# A Fuzzy ARTMAP Based Classification Technique of Natural Textures

Dimitrios Charalampidis
University of Central Florida
Electrical and Computer Engineering
Orlando, Florida 32816
dc19339@pegasus.cc.ucf.edu

Takis Kasparis
University of Central Florida
Electrical and Computer Engineering
Orlando, Florida 32816
kasparis@pegasus.cc.ucf.edu

Michael Georgiopoulos
University of Central Florida
Electrical and Computer Engineering
Orlando, Florida 32816
michaelg@pegasus.cc.ucf.edu

Jannick Rolland
University of Central Florida
Center for Research and Education in Optics
Orlando, Florida 32816
rolland@creol.ucf.edu

## Abstract

*This paper describes an approach to classification of textured grayscale images using a technique based on image filtering and the fractal dimension (FD) and the Fuzzy ARTMAP neural network (FAMNN). Twelve FD features are computed based on twelve filtered versions of the original image using directional Gabor filters. Features are computed in a window and mapped to the central pixel of this window. We implemented a variation of the testing phase of Fuzzy ARTMAP that exhibited superior performance than the standard Fuzzy ARTMAP and the 1-nearest neighbor (1-NN) in the presence of noise. Training was performed using patterns that were extracted from twenty different textures. The performance of classification is also studied with respect to a testing set. Segmentation results are also presented to illustrate that the classification algorithm and its specified parameters are adequate so that more than one texture can be identified in the same image.*

## 1. Introduction

Classification is an important component of texture analysis. 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 texture. Usually, a feature set needs to contain more than one feature to successfully characterize a textural region. Gabor energy ([1]-[3]) and fractal dimension (FD) ([4]-[6]) are two features that are tested here. Neural networks (NN) are often used for classification purposes. Here, the Fuzzy ARTMAP neural network (FAMNN) [7] is examined. Another classification method, the 1-Nearest Neighbor (1-NN) technique is also examined.

A common problem in texture analysis is corruption of textures by noise. In many cases the assumption that the noise is white is a valid one. In this work we implemented a variation of FAMNN that exhibits superior performance than the standard FAMMN and 1-NN when the textures are corrupted by white noise. FAMNN is used because of its fast training phase and its interesting geometric interpretation. Results with respect to a testing set are important since in most cases the textures that will be tested are similar but not exactly the same as the ones that have been used for training. Noisy versions of the training set can also be thought as testing set. The objective here is to improve the performance of classification when the textures are affected by noise while preserving good classification performance for a different testing set. Segmentation results are also presented to illustrate that the classification algorithm and its specified parameters are adequate so that more than one texture can be identified in the same image.

## 2. Background

FAMNN has been introduced and discussed extensively in [7]. In this section we only present some important points of FAMMN that will make clearer the effect of our modification. The FAMNN [7] maps the input patterns, which in our case are the feature vectors of size M, to a label. This operation is performed through the creation of weight vectors $\mathbf{w}_j = \{w_{j1}, \ldots, w_{j2M}\}$, $j = 1, \ldots N$, which are called templates. N is the number of necessary templates so that all the training input patterns are correctly labeled. Each template is associated to a label. Different templates can be associated to the same label. The templates $\mathbf{w}_j$ that are formed during training are compressed representations of the training input patterns. A template $\mathbf{w}_j$ 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 lower and upper endpoints; the lower endpoint is the hyperbox point with the smaller coordinates while the upper endpoint is the point with the largest coordinates.

FAMNN operates in two distinct phases: the training phase and the testing phase. For the training phase, given a list of training input/label pairs, such as $\{\mathbf{I}^1,\mathbf{O}^1\}$, $\{\mathbf{I}^2,\mathbf{O}^2\},\ldots,\{\mathbf{I}^{MP},\mathbf{O}^{MP}\}$, we want to train 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 architecture until the desired mappings are established.

For the testing phase, initialize the values of the templates to the values that they had at the end of the training phase. Then a test pattern will choose to be represented by a template that best matches it according to the FAMNN rules (for more details see [7]). The chosen template will map this input pattern to a distinct label.

## 3. Classification Method

The classification method includes two stages. The first stage is feature extraction. Feature extraction consists of two phases: The first phase is fitering using directional symmetric and real-valued Gabor filters with three different center frequencies (0, 0.05 and 0.1) and four orientations ($0^\circ$, $45^\circ$, $90^\circ$, $135^\circ$). The second phase is computation of the FD space for all filtered versions of the image. The FD values that are computed over a window W centered at the pixel with coordinates (x, y) for all the twelve filtered versions of the image, are mapped to pixel with coordinates (x, y) of the original image. These twelve FD values comprise the feature set. In our previous work [8], this feature set has shown good performance for texture segmentation. In the second stage the neural network is used for training or testing. The training phase is exactly the same as the training phase of FAMNN. The testing phase is a modification of the testing phase of FAMNN that exhibits superior performance than the standard FAMNN if the textures are affected by noise. Also, we examine the case where an image can contain more than one textures. For this reason, the testing phase is iteratively applied so that areas of the image that are considered to be boundaries between different textures are further examined. We denote the modified FAMNN as FAMNN-m. The block diagram of the classification technique is shown in Figure 1.

