Classification of Noisy Signals Using Fuzzy ARTMAP Neural Networks

Dimitrios Charalampidis, *Student Member, IEEE*, Takis Kasparis, *Member, IEEE*, and Michael Georgiopoulos, *Member, IEEE*

***Abstract*—This paper describes an approach to classification of noisy signals using a technique based on the fuzzy ARTMAP neural network (FAMNN). The proposed method is a modification of the testing phase of the fuzzy ARTMAP that exhibits superior generalization performance compared to the generalization performance of the standard fuzzy ARTMAP in the presence of noise. An application to textured grayscale image segmentation is presented. The superiority of the proposed modification over the standard fuzzy ARTMAP is established by a number of experiments using various texture sets, feature vectors and noise types. The texture sets include various aerial photos and also samples obtained from the Brodatz album. Furthermore, the classification performance of the standard and the modified fuzzy ARTMAP is compared for different network sizes. Classification results that illustrate the performance of the modified algorithm and the FAMNN are presented.**

***Index Terms*—Classification, energy, fractal dimension, fuzzy ARTMAP, noise, segmentation, texture.**

I. Introduction

DURING the past few years, neural networks (NNs) have been extensively used for classification and pattern recognition tasks. Examples of such neural network models include the backpropagation NN [1], radial basis function NN [2] and the adaptive resonance theory (ART) NN. The family of ART NN includes the fuzzy ART mapping (ARTMAP) NN (FAMNN) [3], the Fuzzy min–max NN [4], the laterally primed adaptive resonance theory (LAPART) [5], the evidence integration for dynamic predictive mapping (ART-EMAP) [6], and the Gaussian ARTMAP [7]. Like other members of the ARTMAP family, FAMNN has certain advantages over many other NN models and is especially suited to classification problems. One advantage is that FAMNN is faster to train than other neural networks due to the small number of training epochs required by the network to "learn" the input data. FAMNN is considered fast even among members of the ARTMAP family due to the computationally "cheap" mapping between inputs and outputs. Also, the classification results of FAMNN are easily interpretable. Furthermore, compared to standard nearest neighbor techniques which are also commonly used, FAMNN requires less memory since it uses a compressed representation of the data, and for the same reason FAMNN requires less classification time. The existence of memory reduction algorithms [27] does not diminish this advantage of the FAMNN since these algorithms can be used for both nearest neighbor techniques and the FAMNN.

Applications of NN include classification of one–, two–, or three–dimensional (1-D, 2-D, or 3-D) signals. In this work, we introduce a variation of FAMNN that exhibits superior generalization performance compared to that of the standard FAMNN when the signals are corrupted by additive noise. The variation of FAMNN, which is named FAMNN-m, adjusts the regions of dominance of each class to improve the generalization performance, assuming that information about the noise distribution is not available. To demonstrate the generalization improvement of the FAMNN-m over FAMNN, we consider as an example the classification of textured images, which are a case of 2-D signals. However, our approach can be easily extended to 1-D or 3-D signals. Texture is a main characteristic of the surface of an object. In the case of an image, it defines the spatial relationship between the grayscale values of the pixels in a region of the image. For the purpose of classification, textures must be described by parameters, usually denoted as features. The features that are selected must be sufficient to characterize the texture. Examples of features that have been used in the past include Gabor energy [8]–[10], Fourier transform energy [11], second-order statistical features [12], wavelet features [13], [14], and fractal dimension (FD) [15]–[17].

The proposed modification of FAMNN is especially suited for applications where it is required that the feature set captures only the shape characteristics of the signal and not the actual amplitude or average value. Case examples may include encephalographs and electrocardiographs used for medical diagnosis, classification and recognition of speech signals, as well as classification and segmentation of textured images and satellite photos. In such examples, correct classification independent of illumination and volume level (signal amplitude) is important. For instance, if the signal is transmitted through a communication channel, different attenuation may be introduced at different times. In the case of textures, the images may have been obtained using a camera in a badly illuminated environment. This type of distortion is sometimes termed as multiplicative noise. Feature sets that are fairly insensitive to linear transformations and multiplicative noise are desired. Two such feature sets that we use in this work are based on fractal characteristics and normalized energy (NE) measures. In [18]–[20] the independence of fractal dimension (FD) to linear transformations of the signal and to multiplicative noise is illustrated. The energy features that we also use maintain similar independence through proper normalization.

The signals used for training are the best quality data available. The test signals are subjected to further degradation. For

Manuscript received May 4, 2000; revised December 19, 2000.

The authors are with the School of Electrical Engineering and Computer Science, University of Central Florida, Orlando, FL 32816 USA.

Publisher Item Identifier S 1045-9227(01)07564-6.

1045–9227/01$10.00 © 2001 IEEE

instance, one of the two texture sets used for classification is a natural set obtained from University of Berkeley's Kite aerial photography, and it is not a noise free set. Some examples where the assumption "relatively good quality data is available" holds are:

— *Communication systems*: Features are extracted from source signals in order to train the NN. Then, the classifier is placed at the receiver of a classification system and the arriving signals are tested. Therefore, the signals that are available at the receiver site will be more degraded than the signals used for training.

— *Character recognition*: The network is trained using handwritten samples carefully obtained. On the other hand, the test characters may exist on copies, fax, or mail letters, for example, where the quality is not as good as the quality of the training data.

— *Speech recognition*: Human voice is recorded using good-quality recorders, while the test data may have been obtained using worse quality means.

— *Aerial photo classification*: Aerial photos are obtained on a relatively clear day. These photos are used for training the neural network. Testing needs to be performed independently of the weather conditions.

In order to illustrate the improved performance of FAMNN-m over FAMNN, in the presence of noise, we focus our attention on the texture classification problem. For that reason, we use features that are rotation invariant in order to classify similar textures that are viewed from different angles. Also, the features are extracted from small image windows so that identification of different textural regions in an image is more accurate. There is a vast literature on texture segmentation and classification. Among the various feature extraction techniques, Gabor filter banks and energy has been used [8]–[10]. These features are not rotation invariant since they are extracted using directional filters. In Murino *et al.* [21], third-order statistical features are used where it is demonstrated that they exhibit a significant tolerance to white noise. Chaunduri *et al.* [15] use FD-based features. Feature invariance to image transformations is a desirable property and it is often overlooked in the literature.

The modification that is proposed in this paper can be applied to a larger family of classification techniques, including other members of the ART family and k-nearest neighbor techniques. It is easy to show that if the FAMNN has "point" nodes as many as the number of training patterns, it is a 1-nearest neighbor classifier in terms of the L_1 distance. In this paper we present results that show that the modification of FAMNN is effective for different sizes of the network including the case where the FAMNN behaves as a 1-nearest neighbor classifier.

The paper is organized as follows. In Section II, we present an overview of the Fuzzy ARTMAP architecture. In Section III, we discuss the proposed classification technique (FAMNN-m). In Section IV we review the experimental results that compare the performance of FAMNN-m and FAMNN in classifying textures. Finally, in Section V we provide some closing remarks.

II. Fuzzy ARTMAP Neural Network

The FAMNN [3] consists of two fuzzy ART modules designated as ART_a and ART_b, as well as an inter-ART module as shown in Fig. 1. The inputs are presented at the ART_a module and the corresponding outputs at the ART_b module. The inter-ART module determines whether the mapping between the inputs and the outputs is the correct one. In the case where the mapping between inputs and outputs is many to one, the FAMNN operations can be completely described by referring only to the ART_a module. For this reason only the ART_a module is described here.

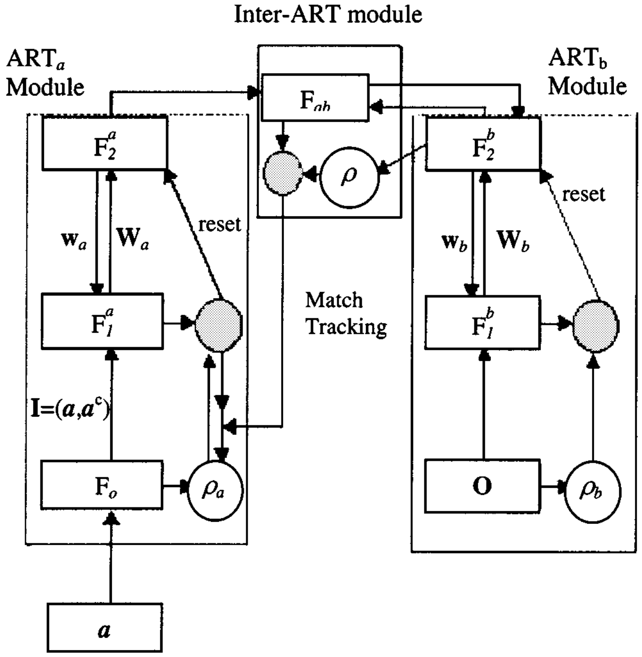


Fig. 1. A block diagram of the fuzzy ARTMAP architecture.

Some preprocessing of the input patterns takes place before they are presented to the ART_a module. The first preprocessing stage takes as an input an M_a-dimensional input pattern and transforms it into an vector $\boldsymbol{a} = (a_1, \ldots, a_{Ma})$, whose every component lies in the interval [0,1]. The second preprocessing stage accepts the vector $\boldsymbol{a}$ as an input and produces a vector $\mathbf{I}$, such that

$$\boldsymbol{I} = (\boldsymbol{a}, \boldsymbol{a}^c) = (a_1, \ldots, a_{Ma}, a_1^c, \ldots, a_{Ma}^c) \tag{1}$$

where

$$a_i^c = 1 - a_i, \qquad 1 \le i \le M_a. \tag{2}$$

The above transformation is called complement coding and it is performed in ART_a at the preprocessor field F_o^a.

Every node j in F_2^a field is connected via a top-down weight with every node i in the F_1^a field. This weight is denoted w_{ji}^a. The vector whose components are equal to the top-down weights emanating from node j in the F_2^a field is designated by $\mathbf{w}_j^a = \{\mathrm{w}_{j1}^a, \ldots, \mathrm{w}_{j2M_a}^a\}$ and it is called ART_a template. FAMNN operates in two distinct phases: training phase and test phase.

For the **training phase**, given a list of MP training input/label pairs, such as $\{\mathbf{I}^1, \mathrm{O}^1\}$, $\{\mathbf{I}^2, \mathrm{O}^2\}, \ldots, \{\mathbf{I}^{MP}, \mathrm{O}^{MP}\}$, we want to train the FAMNN to map every input pattern of the training list to its corresponding label. In order to achieve this goal, the training set is presented repeatedly to the FAMNN until the desired mapping is established for all pairs.

