instruction multiple data (SIMD) parallel machine, structured neural network.**

I. Introduction

SEVERAL approaches are pursued, in the last years, in the field of image processing and recognition based on pyramidal structures (see, for instance, [6], [8], [11], [21], [30], [32], and [33]). Pyramids are compact, multiscale representations that produce good textural features for landscape characterization, providing also support of efficient coarse-to-fine search. On the contrary, neural networks are suitable for learning ill-defined relationships from noisy examples, including relationships between different data types. Efforts toward developing neural network/pyramid techniques for recognition purposes arose as a way to handle problems of scaling with input dimensionality. Indeed, as reported by Geman *et al.* [15], "*important properties of patterns must be built-in or hard-wired, perhaps to be* . . . (tuned) . . . *later by experience, but not learned in any statistical meaningful way*".

The ideas reported in this and other related papers can be traced back to "biased models" for solving most meaningful practical problems, i.e., neural modeling is no longer focused exclusively on learning, but also on the identification of significant structure and weight constraints. This tries to reduce the number of free parameters to learn and consequently the complexity of learning. From the practical standpoint, this results in a less number of learning iterations, without strongly affecting the generalization capabilities of the network. Other possible ways consist of the application of statistical techniques, aimed to reduce the input dimensionality (and consequently the number of weights), like principal component analysis, independent component analysis, or pruning techniques to the weight space (see [19]).

Considerable interest has been gained by convolutional networks where each neuron is linked to a restricted number of neurons through weight sharing. This efficiently reduces the model complexity and the number of weights of the network, with a consequent advantage in learning complexity when high-dimensional images are to be presented directly to the network instead of using explicit feature extraction and data reduction. Moreover, networks with local topology can more effectively be mapped to a locally connected parallel computer than fully connected feedforward networks. For example, le Cun [25], [26] has proposed the use of receptive fields in feedforward nets, which result in special equality constraints on the weights. Fukushima [14] considered a combination of receptive fields and competitive learning to realize a multilayered network for recognizing handwritten digits, capable of extracting and classifying features. Applications of these models have been made in handwritten character and face recognition. Different convolutional networks, which have been demonstrated to discriminate more effectively between different classes of input, have been proposed in literature (see, for instance, [22], [27]). The use of more layers of receptive fields do not assure a successful classification of the features extracted in previous layers, while multilayer perceptrons perform better. Among others, Perantonis and Lisboa [29] have introduced a method for reducing the number of weights of a third-order network used for pattern recognition by imposing invariance under translation, rotation, and scaling.

From the standpoint of relationships between pyramids and networks, Bishof *et al.* [2] have compared neural networks and image pyramids, showing how a modified Hopfield network can be used for irregular decimation and examining the type of knowledge stored and the processing performed by pyramids and neural networks.

These and other papers report significant attempts to incorporate domain knowledge into neural networks (e.g., in terms of weight constraints). Since the incorporation of the domain knowledge turns out to be a complex task for neural networks, an alternative approach is to consider multiscale representations of input patterns, giving a regular structure to the network, without inserting *a priori* knowledge into the network. Moreover, but not less important, both neural networks and pyramids map well onto fast hardware for high-throughput applications. Some works have been developed along this research activity.

In particular, the pyramidal neural network reported by McQuoid [24] is arranged on four layers, one corresponding to the "retina" (or input image) and three layered neuronal areas; the neurons within each area are organized in columns of *ensembles*. The Area 1 is constructed from planes of ensembles, each aligned above one another, forming a number of ensemble columns. An ensemble consists of a 5×5 grid of neurons, each connected to the retina area directly below the corresponding neuron, with the same set of weights. Since there is one column for each input neuron belonging to an input field of 7×7 neurons, the weight sharing here allows the ensemble to realize

Manuscript received October 28, 1999; revised March 12, 2001.

V. Cantoni is with Dipartimento di Informatica e Sistemistica, Universitá di Pavia, 27100 Pavia, Italy (e-mail: cantoni@vision.unipv.it).

A. Petrosino was with INFM, Universitá di Salerno, 84081 Baronissi (Salerno), Italy. He is currently with CPS–National Research Council, Monte S. Angelo, Napoli, Italy; E-mail: alfredo@ventotene.dma.unina.it.

Publisher Item Identifier S 1045-9227(02)01806-4.

some translation tolerance. The same reasoning is adopted to construct the Area 2, where each ensemble consists of just one neuron and a 5×5 input field is used. The Area 3 consists of neurons (a set for each class of objects) whose response is a percentage of active neurons from Area 2 for each class; this area works as a monitor for verifying the presence of simultaneous patterns of activation and can be realized by any algorithm, not necessarily connesionist. This system is applied on images of size 24×24. According to the construction, the first area consists of 15 columns of ensembles (for a total of 735 ensembles and 18 375 neurons), the second area consists of 5×5 (the input field size) columns each constituted by 60 ensembles of just one neuron for a total of 1500 neurons, while the third area consists of seven neurons. The neurons in the overall system are 19 882, while the number of links is 948 735. The learning is *online*, therefore if shown an object, the network adapts to it in just one forward step and further presentation of the object in the same position does not change the system. The aim of the system proposed by McQuoid is to recognize the binary images of the digits "0"–"4," used either for training and for testing the network, without rotation and translation invariance, but in situations of simultaneous presence of the digits (like "34") and in presence of impulsive noise. Due to the adopted patterns and local online learning, the system behaves as an associative memory (it completely memorizes the patterns), but it needs a too large number of weights, when it has been demonstrated that less wide associative memories can accomplish the same task (see for instance [1] and the sections in [19] dedicated to this topic). Furthermore, to classify images of dimension wider than 24×24 pixels, as well as to include more recognition capabilities (e.g., translation, rotation, and scaling invariance), the number of neurons and weights greatly grows.