Consider the rth input/label pair (for example, $\{\mathbf{I}^r, \mathbf{O}^r\}$) from the training list. The bottom-up inputs to all the nodes at the F_2^a field of the ART$_a$ module due to the presentation of the rth input pattern are calculated. These bottom-up inputs to a node j in F_2^a are calculated according to the following:

$$T_j^a(\mathbf{I}^r) = \frac{|\mathbf{I}^r \wedge \mathbf{w}_j^a|}{\beta_a + |\mathbf{w}_j^a|} \tag{3}$$

where β_a is called the ART$_a$ choice parameter, and takes values in the interval (0, ∞). From the set of nodes in F_2^a that satisfy the vigilance criterion, we choose the one that receives the maximum bottom-up input from F_1^a. A node j satisfies the vigilance criterion if

$$\frac{|\mathbf{I}^r \wedge \mathbf{w}_{\mathrm{j}}^a|}{|\mathbf{I}^r|} \geq \rho_a \tag{4}$$

where ρ_a is called the vigilance parameter, and takes values in the interval [0, 1]. Each time that an input pair is presented, it is initialized to a value called the baseline vigilance parameter $\overline{\rho_a}$. If no node satisfies the vigilance criterion, an uncommitted node is selected. A node is called uncommitted if it was not selected before by any input pattern. Assume that node $j_{\max}$ in F_2^a has the maximum bottom-up input from (3) and satisfies the vigilance criterion in (4). If node $j_{\max}$ is an uncommitted node, the mapping of the node $j_{\max}$ is designated to be $\mathbf{O}^{\mathrm{r}}$. Note, that $\mathbf{O}^{\mathrm{r}}$ is the label corresponding to the input pattern $\mathbf{I}^{\mathrm{r}}$ presented in F_1^a. Also the top-down weights are modified according to

$$\mathbf{w}_{j_{\max}}^a = \mathbf{I}^r \wedge \mathbf{w}_{j_{\max}}^a. \tag{5}$$

If $j_{\max}$ is a committed node and due to prior learning node $j_{\max}$ is mapped to a label equal to $\mathbf{O}^r$, then the mapping is correct, and the top-down weights in ART$_a$ are modified according to (5). If $j_{\max}$ is a committed node and due to prior learning node $j_{\max}$ is mapped to a label different than $\mathbf{O}^r$ then the mapping is incorrect. In order to disqualify this node, we increase the vigilance parameter ρ_a in ART$_a$ to a level described by the following:

$$\frac{|\mathbf{I}^r \wedge \mathbf{w}_{j_{\max}}^a|}{|\mathbf{I}^r|} + \varepsilon \tag{6}$$

where ε is taken to be a very small positive constant. Then, we choose another node ($j_{\max}$) that maximizes the bottom-up input of (3) and satisfies the vigilance criterion of (4). This procedure is repeated until a node $j_{\max}$ in F_2^a is found that maximizes the bottom-up of (3), satisfies the vigilance criterion of (4) and is mapped to the correct output. If during a particular presentation of all the input–output pairs (called epoch) no weight changes occur, the learning process is considered complete. Otherwise, the patterns are presented again.

In the previous steps we used the "fuzzy-min" ($\wedge$) operator. The fuzzy min operator of two vectors $\mathbf{w}_1$ and $\mathbf{w}_2$ is a vector whose components are equal to the minimum of the corresponding components of $\mathbf{w}_1$ and $\mathbf{w}_2$. We also used the "size" ($|\cdot|$) operator. The size of a vector $\mathbf{w}$ is defined to be equal to the sum of its components.

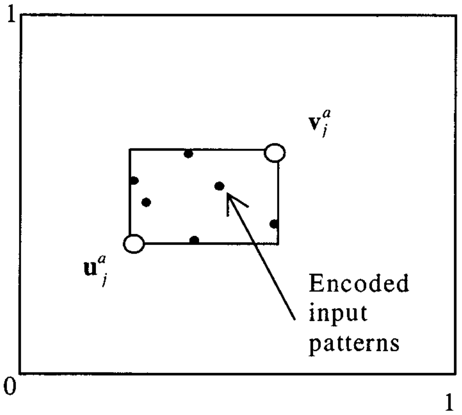


Fig. 2. Representation of the template $\mathbf{w}_j^a = (\mathbf{u}_j^a, (\mathbf{v}_j^a)c)$ in F_2^a field of FAMNN, in terms of the rectangle with endpoints $\mathbf{u}_j^a$ and $\mathbf{v}_j^a$. In the figure, $\mathrm{M}_a = 2$.

For the **test phase** in fuzzy ARTMAP, the values of the top-down weights and the fuzzy ARTMAP parameter values $(\beta_a, \overline{\rho_a})$ are set to the values that they had at the end of the training phase. For each input pattern $\mathbf{I}$ from the test list, we calculate the bottom-up inputs as they were defined in (3) considering all committed nodes and an uncommitted node. From the set of nodes in F_2^a that satisfy the vigilance criterion, the one that receives maximum bottom-up input is chosen and the label of the input pattern is designated as $\hat{\mathbf{O}}$, where $\hat{\mathbf{O}}$ is the label that the node has been mapped to, in the training phase. If an uncommitted node is chosen, the label of the input pattern is designated as "unknown."

The templates $\mathbf{w}_j^a$ that are formed at the F_2^a field of fuzzy ARTMAP during training are compressed representations of the training input patterns. Template $\mathbf{w}_j^a = \{\mathbf{u}_j^a, (\mathbf{v}_j^a)^c\}$ in F_2^a has an interesting geometrical interpretation. It can be represented by a hyper-box in the M$_a$-dimensional space. This hyper-box includes within its boundaries all the training input patterns that were coded by the template. A hyper-box can be defined by its endpoints. These are the points of the boundary of the hyper-box with the smallest and largest coordinates of the hyper-box points, respectively. The first M$_a$ elements of the template $\mathbf{w}_j^a$ define a vector $\mathbf{u}_j^a$ which is the lower left endpoint of the hyper-box and the last M$_a$ elements define the complement $(\mathbf{v}_j^a)^c$ of a vector $\mathbf{v}_j^a$ which is the upper right endpoint of the hyper-box. The representation of a template $\mathbf{w}_j^a$ as a hyper-box is illustrated in Fig. 2.

The maximum allowable size of the hyper-box is controlled by the baseline vigilance parameter value $\overline{\rho_a}$. The larger this vigilance parameter the smaller will be the sizes of the hyper-boxes created in the training phase of FAMNN.

III. The FAMNN Modification

In this section, we present a modification of the FAMNN, named FAMNN-m, that performs well in the presence of noise. Prior to classification, features are extracted from textures. The set of feature vectors extracted is divided into training and test sets. The training phase of FAMNN-m is exactly the same as the training phase of FAMNN. The test phase is a modification of the test phase of FAMNN that exhibits superior generalization performance than the standard FAMNN if the textures are affected by more noise. The training phase is implemented in way so that the modification does not introduce any extra overhead to the test phase.

A. The Noise Model

Consider the mapping between the image $\boldsymbol{Z}$ and the feature vector $\boldsymbol{F}^{Z}$:

$$\boldsymbol{F}^{Z} = \Im\{\boldsymbol{Z}\}. \tag{7}$$

As we mentioned earlier, in many applications it is required that the features capture only the shape characteristics and not the actual amplitude or average value of the signal. Thus, the features are insensitive to linear transformations of the image:

$$\Im\{\boldsymbol{Z}\} = \Im\{a\boldsymbol{Z} + b\} \tag{8}$$

where a and b are constants. According to the previous notation, the feature vector $\boldsymbol{F}^{N}$ extracted from a noise surface $\boldsymbol{N}$ is $\boldsymbol{F}^{N} = \Im\{\boldsymbol{N}\}$. In order to change the mean and variance of a random variable such as $\boldsymbol{N}$, we apply a linear transformation of the form $a\boldsymbol{N} + b$. According to (8), $\boldsymbol{F}^{N}$ is independent of linear transformations of the noise surface, which implies that $\boldsymbol{F}^{N}$ is independent of the mean and standard deviation of the noise $\boldsymbol{N}$. In order to be consistent with the usual terminology of FAMNN, we will refer to complementary encoded feature vectors as *patterns*, and to the complementary encoded feature vector $\boldsymbol{F}^{N}$ of the noise surface *as* the *noise pattern* $\mathbf{I}^{N}$.

We now find an approximate relation between the feature vector extracted from an image $\boldsymbol{Z}$ and the feature vector extracted from the same image corrupted by noise $\boldsymbol{N}$. If $\boldsymbol{Z}_{\mathrm{o}}$ and $\boldsymbol{N}_{\mathrm{o}}$ are normalized images (zero mean and standard deviation 1), the corresponding images are $\boldsymbol{Z} = \sigma_{\mathrm{Z}}\boldsymbol{Z}_{\mathrm{o}} + \mathrm{m}_{\mathrm{Z}}$, $\boldsymbol{N} = \sigma_{\mathrm{N}}\boldsymbol{N}_{\mathrm{o}} + \mathrm{m}_{\mathrm{N}}$ respectively. Then, if the image $\boldsymbol{Z} = \sigma_{\mathrm{Z}}\boldsymbol{Z}_{\mathrm{o}} + \mathrm{m}_{\mathrm{Z}}$ is corrupted by noise $\boldsymbol{N} = \sigma_{\mathrm{N}}\boldsymbol{N}_{\mathrm{o}} + \mathrm{m}_{\mathrm{N}}$ the feature vector extracted will be

$$\begin{aligned} &\Im\left\{[\sigma_{\mathrm{Z}}\boldsymbol{Z}_{\mathrm{o}} + \mathrm{m}_{\mathrm{Z}}] + [\sigma_{\mathrm{N}}\boldsymbol{N}_{\mathrm{o}} + \mathrm{m}_{\mathrm{N}}]\right\} \\ &\quad = \Im\left\{\sigma_{\mathrm{Z}}\boldsymbol{Z}_{\mathrm{o}} + \sigma_{\mathrm{N}}\boldsymbol{N}_{\mathrm{o}}\right\} \\ &\qquad \text{(features independent of mean)} \\ &\quad = \Im\ \left\{[\sigma_{\mathrm{Z}}\boldsymbol{Z}_{\mathrm{o}} + \sigma_{\mathrm{N}}\boldsymbol{N}_{\mathrm{o}}]/(\sigma_{\mathrm{Z}} + \sigma_{\mathrm{N}})\right\} \\ &\qquad \text{(features independent of contrast)} \\ &\quad = \Im\left\{(1-\gamma)\boldsymbol{Z}_{\mathrm{o}} + \gamma\boldsymbol{N}_{\mathrm{o}}\right\} \end{aligned} \tag{9}$$

where $\gamma = \sigma_{\mathrm{N}}/(\sigma_{\mathrm{Z}} + \sigma_{\mathrm{N}})$. The Taylor series expansion around an arbitrary vector $\mathbf{A}$ (linear approximation, assuming that the transformation $\Im\{\cdot\}$ is differentiable at $\mathbf{A}$) is

$$\Im\{\boldsymbol{Z} + \boldsymbol{N}\} \approx \Im\{\mathbf{A}\} + \nabla\Im\{\mathbf{A}\}\left[(1-\gamma)\boldsymbol{Z}_{\mathrm{o}} + \gamma\boldsymbol{N}_{\mathrm{o}} - \mathbf{A}\right]^{\mathrm{T}}. \tag{10}$$

Similarly for $\Im\{\boldsymbol{Z}\}$ and $\Im\{\boldsymbol{N}\}$

$$\begin{aligned} \Im\{\boldsymbol{Z}\} &= \Im\{\sigma_{\mathrm{Z}}\boldsymbol{Z}_{\mathrm{o}} + \mathrm{m}_{\mathrm{Z}}\} \approx \Im\{\mathbf{A}\} + \nabla\Im\{\mathbf{A}\}[\boldsymbol{Z}_{\mathrm{o}} - A]^{\mathrm{T}} \\ \Im\{\boldsymbol{N}\} &= \Im\{\sigma_{\mathrm{N}}\boldsymbol{N}_{\mathrm{o}} + \mathrm{m}_{\mathrm{N}}\} \approx \Im\{\mathbf{A}\} + \nabla\Im\{\mathbf{A}\}[\boldsymbol{N}_{\mathrm{o}} - \mathbf{A}]^{\mathrm{T}}. \end{aligned} \tag{11}$$

Based on (9)–(11)

$$\Im\{\boldsymbol{Z} + \boldsymbol{N}\} \approx (1-\gamma)\Im\{\boldsymbol{Z}\} + \gamma\Im\{\mathrm{N}\} \quad \text{where } \gamma = \sigma_{\mathrm{N}}/(\sigma_{\mathrm{Z}} + \sigma_{\mathrm{N}}). \tag{12}$$

Equivalently, if $\mathbf{I}^{\mathrm{b}}$ is the pattern extracted from image $\boldsymbol{Z}$ and $\mathbf{I}^{a}$ is the pattern extracted from $\boldsymbol{Z} + \boldsymbol{N}$, then

$$\mathbf{I}^{a} \approx (1-\gamma)\mathbf{I}^{\mathrm{b}} + \gamma\mathbf{I}^{\mathrm{N}}. \tag{13}$$

For instance, if the SNR approaches zero, noise becomes dominant and the value of γ is almost equal to one so that $\mathbf{I}^{a} \approx \mathbf{I}^{\mathrm{N}}$. On the other hand, if the signal-to-noise ratio (SNR) approaches infinity, the signal becomes dominant and the value of γ is almost equal to zero so that $\mathbf{I}^{a} \approx \mathbf{I}^{\mathrm{b}}$. For intermediate values of SNR the signal looks like a linear combination of pure signal and noise.