The pyramidal system reported by Honavar and Uhr [20] is a specifically tailored multilayer perceptron trained with the backpropagation learning rule, each layer embedded on a pyramidal level. In a layer, a cluster of nodes is connected to a small window in the layer below; the spatial resolution of the layers logarithmically decreases. The learning is iteratively applied to the network to generate a new node and the weights are changed by using the backpropagation rule (feedback-guided generative learning). The number of neurons needed for the recognition of 32×32 images is almost 12 285, while the number of modifiable weight links is almost 49 140. To recognize six objects, a learning set of four images and a test set of three images for each class have been used in the reported experiments; the classification accuracy achieved has been 100.0% after 20 training epochs, by generating 14 new neurons at each level. Specifically, each image represents a translated version of the original pattern of up to eight pixels apart from the image center. As usual, the greater the image dimension, the higher the number of weights the network needs. Moreover, the tests made are related only to a very restricted number of training and test patterns.

The approach reported in the present paper is mainly aimed to avoid an unconstrained growth of the freedom degrees of the system accordingly to the difficulty of discrimination, to the dimension of the image or to the increase of the pattern classes, enlarging either the category or the discrimination capabilities of the neural system. Our approach is based on a hierarchical system that executes a multiresolution analysis of the images provided as input and classifies through a structured neural network [12]. A peculiar feature of this kind of networks is their capabilities to avoid local minima in reaching the optimal weight configuration [16], [17]. In particular, it can be demonstrated in [16] that in case of linearly separable patterns, for a network with just one hidden layer, a number of outputs equal to the number of classes coded with exclusive coding, and full connections from the input to the hidden layer, subdivided into sublayers corresponding to the number of classes, the cost function is local minima free. The same result can be demonstrated under the "pyramidal hypothesis," i.e., the number of neurons at layer $l+1$ is less than the number of neurons at layer l.

The proposed approach is characterized by a lower number of pixels to manage, a fixed neural network structure and less neural-network weights. The system is shown to be able to successfully classify four objects with tolerance to rotation and translation, testing it over two different data sets, one constituted of similar patterns and the other one of dissimilar patterns.

This paper is organized as follows. In Section II, we shall give details of the proposed system, while Section III reports the classification results on the data sets, together with timings on a parallel single instruction multiple data (SIMD) machine.

II. The Proposed Approach

The system presented is arranged on L pyramidal levels numbered $0, 1, \ldots, L-1$, where L is equal to the log of the image size, i.e., the image size is $2^L \times 2^L$. The structure of the proposed technique is subdivided in two distinct parts (see Fig. 1): a multiresolution analysis and a classification phase. In the first part, the input image at level 0 $(N \times N)$ is processed in order to produce a *filtered* image of $(M \times M)$ pixels with $M < N$. This image is fed as input of the classification phase.

A. Multiresolution Analysis

A common characteristic of digital image is that neighboring pixels are highly correlated. The representation of the image directly in terms of the pixel value is therefore inefficient: most supported information is redundant. A wavelet-like technique is then adopted, based on the hierarchical discrete correlation (HDC) technique [3], in order to remove redundant information from image while producing a reduced image with the same "content;" in the following, we shall refer to this image as the *compressed image*.

Let $g_0(i,j)$ be the original image, the first step is to apply a low-pass filter on g_0 to obtain g_1, a "reduced" version of g_0, having both resolution and sample density decreased. The filtering is performed by a procedure equivalent to a convolution with a Gaussian "family" of local, symmetric weighting functions, so the sequence of images $g_0, g_1, \ldots, g_{p-1}$ is called Gaussian pyramid [4].

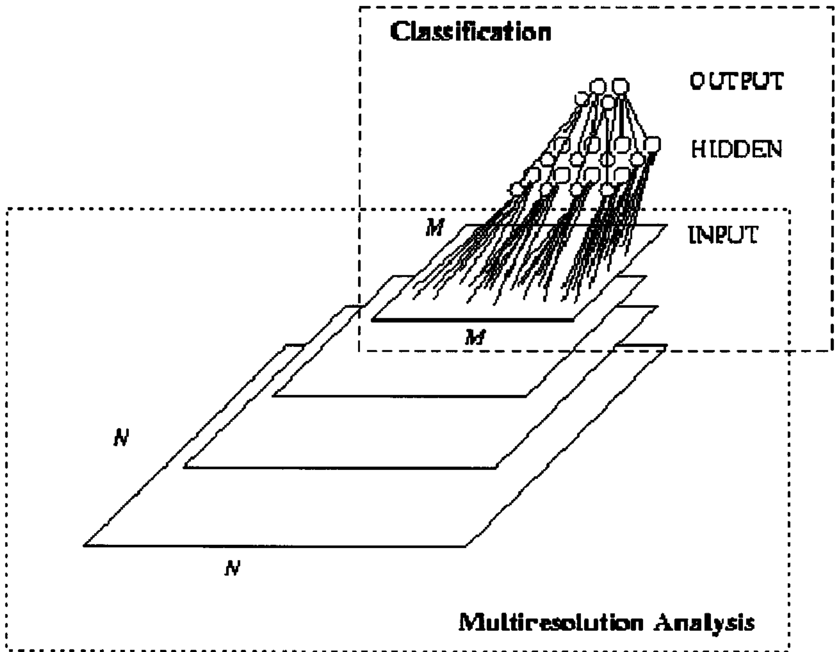


Fig. 1. The proposed approach for neural and pyramidal recognition.