In practice, the parameter γ is not known since information about the noise that has affected the images is not available. As a result, our approach works as follows. We generate noise $\boldsymbol{N}$ of small standard deviation following a known distribution (for example, Gaussian). This noise may or may not be similar to the noise that actually degrades our good quality images. We use the aforementioned generated noise to extract the pattern $\mathbf{I}^{\mathrm{N}}$ from noise only, and the pattern $\mathbf{I}^{a}$ from the noisy version $(\boldsymbol{Z} + \boldsymbol{N})$ of the good quality image $(\boldsymbol{Z})$. Also, the pattern $\mathbf{I}^{b}$ is extracted from the good quality image $\boldsymbol{Z}$. Finally, the patterns $\mathbf{I}^{a}$, $\mathbf{I}^{b}$, $\mathbf{I}^{\mathrm{N}}$ are used to estimate the value of γ according to (13). The estimated value of γ can now be used in the test phase of FAMNN-m with data that are either excited by the original patterns $\mathbf{I}^{a}$ or by patterns extracted from noisier versions of $\boldsymbol{Z}$, for example, $\boldsymbol{Z} + \boldsymbol{N}'$. Note that $\boldsymbol{N}'$ can be different than $\boldsymbol{N}$. The entire approach is pictorially illustrated in Fig. 6. The previously mentioned approach resembles the well-known method of adding noise to the training data to improve the generalization performance of an NN. In this case too, addition of a small noise to the training data does not imply knowledge of the actual noise affecting the data.

Since a pattern consists of many features, γ is estimated separately for each feature i of the feature vector. Since patterns are extracted from many images, the value of γ for feature i is estimated as $\gamma_i = [1$—the slope] of the line that best fits the points $\{(\mathrm{I}_i^{\mathrm{b}} - \mathbf{I}_i^{\mathrm{N}}), (\mathbf{I}_i^{a} - \mathbf{I}_i^{\mathrm{N}})\}$ and γ is the average of all these γ_is. Note that (13) can be equivalently written as $(\mathbf{I}^{a} - \mathbf{I}^{\mathrm{N}}) = (1-\gamma)(\mathbf{I}^{\mathrm{b}} - \mathbf{I}^{\mathrm{N}})$.

Two feature sets are used in this work, namely energy-based and fractal-based features. A detailed description of these feature sets can be found in [25]. We must note that the value of γ_i depends on the feature type used and the standard deviation of noise. Two examples of estimation of the γ parameter are shown in Fig. 3(a) and (b), respectively, for one FD feature and one NE feature. In Fig. 3(a) and (b) the line $(\mathrm{I}_{\mathrm{i}}^{a} - \mathrm{I}_{\mathrm{i}}^{\mathrm{N}}) = (1-\gamma_{\mathrm{i}})(\mathrm{I}_{\mathrm{i}}^{\mathrm{b}} - \mathrm{I}_{\mathrm{i}}^{\mathrm{N}})$ is fitted, and [1—the slope] of the line gives an estimation of γ_{i}. Our experimental results demonstrate that the FAMNN-m exhibits a better classification accuracy than FAMNN even for textures corrupted with different noise distributions than the one assumed to estimate γ.

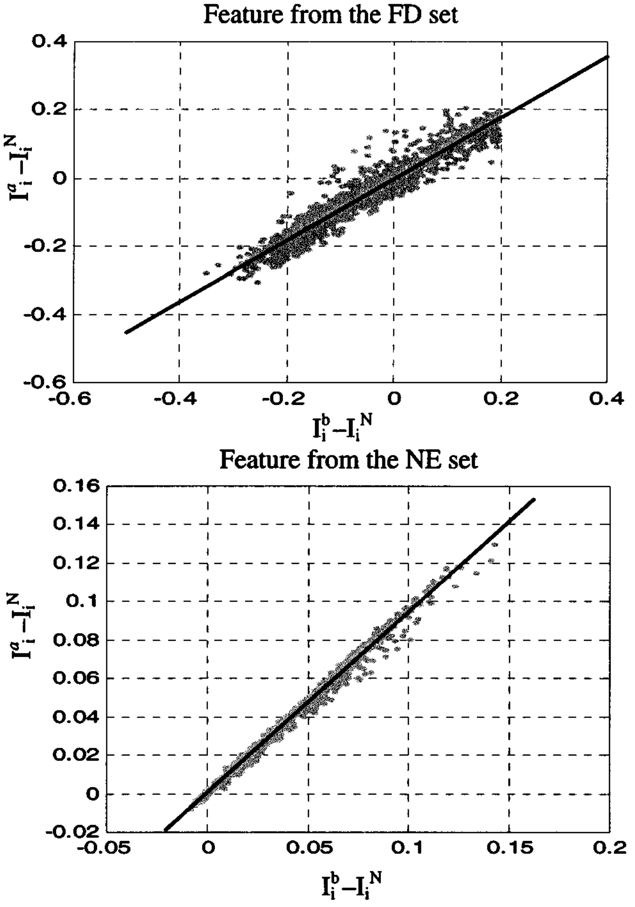


Fig. 3. Linear fit of $I_i^a - I_i^N$ on $I_i^b - I_i^N$ used for estimation of the γ parameter. I_i^b and I_i^a are the ith elements of the patterns (feature vectors) before and after noise is added on the textures. (a) A FD feature is used. (b) A NE feature is used.

B. Training Phase and Test Phase

The training of the FAMNN-m is exactly the same as the training phase of the standard FAMNN. As a reminder, the bottom-up input of node j in F_2^a for a pattern $\mathbf{I}^r$, denoted as $T_j^a(\mathbf{I}^r)$, can be expressed as a ratio of two numbers $\mathrm{N_j}$ and $\mathrm{D_j}$. In particular

$$\mathrm{D_j} = \beta_\alpha + |\mathbf{w}_\mathrm{j}^a| \tag{14}$$

$$\mathrm{N_j} = |\mathbf{I}^r \wedge \mathbf{w}_\mathrm{j}^a|. \tag{15}$$

We notice that after training is over, $\mathrm{D_j}$ remains unchanged. Even if the training is on-line, this quantity remains unchanged when the test phase takes place. The quantities $\mathrm{D_j}$ for every node j, are stored in memory along with the templates $\mathbf{w}_\mathrm{j}$, so that they are not recalculated in the test phase.

If the size of the patterns after complementary encoding is $2\mathrm{M}_a$, then, we are dealing with an M_a-dimensional hyperspace. Each node in the F_2^a layer can then be represented as an M_a-dimensional hyperbox. The training patterns that are associated with this node are included inside the boundaries of this hyperbox. When a pattern is tested, a hyperbox competes with other hyper-boxes to associate this pattern with the class that it represents. Basically, only the hyperboxes that are close to the pattern have some chance to be associated with it. Generally, each hyperbox is dominant in the region that it occupies and in some region around it. The latter, depends on the size of the hyperbox, on the parameter β_a as it can be seen from (3), and on its proximity to other hyper-boxes (for more details see Georgiopoulos *et al.,* [22]). According to (13), patterns that are extracted from a noisy texture will move, at least in most of the cases, toward the direction of the pattern that is extracted from a pure noise surface. Therefore, if in the test phase of the FAMNN we modify the region of influence of each hyperbox so that it gives more emphasis to coordinates that are further away from the noise pattern than the ones that are closer, we have a better chance of correctly classifying patterns extracted from noisy signals. Furthermore, the modification of the test phase of the FAMNN needs to be such that it allows the correct classification of textures used for training as well. We have implemented a variation of the test phase of FAMNN (called, FAMNN-m) to achieve the previously mentioned objectives. In the sequel, we present a simplified analysis that gives some intuition of how we came up with this modification and what is its effect on the classification performance.

First, we define the distance between a template $\mathbf{w}$ and a pattern $\mathbf{I}$ as

$$\mathrm{dist}(\mathbf{w}, \mathbf{I}) = |\mathbf{w}| - |\mathbf{w} \wedge \mathbf{I}|. \tag{16}$$

In the special case where the template $\mathbf{w}$ is a pattern $\mathbf{I}'$, the distance between $\mathbf{I}$ and $\mathbf{I}'$ as it is defined in (16) is equal to

$$\mathrm{dist}(\mathbf{I}', \mathbf{I}) = ||\mathbf{I} - \mathbf{I}'||_1 = \sum_{\mathrm{i}=1}^{\mathrm{M}_a} |\mathbf{I}_\mathrm{i} - \mathbf{I}'_\mathrm{i}|. \tag{17}$$

Fig. 4 presents some examples of the distance as it is defined by (16) and (17).

For a pattern $\mathbf{I}$ at the boundary between the regions of dominance of two nodes with templates $\mathbf{w}_1$ and $\mathbf{w}_2$, the corresponding bottom-up inputs $\mathrm{T}_1(\mathbf{I})$ and $\mathrm{T}_2(\mathbf{I})$ are equal

$$\begin{aligned}\mathrm{T}_1(\mathbf{I}) = \mathrm{T}_2(\mathbf{I}) &\Rightarrow \frac{|\mathbf{w}_1 \wedge \mathbf{I}|}{|\mathbf{w}_1| + \beta_a} = \frac{|\mathbf{w}_2 \wedge \mathbf{I}|}{|\mathbf{w}_2| + \beta_a} \\ &\Rightarrow \frac{|\mathbf{w}_1| - \mathrm{dist}(\mathbf{w}_1, \mathbf{I})}{|\mathbf{w}_1| + \beta_a} = \frac{|\mathbf{w}_2| - \mathrm{dist}(\mathbf{w}_2, \mathbf{I})}{|\mathbf{w}_2| + \beta_a}.\end{aligned} \tag{18}$$

In the case where the textures are affected by more noise, the patterns extracted from the textures, move toward the noise pattern $\mathbf{I}^\mathrm{N}$. Assume that the hyper-box defined by $\mathbf{w}_2$ is closer to the noise pattern than the one defined by $\mathbf{w}_1$, that is, $\mathrm{dist}(\mathbf{w}_1, \mathbf{I}^\mathrm{N}) > \mathrm{dist}(\mathbf{w}_2, \mathbf{I}^\mathrm{N})$. Then, node 2 benefits from the movement of patterns due to noise over node 1 because it tends to "capture" patterns that belong to node 1 and move toward $\mathbf{I}^\mathrm{N}$. In our approach we modify the boundary between the regions of dominance according to our previous discussion, by dividing the bottom-up input of a node with a term that depends on the corresponding template $\mathbf{w}$. We denote this term as $\mathbf{N}(\mathbf{w})$. Assume that $\mathbf{I}'$ is a pattern on the new boundary that corresponds to the pattern $\mathbf{I}$ on the old boundary. We want $\mathbf{I}'$ to be further away from node 1 and closer to node 2, so that the boundary between the regions of dominance moves toward node 2. The boundary movement due to this modification helps