Being $N \times N$ the size of g_0 and p the number of levels, then explicit relationship between successive levels g_l and g_{l-1}, with the coordinate (i,j) referred to g_0, is given by

$$g_l(i,j) = \sum_{m=-k}^{k} \sum_{n=-k}^{k} w(m,n) g_{l-1}(i - m2^{l-1}, j - n2^{l-1}) \tag{1}$$

with $0 < l \leq p-1$, $0 \leq i,j \leq N$. We remark that the $(2k+1) \times (2k+1)$ matrix w represents the *generating kernel* which limits the computational cost [4]; it is chosen to be *small* and *separable*, i.e., $w(m,n) = \hat{w}(m)\hat{w}(n)$, where $\hat{w}$ is *normalized* and *symmetric*

$$\sum_{m=-k}^{k} \hat{w}(m) = 1 \quad \hat{w}(i) = \hat{w}(-i) \quad 0 \leq i \leq k.$$

The image g_l is decimated (or subsampled) of a factor 2^l. We shall refer in the following to the application of (1) and the subsampling as the *REDUCE* function. In order to maintain an equal contribution from odd and even positions, the kernel must satisfy the following relationship:

$$\hat{w}(0) + 2\sum_m \hat{w}(2m) = 2\sum_n \hat{w}(2n-1) = 1/2$$
$$1 < 2m \leq k \quad 1 \leq 2n-1 \leq k$$

In particular, when $k = 2$ (i.e., the support of the kernel is 5×5), $\hat{w}(1) = \hat{w}(-1) = 0.25$, and $\hat{w}(2) = \hat{w}(-2) = 0.25 - \hat{w}(0)/2$. From the computational cost point of view, the construction of the complete Gaussian pyramid requires approximately ten operations per point at all pyramidal levels.

To congruently compare data resulting at two different levels at the higher detail, an interpolated image $g_{l,t}$ by a factor 2^t of the image g_l must be defined. $g_{l,t}$ turns to be of of dimension $(2^tN+1) \times (2^tN+1)$ if the original image g_l has dimension $(N+1) \times (N+1)$. This can be implemented [5] by padding with "zeros," i.e., inserting new even rows and columns with null samples, and convolving the resulting image with a suitable filter w'. Explicitly, for $1 \leq r \leq t$

$$g_{l,r}(i,j) = 4k^2 \sum_{m=-k}^{k} \sum_{n=-k}^{k} w'(m,n) \times g_{l,r-1}\left(\left\lfloor \frac{i-m}{2} \right\rfloor, \left\lfloor \frac{j-n}{2} \right\rfloor\right) \tag{2}$$

with $0 \leq l < p-1$, $0 \leq i,j \leq N$ and i and j even, while $g_{l,r}(i,j) = 0$ otherwise. w' (the *expanding kernel*) is selected on the basis of the same constraints of w. The factor $4k^2$ is introduced to compensate for the fact that $(2k)^2$ out of $(2k+1)^2$ samples in $g_{l,r}$ are zeros. Note that $g_{l,0} = g_l$ and $g_{0,0}, g_{1,1}, \ldots, g_{p-1,p-1}$ will have the same sizes of the original image $I = g_0$. In the following, we shall refer therefore to (2) performing an expansion by a factor 2^t to be the operation *EXPAND*, i.e., $g_{l,t} = EXPAND(g_l, t)$. The error images $L_0, L_1, \ldots, L_{p-1}$, known as *Laplacian Pyramid*, are characterized by

$$L_{p-1} = g_{p-1}$$
$$L_l = g_l - EXPAND(g_{l+1}, 1) \quad 0 \leq l < p-1 \tag{3}$$

They closely resemble the well-known Laplacian operator δ^2, introduced by Marr and Hildreth [23] for edge detection purposes, following a multiresolution lateral inhibition approach. This representation is characterized by the preservation of the *spatial localization* and *spatial frequency* information since the operators involved are local and the Gaussian Pyramid reduces the frequency content implementing a local low-pass filtering. The resulting operators are yet local and are localized in the spatial frequency domain by reducing one octave (for $k = 2$) both the frequency center of the bandpass and the bandwidth. The solution we adopt to construct the Laplacian pyramid is known to be the RE (reduce and expand) solution [4] and has the property of completeness in image recovering. Due to the structure of the proposed system, the only image of the Laplacian pyramid we need is L_{m-1} with $m = \log M$, M being chosen as a prefixed parameter highly correlated to the information content of the images provided to the system. In order to avoid an unconstrained growth of the freedom degrees of the system accordingly to the difficulty of discrimination, to the dimension of the image or to the increase of the pattern classes, the value of M can be chosen fixed, increasing the levels of the Laplacian pyramid. As instance, if for an image size of 64 × 64 the network has 8 × 8 inputs, with three pyramidal levels, the same network structure can be applied to the same number of inputs, but with five pyramidal levels, when the image size grows to 256 × 256.

The multiresolution analysis we adopted can be easily described as two consecutive processing steps.

1) Letting I_0 be the image on level 0, compute $I_i = REDUCE(I_{i-1})$ at level i for $1 \leq i \leq m$, where $m = \log(N/M) + 1$.
2) Compute the input for the neural classifier $L_{m-1} = I_{m-1} - EXPAND(I_m, 1)$.

The output is the detailed image representing the difference between the coarsest and second coarsest approximation images. This coarse detailed image is sensible for shape analysis because much of the noise if filtered out, and this image captures important high frequency information such as edges.

B. The Classification Phase

The proposed approach addresses the problem of supervised classification, under the assumption that the required image-processing steps have already been performed. In particular, our approach is based on the use of structured neural networks [12]. Differently from "unstructured" neural networks, as for instance the fully connected layered, in "structured" networks the output of each hidden-layer fed just one neuron of the next layer. Even if the input neuron can be connected to more neurons of the first layer, the structure of the network resembles a tree-like network. In our approach, the network embedded in layers numbered $m+1, m+2$ consists of single-layered networks of kind 4-1 in number equal to that of neurons in the same layer. Specifically, the classification system is a neural network with three layers of neurons: 1) the *input layer* at level m; 2) the *hidden layer* at level $m+1$; and 3) the *output layer* at level $m+2$. Each neuron belonging to the $(m+1)$th layer is connected to an area T of 4 × 4 neurons of the mth layer and 2 × 2 at layer $m+1$. The *activation function* for every neurons is a sum-sigmoidal one, i.e., $f(\mathrm{net}_j) = 1/(1+e^{-\mathrm{net}_j})$, net_j being a weighted sum of inputs to neuron j, i.e., $\mathrm{net}_j = \sum_{i \in T} w_{ij} o_i$. The learning rule we used is a backpropagation one [31].

Let us call *presentation* a cycle forward/backward of learning for a single pattern, while an *epoch* means the presentation of all the patterns belonging to a set. During the learning phase there are several epochs that may be interrupted by epochs of test on patterns belonging to a validation set.

Both *batch* and *on-line* strategies for weights updating have been tested. According to the first, the weights are updated after each epoch, i.e., the error occurred on each presentation is accumulated during the epoch; in the on-line strategy, the weights are updated after each presentation.

To measure the ratio of patterns uncorrectly recognized, the average mean squared error (AMSE) has been adopted

$$\mathrm{AMSE} = 1/(CN_p) \sum_{j=1}^{N_p} \sum_{i=1}^{C} \left(t_i^j - o_i^j\right)^2$$

where C is the number of classes, N_p the number of training patterns, t_i^j the target response of the neuron i to the jth pattern, and o_i^j the actual response of the neuron i to the presentation of the jth pattern. We recall that t_i^j is equal to one when the jth pattern belongs to the class i and zero otherwise.

The end of the training phase is set dependly by the average mean square error (AMSE), or by the percentage of recognition on test and/or training. The learning goes ahead until the value of AMSE is not less than of the minimum acceptable error γ. After a learning phase a classification phase starts. During the learning phase the system gets a skill to distinguish the input images in the test phase.

It can be verified that the magnitude of the algorithm complexity for preprocessing and learning is

$$O(\log(N/M) + M^2 + C)$$

where $N \times N$ is the image size, $M \times M$ is the dimension of the input supplied to the neural network and C the number of classes to be recognized. The communication bandwidth is linear with M.

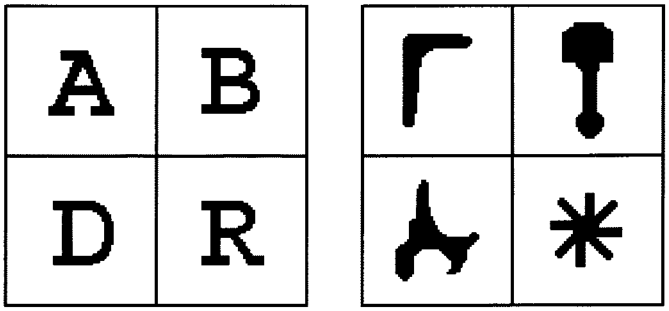
III. Experimental Results

The experiments were performed on the patterns depicted in Fig. 2(a) and (b), using binary images with $N = 64$. Furthermore, choosing $m = 3$, the neural network is structured as follows:

- **level 3**, 8 × 8 neurons, i.e., 64 input neurons.
- **level 4**, 4 × 4 neurons and 16 links per neuron.
- **level 5**, 2 × 2 neurons and four links per neuron, i.e., $C = 4$ output neurons.

Thus the overall number of neurons is 84, while the number of links is 272.

The training and the test sets consist of images representing rotated and shifted patterns. The allowed shifts were five pixels by five pixels to the up, down, left, right and diagonal, for a total number of 25 positions, while the allowed rotations were 5°by 5° for a total of 72 rotations. To avoid confusion, we distinguish even shifts and odd shifts, as it is shown in Fig. 5. The total number of generated images is 7200. Usually, the establishment of initial parameters, like weights and learning rate, is a difficult task due to the fact that a wrong choice of initial weights may imply an high number of training epochs, while a wrong choice of η may prevent the exploration of zone in the energy landscape, where it may occur a minimum error. Thus, we have chosen random weights in [0, 0.1] and tested ten learning sessions, by choosing the best results among them; the learning rate was always initialized to $\eta = 0.1$.



(a) (b)

Fig. 2. (a) The "letter" pattern set and (b) the "tools" pattern set.

A. Experiment 1

The first experiment was performed on a training set consisting by all the rotations of 10° by 10°, starting on 0° and a test set consisting by all the rotations 10° by 10°, starting on 5°, for a total of 144 images for both training and test sets. We adopt two sets of patterns for testing the ability of our system to recognize patterns in different conditions. One set of patterns is depicted in Fig. 2(b), representing "tools" which have been widely used in literature for recognition testing aims. They are characterized by a very dissimilar shape which should turn out in a very good recognition performance.