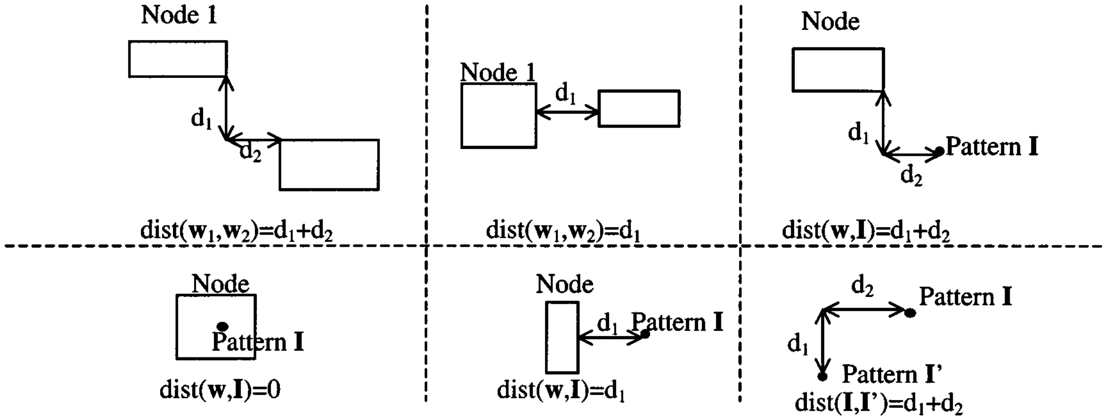


Fig. 4. Some examples of the distance as it is defined in (16) and (17).

node 1 to "recapture" the misclassified patterns. Then, for the pattern $\mathbf{I}'$ on the new boundaries

$$\begin{aligned}\frac{\mathrm{T}_1(\mathbf{I}')}{\mathrm{N}(\mathbf{w}_1)} = \frac{\mathrm{T}_2(\mathbf{I}')}{\mathrm{N}(\mathbf{w}_2)} &\Rightarrow \frac{|\mathbf{w}_1 \wedge \mathbf{I}'|}{\mathrm{N}(\mathbf{w}_1)\,(|\mathbf{w}_1| + \beta_a)}\\ &= \frac{|\mathbf{w}_2 \wedge \mathbf{I}'|}{\mathrm{N}(\mathbf{w}_2)\,(|\mathbf{w}_2| + \beta_a)} \Rightarrow \frac{|\mathbf{w}_1| - \mathrm{dist}(\mathbf{w}_1,\, \mathbf{I}')}{\mathrm{N}(\mathbf{w}_1)\,(|\mathbf{w}_1| + \beta_a)}\\ &= \frac{|\mathbf{w}_2| - \mathrm{dist}(\mathbf{w}_2,\, \mathbf{I}')}{\mathrm{N}(\mathbf{w}_2)\,(|\mathbf{w}_2| + \beta_a)}. \qquad (19)\end{aligned}$$

We consider only the region between the closest boundaries of the two hyperboxes, to illustrate the effect of this modification. If a pattern $\mathbf{I}$ that resides in this region moves to pattern $\mathbf{I}'$ that resides also in the same region then $\mathbf{I}'$ is further away from node 1 than $\mathbf{I}$ by a distance equal to dist($\mathbf{I}$, $\mathbf{I}'$) and closer to node 2 by the same distance. Equation (19) becomes

$$\begin{aligned}&\frac{|\mathbf{w}_1| - \mathrm{dist}(\mathbf{w}_1,\, \mathbf{I}) - \mathrm{dist}(\mathbf{I},\, \mathbf{I}')}{\mathrm{N}(\mathbf{w}_1)\,(|\mathbf{w}_1| + \beta_a)}\\ &\quad = \frac{|\mathbf{w}_2| - \mathrm{dist}(\mathbf{w}_2,\, \mathbf{I}) + \mathrm{dist}(\mathbf{I},\, \mathbf{I}')}{\mathrm{N}(\mathbf{w}_2)\,(|\mathbf{w}_2| + \beta_a)}. \qquad (20)\end{aligned}$$

Solving for dist($\mathbf{I}$, $\mathbf{I}'$), we find that it is equal to

$$\begin{aligned}&\mathrm{dist}(\mathbf{I},\, \mathbf{I}')\\ &\quad = \frac{[\mathrm{N}(\mathbf{w}_2) - \mathrm{N}(\mathbf{w}_1)]\,[|\mathbf{w}_1| + \beta_a]\,[|\mathbf{w}_2| + \beta_a]}{\mathrm{N}(\mathbf{w}_1)\,(|\mathbf{w}_1| + \beta_a) + \mathrm{N}(\mathbf{w}_2)\,(|\mathbf{w}_2| + \beta_a)}\,\mathrm{T}_1(\mathbf{I}). \qquad (21)\end{aligned}$$

We want to select the term N($\mathbf{w}$) in a way that satisfies the following property:

- dist($\mathbf{I}$, $\mathbf{I}'$) should be proportional to dist($\mathbf{w}_1$, $\mathbf{I}^{\mathrm{N}}$)− dist($\mathbf{w}_2$, $\mathbf{I}^{\mathrm{N}}$).

Motivation: In the case where the textures are affected by more noise, the patterns extracted from the textures, move toward the noise pattern $\mathbf{I}^{\mathrm{N}}$. If $\mathrm{dist}(\mathbf{w}_1, \mathbf{I}^{\mathrm{N}}) > \mathrm{dist}(\mathbf{w}_2, \mathbf{I}^{\mathrm{N}})$ then we want to move the boundary between the regions of dominance toward node 2 which is closer to the noise pattern $\mathbf{I}^{\mathrm{N}}$, in order to increase the probability of recapturing the patterns that moved toward $\mathbf{I}^{\mathrm{N}}$ and belonged to node 1. It is reasonable then to move the boundary toward node 2 by an amount which is proportional to the difference in distance of the two nodes $\mathbf{w}_1$ $\mathbf{w}_2$ from $\mathbf{I}^{\mathrm{N}}$. For instance, if the distance of two nodes from $\mathbf{I}^{\mathrm{N}}$ is the same: $\mathrm{dist}(\mathbf{w}_1, \mathbf{I}^{\mathrm{N}}) = \mathrm{dist}(\mathbf{w}_2, \mathbf{I}^{\mathrm{N}})$ we do not want to favor one node versus the other.

We select $\mathrm{N}(\mathbf{w}) = \mathrm{M}_a - \gamma\ \mathrm{dist}(\mathbf{w},\, \mathbf{I}^{\mathrm{N}})$, where γ is a small constant associated with the amount of noise that has affected the textures [see (13)]. Next, we show that this choice for N($\mathbf{w}$) satisfies the aforementioned desired property:

Let us first illustrate the effect of the terms N($\mathbf{w}_1$) and N($\mathbf{w}_2$). For example, if $\gamma = 0.1$, $\mathrm{dist}(\mathbf{w}_1, \mathbf{I}^{\mathrm{N}}) = 0.5\ \mathrm{M}_a$, and dist($\mathbf{w}_2$, $\mathbf{I}^{\mathrm{N}}$) $= 0.4\ \mathrm{M}_a$. Then, $\mathrm{N}(\mathbf{w}_1) = 0.95\ \mathrm{M}_a$ and $\mathrm{N}(\mathbf{w}_2) = 0.96\ \mathrm{M}_a$ and N($\mathbf{w}_2$)− $\mathrm{N}(\mathbf{w}_1) = 0.01\ \mathrm{M}_a$. On the other hand, if $\mathrm{dist}(\mathbf{w}_2, \mathbf{I}^{\mathrm{N}}) = 0.3\ \mathrm{M}_a$ then $\mathrm{N}(\mathbf{w}_2) = 0.97\ \mathrm{M}_a$ and N($\mathbf{w}_2$)− $\mathrm{N}(\mathbf{w}_1) = 0.02\,\mathrm{M}_a$. Consequently, what significantly affects the dist($\mathbf{I}$, $\mathbf{I}'$) is not the actual values of N($\mathbf{w}_1$) and N($\mathbf{w}_2$) which are very similar and close to M_a, but the difference of the two. We can thus rewrite (25) as follows:

$$\begin{aligned}&\mathrm{dist}(\mathbf{I},\, \mathbf{I}')\\ &\quad \approx \frac{[\mathrm{N}(\mathbf{w}_2) - \mathrm{N}(\mathbf{w}_1)]\,[(|\mathbf{w}_1| + \beta_a)\,(|\mathbf{w}_2| + \beta_a)]}{\mathrm{M}_a\,[|\mathbf{w}_1| + |\mathbf{w}_2| + 2\beta_a)]}\,\mathrm{T}_1(\mathbf{I}).\\ &\qquad\qquad (22)\end{aligned}$$

By selecting $\mathrm{N}(\mathbf{w}) = \mathrm{M}_a - \gamma\ \mathrm{dist}(\mathbf{w},\, \mathbf{I}^{\mathrm{N}})$, we have [N($\mathbf{w}_2$)− N($\mathbf{w}_1$)] $=$ γ [dist($\mathbf{w}_1$, $\mathbf{I}^{\mathrm{N}}$)− dist($\mathbf{w}_2$, $\mathbf{I}^{\mathrm{N}}$)]. Therefore, dist($\mathbf{I}$, $\mathbf{I}'$) is proportional to [dist($\mathbf{w}_1$, $\mathbf{I}^{\mathrm{N}}$)− dist($\mathbf{w}_2$, $\mathbf{I}^{\mathrm{N}}$)] and the desired property is satisfied. The other factor in (22) being equal to

$$G = \frac{[(|\mathbf{w}_1| + \beta_a)\,(|\mathbf{w}_2| + \beta_a)]}{\mathrm{M}_a\,[|\mathbf{w}_1| + |\mathbf{w}_2| + 2\beta_a)]}$$

$$\mathrm{T}_1(\mathbf{I}) = \frac{1}{\mathrm{M}_a \left\{ \dfrac{1}{|\mathbf{w}_1| + \beta_a} + \dfrac{1}{|\mathbf{w}_2| + \beta_a} \right\}}\,\mathrm{T}_1(\mathbf{I})$$

does not depend on the factors N($\mathbf{w}_1$) and N($\mathbf{w}_2$). Term G is not a direct result of our modification. It is a result of the characteristics of the bottom-up input as it is defined in the FAMNN architecture.

It was mentioned earlier that the bottom-up input is not only a measure of proximity, but it also depends on the template sizes and on the β_a parameter. Term G reflects the effect of these bottom-up input characteristics to dist($\mathbf{I}$, $\mathbf{I}'$). Some intuition about the effect of this term is shown in Fig. 5. In Fig. 5(a) and

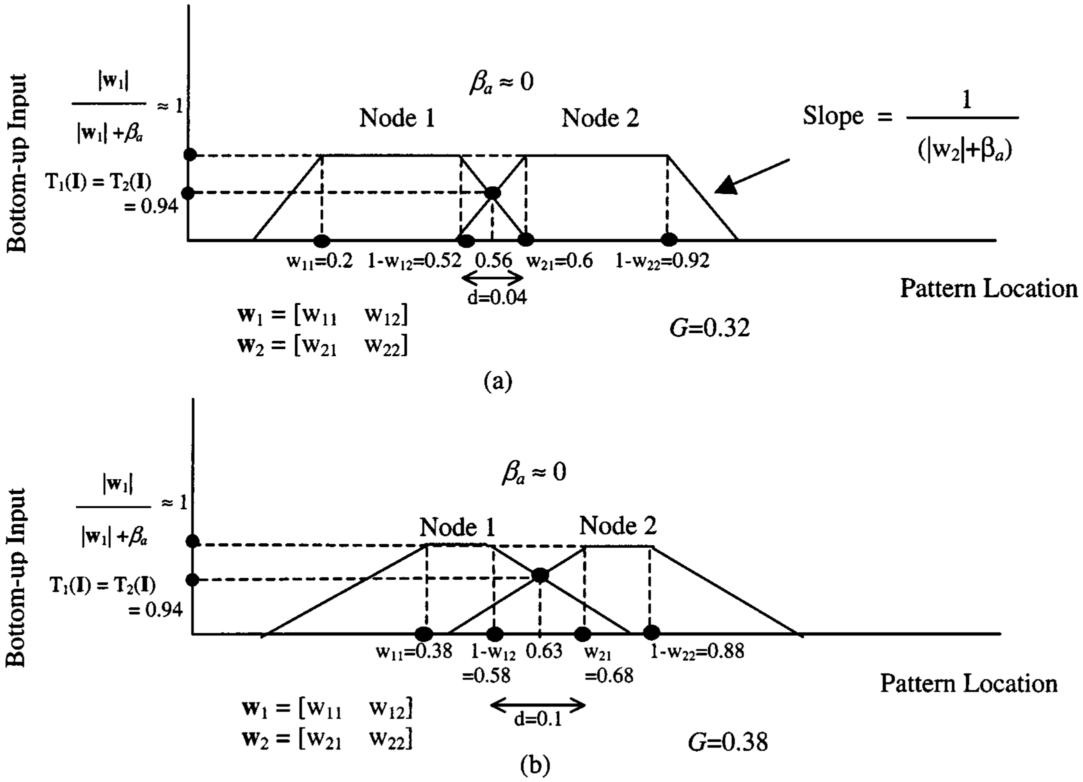


Fig. 5. Bottom-up inputs as a function of the input pattern and boundaries between regions of dominance for (a) Large $1/(\mathrm{w}_1 + \beta_a)$, $1/(\mathrm{w}_2 + \beta_a)$, (b) small $1/(\mathrm{w}_1 + \beta_a)$, $1/(\mathrm{w}_2 + \beta_a)$.