On the contrary, we adopt the set of patterns depicted in Fig. 2(a), to test our system also in condition of similarity among patterns according to their geometrical properties. As instance, the Euler number [18] of letters A, R, and D is 1 and similar considerations can be made measuring the geometrical properties by means of Fourier or Moment descriptors. This second pattern set is considered not allowing strong classification performance.

In these experiments we have used an online strategy and a sigmoid activation function for each level of the neural network. When applied to the "tools" pattern set the learning was stopped after 26 learning epochs, with 100% of recognition percentage on training and 93.75% on test, while it was stopped after only seven epochs with 98.61% of recognition on training and 99.31% on test, when we provided the "letter" pattern set.

To give an idea about how the learning works in the structured pyramidal network, the Fig. 3(a) depicts the changes of the AMSE with respect to the number of training by using the "letter" data set, while Fig. 3(b) shows how the correct classification percentage on training data sets changes during the learning session on the same data set.

B. Experiment 2

The neural network used in the Experiment 1 uses a total of 272 weights. Obviously, the training set adopted in Experiment 1 of 144 patterns, simply memorizes each pattern. This fact explains the drop-off in performance between the training and testing images for the "tools" patterns, while the performance drop-off does not occur in the "letter" patterns because the AMSE threshold was set higher, and the neural network was not allowed to completely "memorize" the patterns.

The second experiment is more complex. Indeed, the training set consists, for all classes, of all 10° by 10° rotation, starting for 0° of all odd shifts, and all 10° by 10° degree, starting from 5°, of all even shifts (Fig. 5). The number of all resulting training images is 3744, while the test set consists of 3456 images.

After setting the initial parameters, the learning phase was performed with online strategy and sigmoidal activation function, on both levels 4 and 5. When applied to the "tools" pattern set the learning was stopped after 319 epochs, providing a correct classification percentage of 90.465% on training, i.e., 3387 patterns recognized over 3744. The *averaged mean square error* (AMSE) was 0.097 716 9 and its value was the condition for the learning termination, by fixing the maximum acceptable AMSE value to be 0.1. In the test phase there was a percentage 87.73% of recognized patterns, i.e., 3032 recognized patterns over 3456. When we provided the "letter" pattern set, the training was stopped after 256 epochs resulting in 89.1% of correct classification percentage on the training set and 88.7% on the test set. In Table I the parameter settings and the correct classification percentages are reported for all the experiments made.

Fig. 4(a) depicts the changes of the AMSE with respect to the number of training by using the "letter" data set (the learning stops just before the peak of AMSE after nearly 250 epochs), while Fig. 4(b) shows how the correct classification percentage on training data sets changes during the learning session on the same data set. In Table II, the classification accuracies achieved by the Honavar and Uhr system on the same data sets are shown. Specifically, these accuracies correspond to a system structured with four layers (two hidden layers), such that the first layer consists of 64×64 input neurons, the second layer of 16×16 hidden neurons, the third layer of 8×8 hidden neurons, the fourth layer of 2×2 output neurons. It should be remarked that on the same data and the same initial parameter settings, the application of the multiresolution analysis as described in the present paper has allowed to achieve better classification accuracies. As instance, let us consider the application of a network with structure 64-32-16-4 to 3744 training images and 3456 test images of size 8×8 taken from the "letter" data set obtained after the application of three layers of the multiresolution analysis. The classification accuracies have been of 99.97% for training and 98.81% for test, outperforming the result obtained by applying the Honavar-Uhr system and shown in $2-b$ of Table II.

The effect of the use of the Laplacian pyramid can be more appreciated if one considers that by using the Gaussian pyramid more epochs are needed to reach comparable results. For instance, let us consider the experiment denoted by $2-b$ in Table I on the "letter" patterns, and letting unchanged the classification settings, we applied the the Gaussian Pyramid level I_{m-1} as input instead of the Laplacian Pyramid level L_{m-1}. After 2856 epochs and the same minimum acceptable error γ used in the experiment $2-b$, the correct classification percentages have been 91.98% on the training set and 86.25% on the test set. The greater number of training epochs and worst classification percentage on test is due to the inefficient feature detection pro-

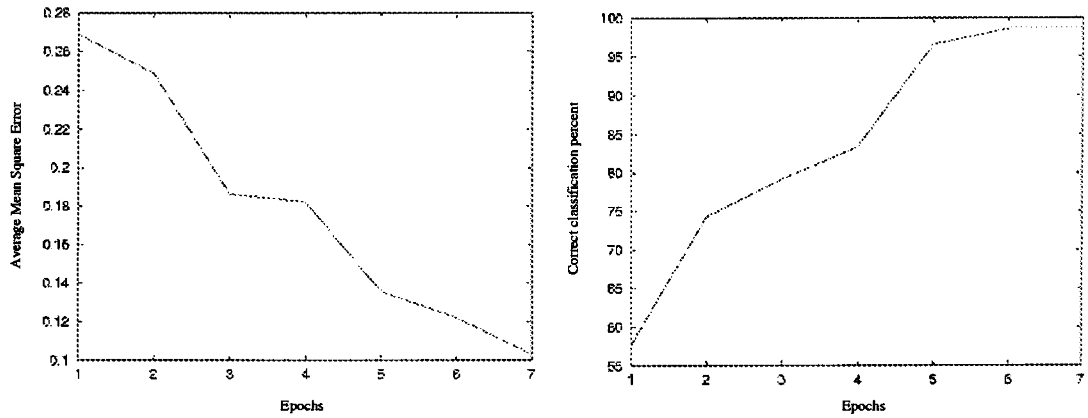


Fig. 3. (a) Average mean square error and (b) correct classification percentage during the learning session of the Experiment 1 over the "letter" data set.