(b) we depict the bottom-up inputs of two neighboring 1-D templates with respect to the input pattern **I**. We have assumed that the templates in Fig. 5(a) are larger than the ones in Fig. 5(b). In both cases, the input pattern **I** for which the two bottom-up inputs $\mathrm{T}_1(\mathbf{I})$ and $\mathrm{T}_2(\mathbf{I})$ are equal, defines the boundary between the regions of dominance of the two nodes. The bottom-up input is constant inside the region defined by a template as it is shown in Fig. 5. The bottom-up input of a node decreases linearly with the distance of the pattern from it. The slope of the linear decrease is, for node 1 for example, equal to $1/(|\mathrm{w}_1| + \beta_a)$. Assume that the value of the bottom-up input on the boundary between nodes is the same for both examples in Fig. 5(a) and (b). Then, the larger the slopes $1/(|\mathrm{w}_1| + \beta_a)$ and $1/(|\mathrm{w}_2| + \beta_a)$ the closer the boxes defined by the templates are. The term G is inversely proportional to the sum of the two slopes, which is desired; the further away the nodes are, the larger the dist(**I**, **I**′) is. For instance, for the nodes in Fig. 5(a) which are close $G = 0.32$, and for the nodes in Fig. 5(b) which are further away $G = 0.38$.

According to our previous discussion, we define the bottom-up input for node j of the FAMNN-m in the test phase to be equal to

$$Tm_j^a(\mathbf{I}') = \frac{1}{\mathrm{M}_a - \gamma \mathrm{dist}(\mathbf{w}_\mathrm{j}^a, \mathbf{I}^N)} \frac{|\mathbf{I}' \wedge \mathbf{w}_\mathrm{j}^a|}{\beta_a + |\mathbf{w}_\mathrm{j}^a|} \tag{23}$$

where $\mathbf{I}^r$ is the rth input pattern from the list of test patterns. The value of γ is computed according to the method described in Section III-A. The parameter γ and the noise pattern $\mathbf{I}^\mathrm{N}$ are computed in the training phase (see Fig. 6 for a pictorial illustration) and used in the test phase. Also, the denominator of the bottom-up input $Tm_\mathrm{j}^a(\mathbf{I}^r)$ of the FAMNN-m is $\mathrm{D_j} = (\mathrm{M}_a - \gamma$ dist$(\mathbf{w}_\mathrm{j}^a, \mathbf{I}^\mathrm{N}))(\beta_\alpha + |\mathbf{w}_\mathrm{j}^a|)$ which is stored in the training phase and it is not recalculated in the test phase.

A two-dimensional example that gives some insight about the effect of the modification is shown in Fig. 7. The input patterns in this example belong to two classes denoted as class 1 and class 2. They are represented by "$+$" and "$\times$," respectively. These patterns are NE-based feature vectors that are actually extracted from two aerial textures, respectively. Half of the patterns were extracted from best quality textures available, and half from the two textures when noise is present. A part of the patterns that were extracted from the best quality textures available is used for training [Fig. 7(a)]. The rest of the patterns that were extracted from the best quality textures available [Fig. 7(c)] and all the patterns that were extracted from the noisy textures [Fig. 7(b) and (d)] are used for testing. The "light gray" area represents the region of dominance of the hyperboxes that belong to class 2, and the "white" area the region of dominance of the hyper-boxes that belong to class 1. The "dark gray" area belongs to class 2 for the standard FAMNN and to class 1 for the FAMNN-m.

We can see that the boundary between the regions of dominance of the two classes has been shifted toward class 2, which is the one closer to the noise pattern. In Fig. 7(a), we depict the patterns that are used for training and the boxes that have encoded these patterns. Both FAMNN and FAMNN-m classify correctly all patterns (the $\times$s are in the "light gray" area and the $+$s in the "white" area). In Fig. 7(b), we depict the patterns that are extracted from the textures that were used for training, but

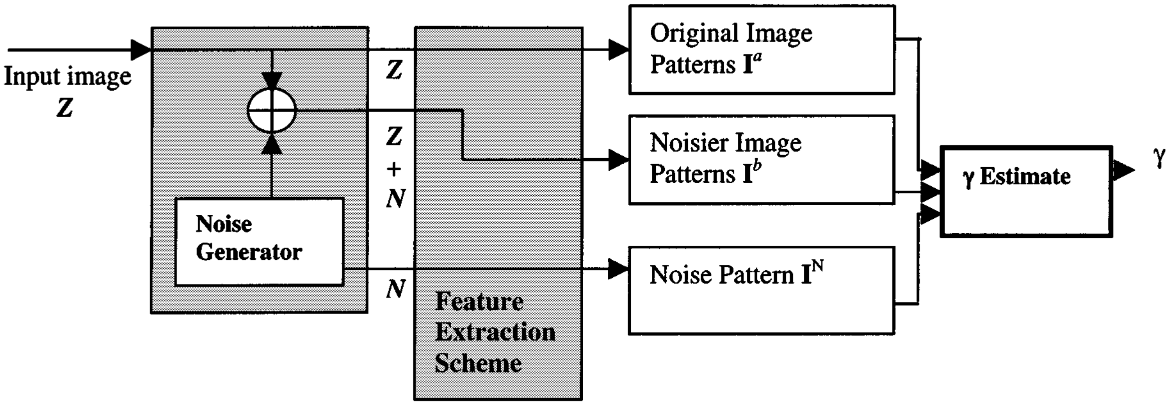

Input image
Z
Z
Z
+
N
N
Noise Generator
Feature Extraction Scheme
Original Image Patterns $\mathbf{I}^a$
Noisier Image Patterns $\mathbf{I}^b$
Noise Pattern $\mathbf{I}^N$
γ Estimate
γ

Fig. 6. Block diagram depicting calculation of the γ parameter.

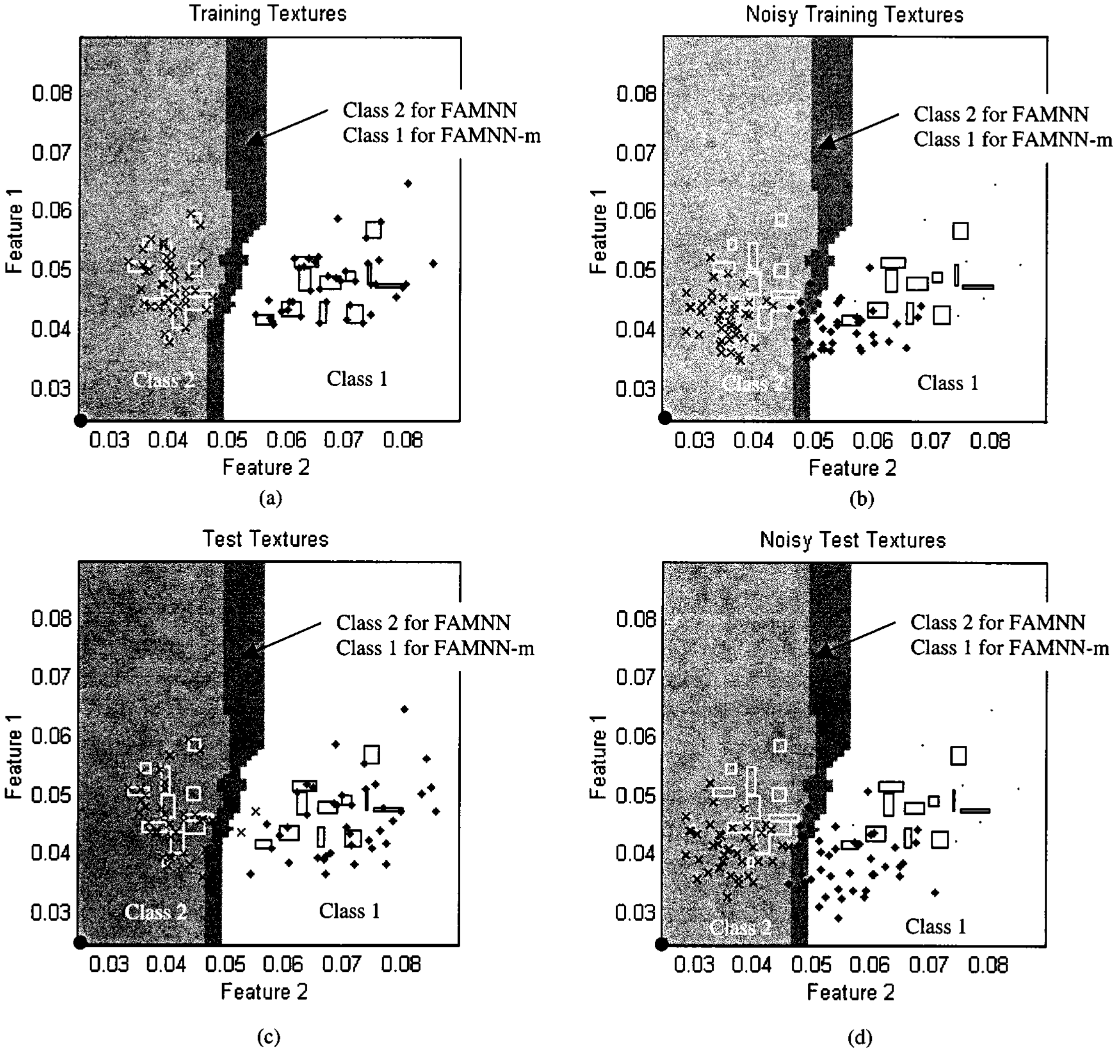


Fig. 7. A comparison of FAMNN and FAMNN-m in two-dimensions. The “light gray” area and the “white” area represent the regions of dominance of class 2 and class 1, respectively. The “dark gray” area is included in the region of dominance of class 2, if FAMNN is used, and in the region of dominance of class 1, if FAMNN-m is used. (a) Patterns extracted from the best quality textures that were used for training. (b) Patterns extracted from the textures that were used for training when noise is added. (c) Patterns extracted from best quality test textures. (d) Patterns extracted from the test textures when noise is added.

now the textures are affected by additive noise. We notice a displacement of the patterns with respect to the corresponding ones in Fig. 7(a). The patterns have moved toward the noise vector. As a result class 2, which is closer to the noise vector, tends to “capture” the patterns that belong to class 1. For Fig. 7(b) we see that ten patterns are misclassified for FAMNN, but only one for FAMNN-m. Similarly, Fig. 7(c) depicts the regions of dominance for FAMNN and FAMNN-m and the test patterns that have been extracted from the aerial textures. In this case, FAMNN misclassifies two patterns and FAMNN-m four patterns. On the other hand, in Fig. 7(d) we see that the movement of patterns toward noise leads FAMNN to misclassify seven patterns, but FAMNN-m misclassifies only one. For this example the value of γ was selected equal to 0.45. We must note, that the noise distribution that affects the textures can have an arbitrary mean. On the other hand, the displacement of patterns in the feature space is the result of the noise addition to the signal.