TABLE I
CLASSIFICATION ACCURACIES (IN PERCENT) AND SETTINGS FOR "TOOLS" (1-a AND 2-a) AND "LETTER" (1-b AND 2-b) DATA SETS. AMSE STANDS FOR AVERAGE MEAN SQUARE ERROR

Experiment	Training examples	Test examples	Training epochs	AMSE	Classification on training	Classification on test
1-a	144	144	26	0.018993	100.0 %	93.75%
1-b	144	144	7	0.092965	98.61 %	99.31%
2-a	3744	3456	319	0.097716	90.46 %	87.73%
2-b	3744	3456	256	0.091234	89.1 %	88.7 %

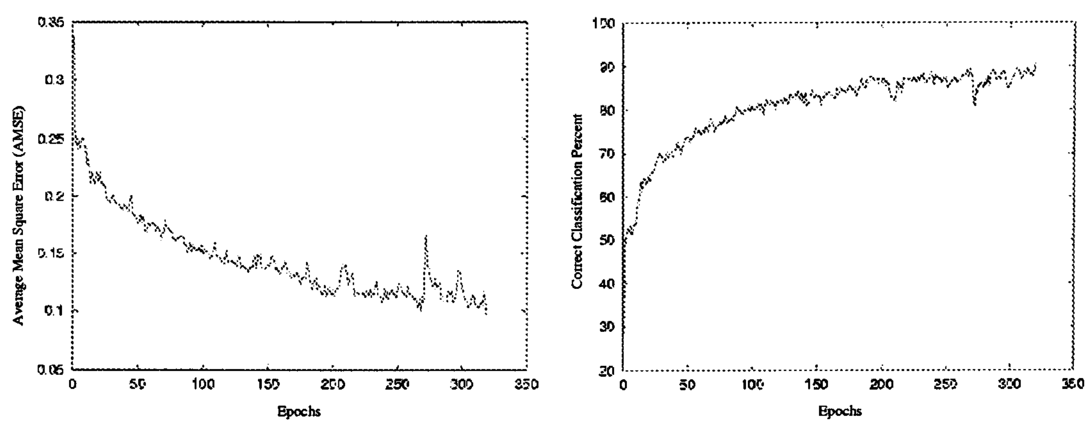


Fig. 4. (a) Average mean square error and (b) correct classification percentage during the learning session of the Experiment 2 over the "letter" data set.

vided by the Gaussian pyramid, with respect to the Laplacian pyramid.

To test that the system is capable to constrain the growth of the freedom degrees to adapt, we have applied the same network structure to the problem of recognizing images of "aircrafts" of dimension 256×256, subdivided in training and test data sets as described above for the "letter" and "tools" data sets. By applying five Laplacian pyramid levels to the images, we achieved a correct classification percentage of recognition over the test data sets of nearly 96%.

C. Parallel Implementation and Timings

The neural classification system has been mapped onto a pyramidal architecture as depicted in Fig. 6(a). The architecture is characterized by the fact that the number of processors on each level is reduced by a factor of $1/4$ from level i to level $i+1$ with $0 \leq i < L-1$, starting from the pyramid base (level 0). The interlayer and intralayer connectivity of each processing element (PE) follows the interconnection scheme of Fig. 6(b). Details about the architecture can be found in [6], [7].

TABLE II
CLASSIFICATION ACCURACIES (IN PERCENT) AND SETTINGS FOR "TOOLS" (1-a AND 2-a) AND "LETTER" (1-b AND 2-b) DATA SETS ACHIEVED BY THE HONAVAR AND UHR SYSTEM STRUCTURED WITH FOUR LEVELS

Experiment	Training examples	Test examples	Classification on training	Classification on test
1-a	144	144	98.7 %	90.3 %
1-b	144	144	97.4 %	93.1 %
2-a	3744	3456	89.3 %	84.6 %
2-b	3744	3456	88.2 %	83.4 %

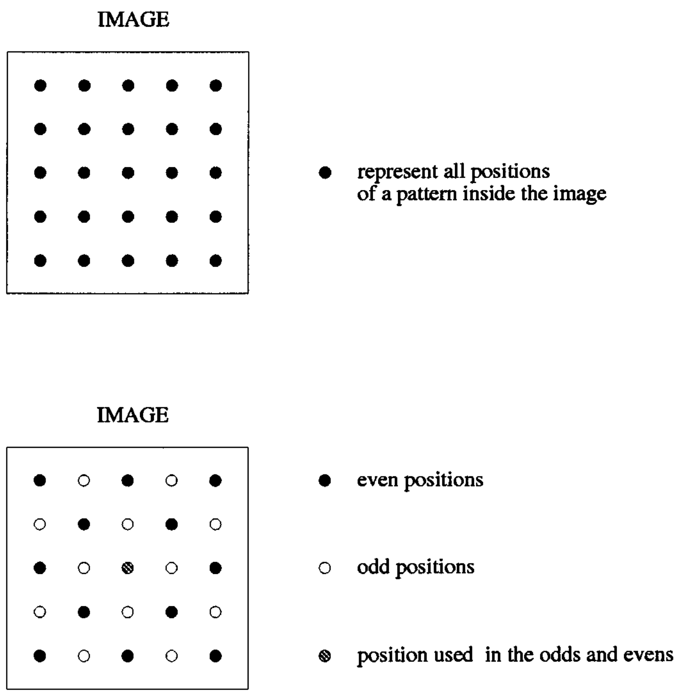


Fig. 5. The positions in an image adopted to construct training and test data sets.