From this example we notice the effect of the modification. For textures that have not been affected by extra noise FAMNN may have slightly better classification performance than FAMNN-m. For the textures though, that are affected by extra noise FAMNN-m gives significantly better classification results.

IV. Experimental Results

We performed a large number of experiments in order to illustrate the effectiveness of the proposed FAMNN-m for classification of signals affected by additive noise. We considered the application of texture classification. We compared FAMNN-m with 1) the original FAMNN and 2) with the original FAMNN when it is trained with both noisy and best quality data available. Although the 2) results may be a fairer comparison, the 1) results are significant because they convey a more complete picture of the FAMNN-m advantages over FAMNN.

We used two different texture sets. The first texture set consisted of 20 textures obtained from the Brodatz album [23]. The second texture set consisted of 20 textures obtained from aerial images [24]. The training and testing on these two texture sets were performed separately. We extracted a large number of patterns from both texture sets, as we will describe later. We considered these two texture sets because we wanted to examine the classification performance of FAMNN-m on a set, such as Brodatz textures, obtained in ideal environmental conditions, and a set, such as aerial textures, that represent a more realistic situation.

Two feature sets with different characteristics were extracted from each texture set. The first one consisted of 12 NE-based features [25], and the second one consisted of six FD-based features [25]. The NE-based feature set is more robust to noise than the FD-based feature set, but the classification results that were obtained with the FD-based feature set are better when extra noise is not present. The training and testing were performed separately for each feature set.

Let us consider the NE-based feature set. For Brodatz textures, a total of 2560 feature vectors (or equivalently patterns) that were extracted from nonoverlapping windows of size 16 × 16 were used for training. A total of 1280 feature vectors were used for testing in the case where extra noise is not present. For aerial textures, a total of 1280 feature vectors that were extracted from nonoverlapping windows of size 16 × 16 were used for training. A total of 640 feature vectors were used for testing in the case where no extra noise is added. Similarly, for the FD-based feature set, 2560 training feature vectors and 1280 test feature vectors when no extra noise is added were extracted from Brodatz textures. For aerial textures, 1280 training feature vectors and 640 test feature vectors when no extra noise is added were extracted. The size of the windows was selected to be relatively small (16 × 16) so that segmentation of more that one texture in the same image is possible if desired. A similar window size of 17 × 17 was selected in [15].

We considered three different types of additive noise: Gaussian noise, uniform noise and exponential noise. For each type of noise we considered different values of standard deviation. Approximately 82 000 NE-based and 82 000 FD-based feature vectors were extracted from the 20 Brodatz textures for *each* type of noise and for *each* value of standard deviation. Approximately 20 000 NE-based and 20 000 FD-based feature vectors were extracted from the 20 aerial textures for *each* type of noise and for *each* value of standard deviation.

The scope of this paper is to illustrate the superior performance of FAMNN-m over FAMNN in the presence of noise independently of the size of the network. For this reason, we compared the FAMNN and the FAMNN-m for different network sizes. In order to achieve different network sizes, we used different values of baseline vigilance $\overline{\rho_a}$. The value of β_a was selected to be equal to one for all FAMNNs and FAMNN-ms.

All combinations of the parameters described above were considered. More specifically, the classification performance of FAMNN and FAMNN-m was examined on the two texture sets, for the two feature sets, for three different types of noise, for different values of the standard deviation of noise, and for different sizes of the networks. The classification results are shown in Figs. 8–11 and Tables I and II.

The distance between the noise pattern extracted from a pure noise texture and the patterns extracted from other textures is relatively large, independently of the type of noise. Consequently, the noise pattern could have been extracted from any type of white noise. For our experiments the noise pattern $\mathbf{I}^{\mathrm{N}}$ was extracted from a pure Gaussian noise texture for both FD-based and NE-based feature sets. The noise pattern was estimated as it was described in Section III-A. More specifically, feature vectors were extracted from 16 × 16 samples of the noise texture. The noise feature vector $\boldsymbol{F}^{\mathrm{N}}$ was estimated as the average of all these feature vectors (each element of the noise feature vector was estimated as the average of the corresponding elements of all feature vectors extracted from the noise texture). The noise pattern $\mathbf{I}^{\mathrm{N}}$ is the complementary encoded noise feature vector $\boldsymbol{F}^{\mathrm{N}}$.

The value of γ was estimated following the approach that was described in Section III-C. The value of γ was estimated for standard deviation of noise equal to 14.2, but it was kept constant in the test phase, since it is assumed that information about the noise that has corrupted the textures is not available. Nevertheless, the classification performance of FAMNN-m over FAMNN is improved independently of the standard deviation or distribution of noise that has affected the textures. The value of γ was found to be approximately equal to 0.1 for the NE-feature set, and approximately equal to 0.2 for the FD-feature set.

In summary, all experiments show that the PCC is always larger for FAMNN-m when the standard deviation of noise is larger than 7.2, independently of the type of the noise, the texture set and the feature set. The difference is larger for larger values of the standard deviation. When no extra noise is added

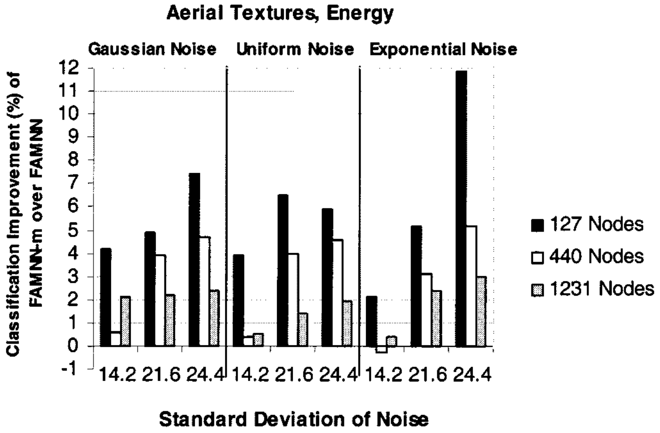


Fig. 8. Comparison of FAMNN and FAMNN-m in terms of percentage of correct classification on the aerial textures, for different number of nodes where the NE-based feature set is used and when (a) Gaussian noise is added, (b) uniform noise is added, and (c) exponential noise is added.

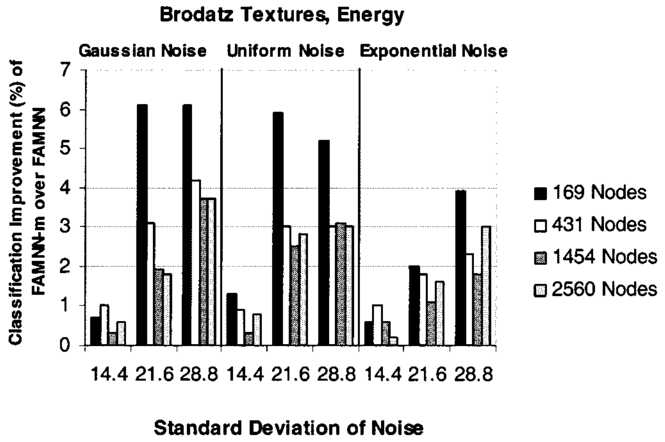


Fig. 9. Comparison of FAMNN and FAMNN-m in terms of percentage of correct classification on the Brodatz textures, for different number of nodes where the NE-based feature set is used and when (a) Gaussian noise is added, (b) uniform noise is added, and (c) exponential noise is added.

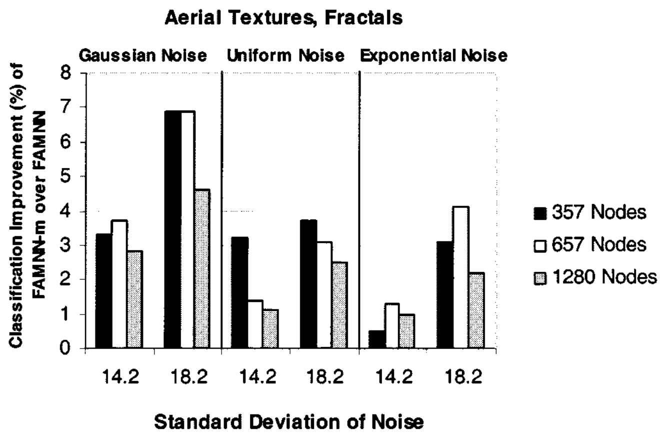


Fig. 10. Comparison of FAMNN and FAMNN-m in terms of percentage of correct classification on the aerial textures, for different number of nodes where the FD-based feature set is used and when (a) Gaussian noise is added, (b) uniform noise is added, and (c) exponential noise is added.

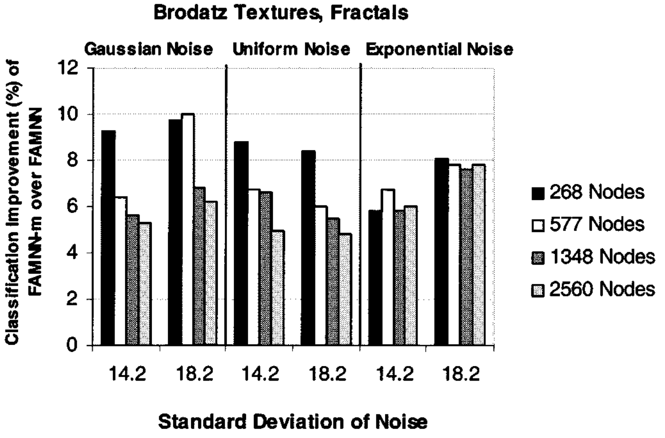


Fig. 11. Comparison of FAMNN and FAMNN-m in terms of percentage of correct classification on the Brodatz textures, for different number of nodes where the FD-based feature set is used and when (a) Gaussian noise is added; (b) uniform noise is added; (c) exponential noise is added.

and when the standard deviation of noise is equal to 7.2, the PCC is similar for the FAMNN and the FAMNN-m. The PCC is slightly better for FAMNN when extra noise is not present, which is the tradeoff for the significantly improved performance of FAMNN-m over FAMNN when noise is present. More specifically, when extra noise is not present, the aerial textures average PCC of the FAMNN-m networks is 84.5% and 88.5% for NE and FD, respectively, and the average Brodatz textures PCC is 89.4% and 94% for NE and FD, respectively. The average PCC of the FAMNN networks is 85.7% and 87.8% for NE and FD, respectively, and for aerial textures, and 89.9% and 94.7% for NE and FD, respectively, and the Brodatz textures. When the standard deviation of additive noise is 7.2, the average PCC of the FAMNN-m networks and for all three types of noise is 84.9% and 85.8% for NE and FD, respectively, and for aerial textures, and 89.6% and 91.7% for NE and FD, respectively, and the Brodatz textures. The average PCC of the FAMNN networks is 85.4% and 84.5% for NE and FD, respectively, and for aerial textures, and 89.8% and 91.2% for NE and FD, respectively, and the Brodatz textures.

The results for higher values of the standard deviation of noise are shown in Tables I and II for the FD-based feature set and the NE-based feature, respectively. We notice that when noise is present and for a specified number of nodes the PCC of FAMNN-m is *always* larger than the PCC of FAMNN for the same number of nodes. The difference in PCC varies between 1% and 12.1%. For instance, the improvement in PCC of FAMNN-m over FAMNN is 11.9% for 127 nodes and when exponential noise of standard deviation 24.4 has affected the aerial textures and when the NE-vector is used. Also, the improvement in PCC of FAMNN-m over FAMNN is 9.7% for 268 nodes and when Gaussian noise of standard deviation 18.2 has affected the Brodatz textures and the FD-vector is used.

Fig. 8 presents the classification improvement of FAMNN-m over FAMNN when the NE-based feature set is used, when the aerial texture set is corrupted by Gaussian, uniform, and exponential noise. The values for standard deviation of noise were equal to 14.2, 21.6, and 24.4 and they correspond to SNRs of 17 dB, 9 dB, and 8 dB, respectively. Three sizes of FAMNN and FAMNN-m were examined: networks with 127 nodes, networks with 440 nodes, and networks with 1231 nodes. Fig. 9 illustrates the classification improvement of FAMNN-m over

TABLE I

PERCENTAGE OF CORRECT CLASSIFICATION OF THE FD-BASED FEATURE VECTORS: (a) EXTRACTED FROM THE 20 AERIAL TEXTURES AND (b) EXTRACTED FROM THE 20 BRODATZ TEXTURES

FD –Vector		***Aerial Textures***					
Type of Noise	**:**	**Gaussian**		**Uniform**		**Exponential**	
Standard Deviation	**:**	14.2	18.2	14.2	18.2	14.2	18.2
FAMNN 357 Nodes		69.3	55.4	66.2	54.4	73.3	65.4
FAMNN-m 357 Nodes		**72.6**	**62.3**	**69.4**	**58.1**	**73.8**	**68.5**
FAMNN 657 Nodes		69.6	55.6	71.4	57.0	74.0	64.7
FAMNN-m 657 Nodes		**73.3**	**62.5**	**72.8**	**60.1**	**75.3**	**68.8**
FAMNN 1280 Nodes		70.2	56.7	71.3	56.7	76.2	67.4
FAMNN-m 1280 Nodes		**73.0**	**61.3**	**72.4**	**59.2**	**77.2**	**69.6**

(a)

		Brodatz Textures					
Type of Noise	**:**	**Gaussian**		**Uniform**		**Exponential**	
Standard Deviation	**:**	14.2	18.2	14.2	18.2	14.2	18.2
FAMNN 268 Nodes		72.4	55.7	63.6	47.1	81.0	66.2
FAMNN-m 268 Nodes		**81.7**	**65.4**	**72.4**	**55.5**	**86.8**	**74.3**
FAMNN577 Nodes		73.5	56.0	63.7	49.2	80.0	67.0
FAMNN-m 577 Nodes		**79.9**	**66.0**	**70.4**	**55.2**	**86.7**	**74.8**
FAM NN1348 Nodes		74.2	56.6	66.3	50.9	81.3	66.7
FAMNN-m 1348 Nodes		**79.8**	**63.4**	**72.9**	**56.4**	**87.1**	**74.3**
FAMNN2560 Nodes		74.2	57.1	66.3	50.3	81.0	67.5
FAMNN-m 2560 Nodes		**79.5**	**63.3**	**71.2**	**55.1**	**87.0**	**75.3**

(b)

TABLE II

PERCENTAGE OF CORRECT CLASSIFICATION OF THE NE-BASED FEATURE VECTORS: (a) EXTRACTED FROM THE 20 AERIAL TEXTURES, (b) EXTRACTED FROM THE 20 BRODATZ TEXTURES

NE –Vector		***Aerial Textures***								
Type of Noise	**:**	**Gaussian**			**Uniform**			**Exponential**		
Standard Deviation	**:**	14.4	21.6	24.4	14.4	21.6	24.4	14.4	21.6	24.4
FAMNN-127 Nodes		77.8	68.4	62.7	79.1	67.3	64.1	80.8	72.7	64.5
FAMNN-m 127 Nod.		**82.0**	**73.3**	**70.1**	**83.0**	**73.8**	**70.0**	**82.9**	**77.9**	**76.4**
FAMNN-440 Nodes		84.0	74.1	69.4	83.6	72.3	70.1	85.2	79.1	74.0
FAMNN-m 440 Nod.		**84.6**	**78.0**	**74.1**	**84.0**	**76.3**	**74.7**	**84.9**	**82.2**	**79.2**
FAMNN-1231 Nodes		82.4	70.6	66.2	82.0	69.6	66.9	83.5	75.5	70.6
FAMNN-m 1231 N.		**84.5**	**72.8**	**68.6**	**82.5**	**71.0**	**68.8**	**83.9**	**77.9**	**73.6**

(a)

		Brodatz Textures								
Type of Noise	**:**	**Gaussian**			**Uniform**			**Exponential**		
Standard Deviation	**:**	14.4	21.6	28.8	14.4	21.6	28.8	14.4	21.6	28.8
FAMNN-169 Nodes		87.9	77.6	69.6	87.2	76.2	67.3	88.4	84.5	80.1
FAMNN-m 169 Nodes		**88.6**	**83.7**	**75.7**	**88.5**	**82.1**	**72.5**	**89.0**	**86.5**	**84.0**
FAMNN-431 Nodes		88.4	81.5	74.7	88.3	80.5	74.1	88.4	84.7	82.3
FAMNN-m-431 Nodes		**89.4**	**84.6**	**78.9**	**89.2**	**83.5**	**77.1**	**89.4**	**86.5**	**84.6**
FAM-1454 Nodes		90.0	86.4	80.4	89.7	85.1	76.0	89.7	88.2	86.3
FAMNN-m-1454 Nodes		**90.3**	**88.3**	**84.1**	**90.0**	**87.6**	**79.1**	**90.3**	**89.3**	**88.1**
FAMNN-2560 Nodes		89.4	84.8	78.7	88.8	82.8	74.6	89.7	87.1	83.8
FAMNN-m-2560 Nodes		**90.0**	**86.6**	**82.4**	**89.6**	**85.6**	**77.6**	**89.9**	**88.7**	**86.8**

(b)

FAMNN when the NE-based feature set is used, when the Brodatz texture set is corrupted by Gaussian, uniform and exponential noise. The values for standard deviation of noise were equal to 14.2, 21.6, and 24.4 and the correspond to SNR of 18 dB, 10 dB, and 8 dB, respectively. Four sizes of FAMNN and FAMNN-m were examined: networks with 169 nodes, 431 nodes, 1454 nodes, and 2560 nodes.

Fig. 10 shows the classification improvement of FAMNN-m over FAMNN when the FD-based feature set is used, when the aerial texture set is corrupted by Gaussian, uniform, and exponential noise. The values for standard deviation of noise were equal to 14.4 and 18.8, and they correspond to SNR of 17 dB and 10 dB, respectively. Three sizes of FAMNN and FAMNN-m were examined: networks with 357 nodes, 657 nodes, and 1280 nodes. Fig. 11 shows the classification improvement of FAMNN-m over FAMNN for the FD-based feature set, when the Brodatz texture set is corrupted by Gaussian, uniform, and exponential noise. The values for standard deviation of noise were equal to 14.4 and 18.8 and they correspond to SNR of 17 dB and 10 dB respectively. Four sizes of FAMNN and FAMNN-m were examined: networks with 268 nodes, 577 nodes, 1348 nodes, and 2560 nodes.

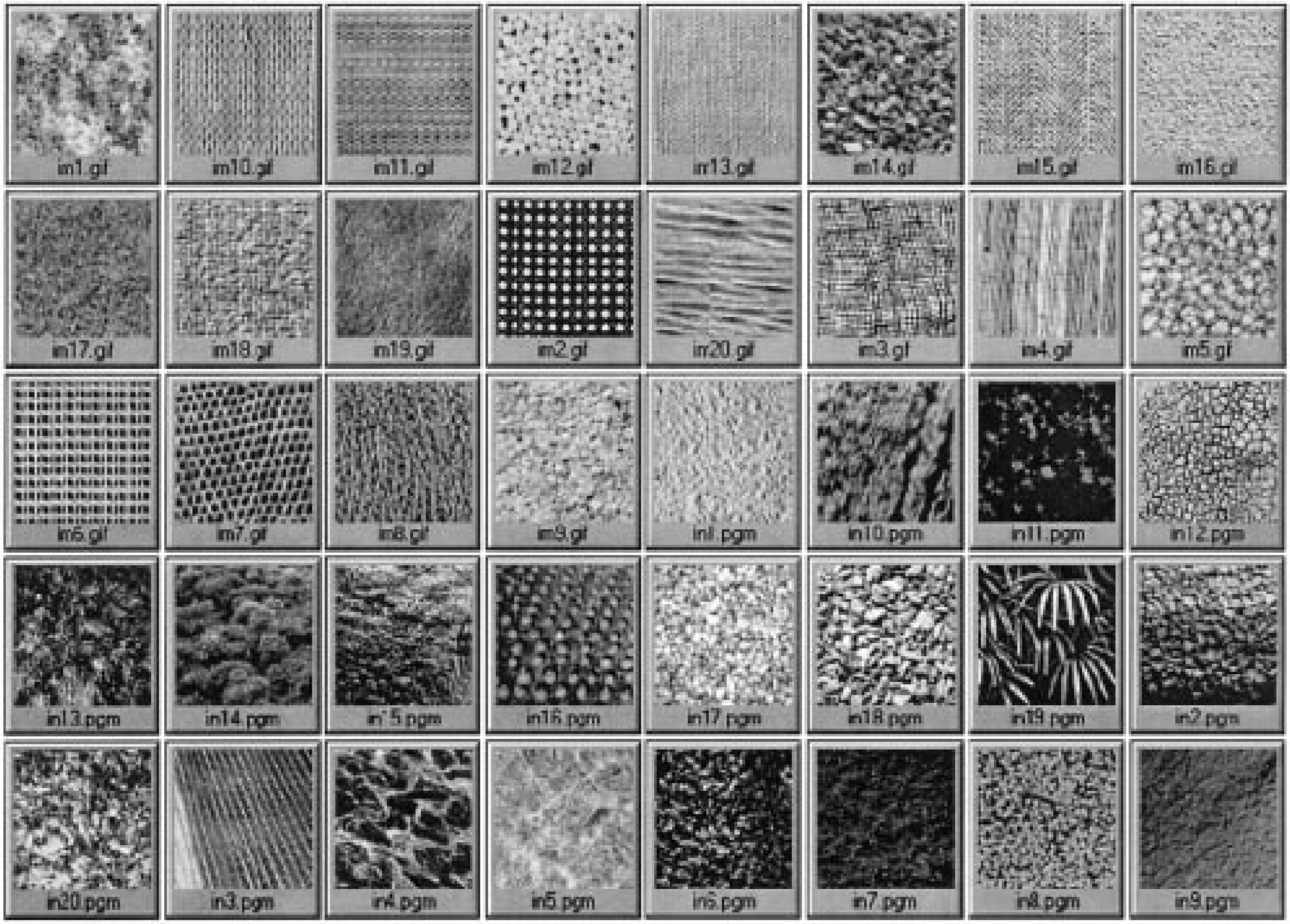


Fig. 12. Samples from the Brodatz textures (named im$*.*$) and the aerial textures (named in$*.*$).

In Fig. 12 we present samples of the textures used for classification. The images with name im$*.*$ are textures from the Brodatz texture set, while the images with name in$*.*$ are textures from the aerial texture set. In Fig. 13(a), we present a sample of the original texture, and in Fig. 13(b) the same texture corrupted by additive uniform noise of standard deviation 28.8.

Generally, in applications where the data are noisy, it is a common practice to train NNs with the noisy data. We have examined the effect of this approach in our application by training the NN with patterns extracted from the best quality textures available, and with patterns extracted from textures affected by noise with standard deviation 7.2. We used the energy feature vector. The resulting training set consisted of 2560 patterns.