Without loss of generality, we consider a one-to-one correspondence between the processors of the base and the pixels of the input image, i.e., we shall consider $2^L \times 2^L$ processors at level 0. From the processor allocation standpoint, each processor gets in its own local memory the weights of the links between the neurons allocated to it and those belonging to the 2×2 subset of the previous layer. Based on the pyramidal interconnection architecture, each processor get inputs and send outputs through *broadcasting* periods. Moreover, the pixels at level m are fed to the neural network, while the task of the processor at the apex of the pyramid is to get the outputs of processors at level $m + 1$ and put out the result. The parallel classification algorithm is sketched in more details in Fig. 7.

The proposed system has been implemented on a single instruction multiple data (SIMD) architecture with an *hypercube* interconnection structure, i.e., a TMC Connection Machine 200 [10], [9] configurated with 4096 1-bit processors. We adopted

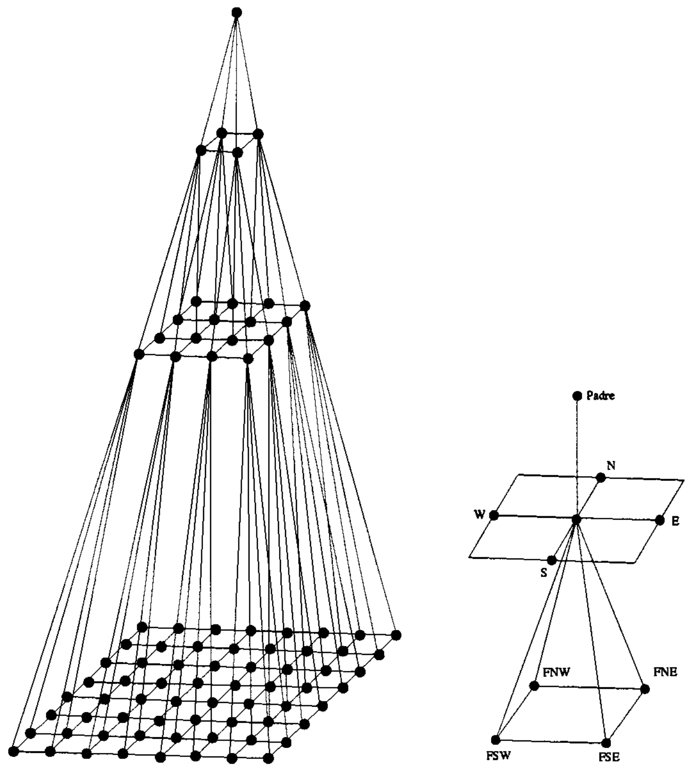


Fig. 6. (a) The pyramidal architecture and (b) the interlayer and intralayer connectivity of the processors.

the mapping scheme reported in [34] to embed the pyramidal interconnection structure in the hypercube. The entire system, i.e., the multiresolution analysis and the classification algorithm performed the quoted test phase with 3744 patterns in almost 161 milliseconds.

IV. CONCLUSION

This paper presents a parallel pyramidal technique for 2-D object recognition based on neural networks. Other algorithms based on neural networks, contour decomposition and attentional mechanisms have been proposed in literature to solve this problem. The method discussed in this paper is characterized by the combined use of structured neural networks and an highly efficient feature extraction mechanism, which leads to a lower number of weights and a fixed neural network structure. In particular, high granularity features which characterize the tone of the objects cannot be recognized due to the presence of noise. The use of the Laplacian pyramid sieves out noise and high correlation among data. The images at the upper levels of the pyramid result from the filtering activity, which leads to a cheaper recognition. However, in some cases important details are smoothed away by the filtering process and cannot be recovered at the higher levels of resolution; this results in a poor classification percentage on some patterns. However, in our experiments we observed that the online strategy of neural learning achieves the best results also in noisy cases. The reported experiments on different data sets demonstrate that our approach is at least reasonable for treating classes of data. An interesting issue

The Classification Algorithm.

1. **Randomize the initial set of weights on levels m and $m+1$.**
2. **Forward Phase**
   a) **get input values x_i, for each neuron i, from processor sons.**
   b) **send x_i to the neighboring processors.**
   c) **compute $net_i = \sum_j w_{ji} x_j$ and $o_i = f(net_i)$, $1 \le i \le n, 1 \le j \le N$.**
3. **Backward Phase**
   a) **compute**

$$\delta_i = \begin{cases} f'(net_i)(t_i - o_i) & \text{if level m+1} \\ f'(net_i)p_i & \text{if level m} \end{cases} \qquad 1 \le i \le n \quad .$$

   b) **compute $w_{ji} = w_{ji} + \eta \delta_i x_j$, $1 \le i \le n, 1 \le j \le N$.**
   c) **compute $s_j = \delta_i w_{ji}$, $1 \le i \le n, 1 \le j \le N$.**
   d) **send $s_j, 1 \le j \le N$ to the neighboring processors.**
   e) **compute $p_j = \sum_{i \in S} s_i, \quad 1 \le j \le N$, where S is the set of the index neurons on the some level of neuron j, which are connected to the some neuron of previous level.**
   f) **send $p_j, 1 \le j \le N$ to the processor sons.**
4. **If the error $AMSE > \gamma$ go to step 2.**

Fig. 7. The classification algorithm mapped on the pyramidal structure.

is to extend our results to models dealing with other *wavelet* functions, different from the Laplacian Pyramid.

REFERENCES

[1] E. R. Caianiello, A. Petrosino, R. Tagliaferri, and A. De Benedictis, "Neural associative memories with minimum connectivity," *Neural Networks*, vol. 5, no. 3, pp. 433–439, 1992.