We have compared the PCC obtained from this approach (training the NN with noisy data) with the PCC obtained from our modification. In order to make the comparison as fair as possible, we obtained in both cases the "optimum" network size. We made sure the PCC for the best quality test set available was the highest one. Furthermore, the value of γ was set equal to the average value of all γ_{i}s that are estimated from the best fit of the lines $(\mathbf{I}_{\mathrm{i}}^{a} - \mathbf{I}_{\mathrm{i}}^{N}) = (1-\gamma_{\mathrm{i}})(\mathbf{I}_{\mathrm{i}}^{\mathrm{b}} - \mathbf{I}_{\mathrm{i}}^{N})$ for standard deviation of noise equal to 7.2. We used a test set consisting of 2560 patterns, which is a small sample of the test set used in the previous experiments. The comparison results are presented in Table III. We notice that the PCC for both cases is almost equal when the test set is extracted from the original textures. The improvement of our modification in terms of PCC over the NN trained with noisy data is not as much as in the case where the NN is trained with the best quality data available. Nevertheless, the improvement is in some cases more than 2%. This result shows that training the NNs with noisy data improves the PCC but not as much as our method does.

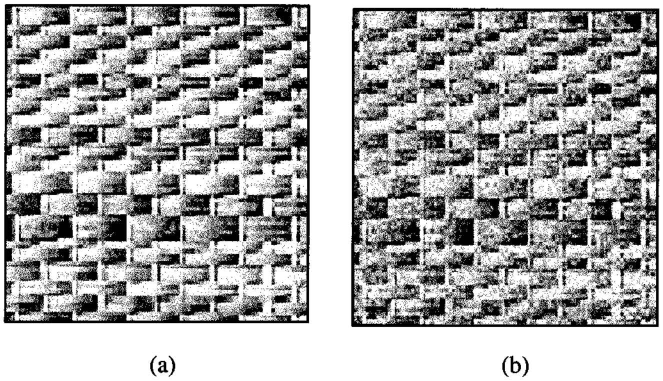


Fig. 13. (a) Best quality texture (b) texture where white noise with standard deviation 24.4 has been added to the best quality texture.

Furthermore, the number of nodes created after training with both best quality data available and noisy data is larger (368

TABLE III
PERCENTAGE OF CORRECT CLASSIFICATION FOR FAMNN-m TRAINED WITH THE BEST QUALITY TEXTURES, AND FAMNN WHEN IT IS TRAINED WITH BEST QUALITY TEXTURES AFFECTED BY NOISE

NE –Vector		**Aerial Textures**			
Standard Deviation of noise:		**nonoise**	**14.4**	**21.6**	**24.4**
Gaussian Noise	FAMNN (trained with noisy data)	78.1	74.5	66.7	60.2
	FAMNN-m	**78.0**	**74.9**	**68.0**	**60.9**
Uniform Noise	FAMNN (trained with noisy data)	78.1	75.2	65.5	59.6
	FAMNN-m	**78.0**	**76.3**	**66.7**	**61.6**
Exponential Noise	FAMNN (trained with noisy data)	78.1	75.5	68.9	64.1
	FAMNN-m	**78.0**	**76.9**	**71.2**	**66.7**

nodes) compared to the number of nodes when the network is trained with only the best quality data (200 nodes). An immediate consequence of creating more nodes is that the training phase is more time consuming. Another advantage of our method is flexibility, since only one parameter, γ, needs to be estimated in order to obtain the "highest possible" PCC for the noisy textures. Therefore, the networks do not have to be retrained. If we train the network with patterns extracted from noisy textures, we require a new training process each time we need to obtain the "highest possible" PCC for the noisy textures.

V. SUMMARY AND CONCLUDING REMARKS

In this paper we introduced a variation of the testing phase of the FAMNN that we named FAMNN-m. We demonstrated that FAMNN-m exhibits superior generalization performance compared to the generalization performance of FAMNN in the classification of noisy signals. These results are valid independently of the type of noise affecting the signals. Moreover, the results are valid independently of the size of the ART architectures created. The introduced modification of FAMNN was based on the fact that values of signal features which are distant from feature values that correspond to a pure noise signal, are affected more severely than values of signal features that are close. The proposed modification of FAMNN is especially suited for applications where it is required that the feature set captures only the shape characteristics of the signal and not its actual amplitude or average value. If the variance of the noise that contaminates the signals is estimated, then the classification results of FAMNN-m could be further improved by producing a more accurate estimate of the parameter γ. The application used in this paper, to illustrate the points above, is classification of noisy textured images.

REFERENCES

[1] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representation by error propagation," in *Parallel Distribution Processing: Explorations in the Microstucture of Cognition*. Cambridge, MA: MIT Press, 1986, vol. 1, ch. 8.

[2] J. E. Moody and C. J. Parker, "Fast learning in networks of locally tuned processing units," *Neural Comput.*, vol. 1, pp. 281–294, 1989.

[3] G. A. Carpenter, S. Grossberg, N. Markuzon, J. H. Reynolds, and D. B. Rosen, "Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps," *IEEE Trans. Neural Networks*, no. 3, pp. 698–713, 1992.

[4] P. K. Simpson, "Fuzzy min–max neural networks—Part 1: Classification," *IEEE Trans. Neural Networks*, vol. 3, pp. 698–713, Sept. 1992.

[5] M. J. Healy, T. P. Caudell, and S. D. G. Smith, "A neural-network architecture for pattern sequence verification through interferencing," *IEEE Trans. Neural Networks*, vol. 4, pp. 9–20, 1993.

[6] G. A. Carpenter and W. D. Ross, "ART-EMAP: A neural network architecture for object recognition by evidence accumulation," *IEEE Trans. Neural Networks*, vol. 6, pp. 805–818, 1995.

[7] J. R. Williamson, "Gaussian ARTMAP: A neural network for fast incremental learning of noisy multidimensional maps," *Neural Networks*, vol. 9, no. 5, pp. 881–897, 1996.

[8] A. K. Jain and F. Farrokhinia, "Unsupervised texture segmentation using Gabor filters," *Pattern Recognition*, vol. 24, no. 12, pp. 1167–1185, 1991.

[9] D. Dunn and W. E. Higgins, "Optimal Gabor filters for texture segmentation," *IEEE Trans. Image Processing*, vol. 4, pp. 947–964, July 1995.

[10] A. Teuner, O. Pichler, and B. J. Hosticka, "Unsupervised texture segmentation of images using tuned matched Gabor filters," *IEEE Trans. Image Processing*, vol. 4, pp. 863–870, June 1995.

[11] R. Bajscy, "Computer identification of visual surfaces," *Comput. Graphics Image Processing*, vol. 2, pp. 118–130, 1973.

[12] P. C. Chen and T. Pavlidis, "Segmentation by texture using correlation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-5, pp. 64–69, Jan. 1983.

[13] M. Unser, "Texture classification and segmentation using wavelet frames," *IEEE Trans. Image Processing*, vol. 4, pp. 1549–1560, Nov. 1995.

[14] R. N. Strickland, "Wavelet transform methods for image detection and recovery," *IEEE Trans. Image Processing*, vol. 6, pp. 724–734, May 1997.

[15] B. B. Chaundhuri and N. Sarkar, "Texture segmentation using fractal dimension," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, pp. 72–77, Jan. 1995.

[16] B. Dubuc, C. Roques-Carmes, C. Tricot, and S. W. Zucker, "The variation method: A technique to estimate the fractal dimension of surfaces," *SPIE, Visual Commun. Image Processing II*, vol. 845, pp. 241–248, 1987.

[17] T. Kasparis, N. S. Tzannes, M. Bassiouni, and Q. Chen, "Texture description based on fractal and energy features," *Comput. Elect. Eng.*, vol. 21, no. 1, pp. 21–32, 1995.

[18] J. Garding, "Properties of fractal intensity surfaces," , pp. 319–324, 1988.

[19] A. P. Pentland, "Fractal based description of natural scenes," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 6, pp. 661–674, Nov. 1984.

[20] D. Charalampidis, T. Kasparis, and J. Rolland, "Segmentation of textured images based on multiple fractal feature combinations," *Proc. SPIE, Visual Inform. Processing VII*, pp. 25–37, 1998.

[21] V. Murino, C. Ottonello, and S. Pagnan, "Noisy texture classification: A higher-order statistics approach," *Pattern Recognition*, vol. 31, pp. 383–393, 1998.

[22] M. Georgiopoulos, H. Fernlund, G. Bebis, and G. L. Heileman, "Order of search in fuzzy ART and fuzzy ARTMAP: Effect of the choice parameter," *Neural Networks*, vol. 9, no. 9, pp. 1541–1559, 1996.

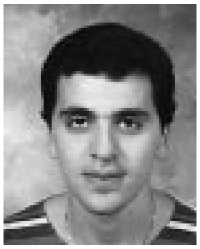
[23] P. Brodatz, *Textures: A Photographic Album for Artists and Designers*. New York: Dover, 1966.

[24] Univ. Berkeley., Berkeley, CA. [Online]. Available: http://www.arch.ced.berkeley.edu/kap/

[25] D. Charalampidis, T. Kasparis, and M. Georgiopoulos, "Classification of noisy patterns using ARTMAP-based neural networks," in *Proc. SPIE*, Orlando, FL, 2000.

[26] ——, "Texture classification using ART-based neural networks and fractals," in *Proc. SPIE*. Orlando, FL: AeroSense, Ap. 1998.

[27] D. R. Wilson and T. R. Martinez, "Reduction techniques for instance-based learning algorithms," *Machine Learning*, vol. 38, pp. 257–286, 2000.



Dimitrios Charalampidis (S'99) received the Diploma degree in electrical engineering and computer technology from the University of Patras, Patras, Greece, in 1996 and the M.S. degree in electrical engineering from the University of Central Florida, Orlando, in 1998. He is currently pursuing the Ph.D. degree in the School of Electrical Engineering and Computer Science at University of Central Florida.

His research interests include image processing, digital signal processing, neural networks, pattern recognition, and applications of signal processing to remote sensing. He has published more than ten papers in journals and conferences. Currently, his research emphasis is on neural-network algorithms and image segmentation and classification.

Mr. Charalampidis is a member of the IEEE Signal Processing Society.

Takis Kasparis (S'82–M'85) received the Diploma degree in electrical engineering from the National Technical University of Athens, Athens, Greece, in 1980 and the M.E.E.E. and Ph.D. degrees in electrical engineering from the City College of New York, New York, in 1982 and 1988, respectively.

From 1985 to 1989, he was an Electronics Consultant. In 1989, he joined the Electrical Engineering Department of the University of Central Florida, Orlando, where he is presently an Associate Professor. His research interests include digital signal and image processing, texture analysis, and computer vision.

Dr. Kasparis has served on the steering committee of several conferences and he is currently an Associate Editor for *Pattern Recognition*. He is a member of the Technical Chamber of Greece.

Michael Georgiopoulos (S'82–M'83) received the Diploma degree in electrical engineering from the National Technical University of Athens in 1981. He also received the M.S. and Ph.D. degrees in electrical engineering from the University of Connecticut, Storrs, in 1983 and 1986, respectively.

In 1987, he joined the University of Central Florida, Orlando, where he is currently an Associate Professor in the School of Electrical Engineering and Computer Science. He has been conducting research in neural networks and applications of neural networks for over ten years, and he has published more than 140 papers in journals, conferences, and books. He has also coauthored a book entitled *Applications of Neural Networks in Electromagnetics* (Boston, MA: Artech, 2000). He has worked on a variety of research topics throughout his career, including communication networks, spread-spectrum communications, neural networks and applications of neural networks in computer-generated forces modeling, smart antennas, pattern recognition and image processing, electromagnetics, computer vision, manufacturing, and remote sensing. Currently, his research emphasis is on neural-network algorithms (with special emphasis on ART neural-network architectures), design of smart antennas using neural networks, and modeling of computer-generated forces using neural networks and symbolic techniques.

Dr. Georgiopoulos is currently an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS. He is a member of the International Neural Network Society and a member of the Technical Chamber of Greece. He has served as the Technical Program Chair of the 1996 Southcon conference, and he has also served as the program committee member and session chair of several international neural network conferences.