[2] H. Bishof, "Neural networks and image pyramids," in *Proc. Pattern Recognition*, vol. 62, H. Bishof and W. Kropatsch, Eds, 1992, pp. 249–260.

[3] P. J. Burt, "Fast filter transforms for image processing," *Comput. Graphics Image Processing*, vol. 16, pp. 20–51, 1981.

[4] P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Trans. Commun.*, vol. COM-31, pp. 532–540, 1983.

[5] P. J. Burt, C. H. Anderson, J. O. Sinninger, and G. van der Wal, "A pipeline pyramid machine," in *Pyramidal Systems for Computer Vision*, V. Cantoni and S. Levialdi, Eds. Berlin, Germany: Springer-Verlag, 1986, pp. 133–152.

[6] V. Cantoni and S. Levialdi, *Pyramidal Systems for Computer Vision*. Berlin, Germany: Springer-Verlag, 1987.

[7] V. Cantoni, V. Di Gesù, M. Ferretti, and S. Levialdi, "The PAPIA system," *J. VLSI Signal Processing*, vol. 2, pp. 195–217, 1991.

[8] V. Cantoni and M. Ferretti, *Pyramidal Architectures for Computer Vision*. New York: Plenum, 1994.

[9] *Thinking Machines Corporation*, Nov. 1990. Connection Machine Model CM-200 User's Guide, Version 6.0.

[10] *Thinking Machines Corporation*, Nov. 1990. C* Programming Guide, Version 6.0.

[11] C. R. Dyer, "Multiscale image understanding," in *Parallel Computer Vision*, L. Uhr, Ed. New York: Academic, 1987.

[12] O. K. Ersoy and D. Hong, "Parallel, self-organizing, hierarchical neural networks," *IEEE Trans. Neural Networks*, vol. 1, pp. 167–178, 1990.

[13] S. E. Fahlman, "Faster-learning variations on backpropagation: An empirical study," in *Proc. Connectionist Models Summer School*, June 1988.

[14] K. Fukushima, "Neocognitron: A hierarchical neural network capable of visual pattern recognition," *Neural Networks*, vol. 1, pp. 119–130, 1998.

[15] S. Geman, E. Bienenstock, and R. Dourstat, "Neural networks and the bias/variance dilemma," *Neural Computation*, vol. 4, no. 1, pp. 1–58, 1992.

[16] M. Gori and A. Tesi, "On the problem of local minima in backpropagation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, pp. 76–86, 1992.

[17] M. Bianchini and M. Gori, "Optimal learning in artificial neural networks: A review of theoretical results," *Neurocomput.*, vol. 13, no. 5, pp. 313–346, Oct. 1996.

[18] R. C. Gonzalez and P. Wintz, *Digital Image Processing*. More , MA: Addison-Wesley, 1987.

[19] S. Haykin, *Neural Networks: A Comprehensive Foundation*. New York: Macmillan/IEEE Press, 1994.

[20] V. Honavar and L. Uhr, "Brain-structured connectionist networks that perceive and learn," *Connection Sci.*, vol. 1, no. 2, pp. 139–159, 1989.

[21] R. Hummel, "The scale-space formulation of pyramid data structures," in *Parallel Computer Vision*, L. Uhr, Ed. New York: Academic, 1987.

[22] D. R. Lovell, T. Downs, and A.-C. Tsoi, "Evaluation of the neocognitron," *IEEE Trans. Neural Networks*, vol. 8, pp. 1098–1105, 1997.

[23] D. Marr and E. C. Hildreth, "Theory of edge detection," in *Proc. Roy. Soc. London*, vol. B-207, 1980, pp. 187–217.

[24] M. R. J. McQuoid, "Neural ensembles: Simultaneous recognition of multiple 2-D visual objects," *Neural Networks*, vol. 6, pp. 907–917, 1993.

[25] Y. le Cun *et al.*, "Generalization and network design strategies," in *Connectionism in Perspective*, R. Pfeifer *et al.*, Eds. Amsterdam, The Netherlands, 1989, pp. 143–155.

[26] *The Handbook of Brain Science and Neural Networks*, M. Arbib, Ed., MIT Press, Cambridge, MA, 1995, pp. 225–258.

[27] C. Neubauer, "Evaluation of convolutional neural networks for visual recognition," *IEEE Trans. Neural Networks*, vol. 9, pp. 685–696, 1998.

[28] Y. Pao, *Adaptive Pattern Recognition and Neural Networks*: Addison-Wesley, 1992.

[29] S. J. Perantonis and P. J. G. Lisboa, "Translation, rotation and scale invariant pattern recognition by high-order neural networks and moment classifiers," *IEEE Trans. Neural Networks*, vol. 3, pp. 241–251, 1992.

[30] A. Rosenfeld, *Multiresolution Image Processing and Analysis*: Springer-Verlag, 1984.
[31] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distribuited Processing: Exploration in the the Microstructure of Cognition*, 1986, ch. 8.
[32] L. Uhr, "Layered recognition cone networks that preprocess, classify and describe," *IEEE Trans. Comput.*, vol. C-21, pp. 758–768, 1972.
[33] ——, "Pyramid multi-computer structures and augmented pyramids," in *Computing Structures for Image Processing*, M. Duff, Ed. New York: Academic, 1983.
[34] S. G. Ziavras and M. A. Siddiqui, "Pyramid mappings onto hypercubes for computer vision: Connection machine comparative study," *Concurrency: Practice and Experience*, vol. 5, no. 6, pp. 471–489, 1993.