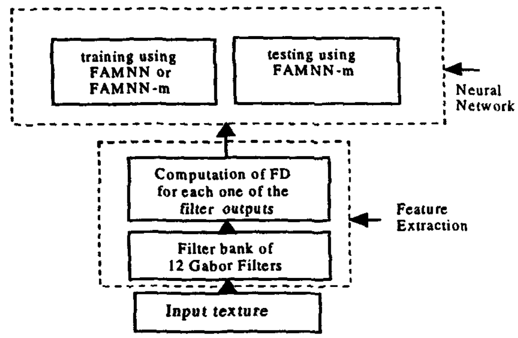



**Figure 1. Block diagram of classification method**

### 3.1 Important Observations

The approach we follow takes into account the fact that the FD of an image tends to increase as the variance of the noise that it is affected by, increases. This fact has been shown experimentally, and one representative example is shown in Figure 2 for Gaussian and Uniform white noises. In our experiments we do not use noise of standard deviation more than 24.5.

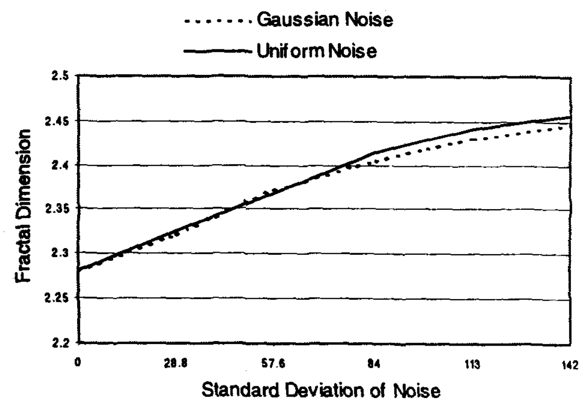



**Figure 2. FD versus standard deviation of noise**

In Figure 2, even though the FD has changed only 0.05 as the standard deviation of noise has increased to 24.5, this change is significant since the minimum and maximum values of FD for a two dimensional surface are 2 and 3 respectively.

The classification approach is based on the assumption that smaller FD values tend to increase more than larger FD values. This assumption seems reasonable since small FD means that the surface of the image is smooth. If this image is affected by additive white noise, then it will be rougher. If FD has a large value, then the surface of the image is already rough so that additive noise will not affect it so much in that respect.

The FAMNN is used because it is a fast algorithm that converges in a small number of iterations. The training process is less time consuming than the training process of other common neural networks. The testing phase of FAMNN is less time consuming than nearest neighbor techniques since only a compressed version of the original training data information is used.

### 3.2 Training

The networks were trained using 5120 feature vectors. These have been extracted from 20 images of size 256 x 256. The first step is smoothing of the features by averaging their values over a window so that their robustness to describe texture is increased. The smoothing window has size 33 x 33. The feature vector that corresponds to pixel with coordinates x, y is the smoothed feature vector over the window which is centered at x, y.

The second step is to select the feature vector that is mapped to the pixel that is located at the center of a window of size 16 x 16. If all non-overlapping windows are considered, then we have a total of 256 feature vectors selected from each image. This sampling of the set of the feature vectors is necessary so that the size of the training set is decreased without loosing important information since the values of the features that correspond to pixels that are close, are similar especially after smoothing of the features has taken place.

### 3.3 Testing

We have implemented a variation of the testing phase of FAMNN, referred to as FAMNN-m. The template chosen in FAMNN to represent a test input pattern is the one that maximizes the bottom-up input function and satisfies the vigilance criterion (more details about the bottom-up input function and the vigilance criterion can be found in [7]). The bottom-up input for every template in FAMNN-m is modified so that preference is given to templates whose corresponding hyper-boxes have upper endpoints with smaller coordinates, since the input patterns that were encoded by them have smaller elements so they are expected to increase more in the presence of noise, tending to be mapped to hyper-boxes with larger coordinates.

The modification of the bottom-up input function in FAMNN-m is expressed as multiplication of the bottom-up input function of FAMNN by

$$G(F) = \frac{M}{M + aF} \tag{1}$$

where $M$ is equal to the number of elements of the feature vectors. Also, $F$ is the summation of the coordinates of the upper corner of the hyper-box that is defined by the template, and it is equal to:

$$F = \sum_{i=M+1}^{2M} (1 - w_{ji}) \tag{2}$$

and $\alpha$ is a constant that has the largest possible value, so that the correct classification of the training set will not be less than 99.5%. This value was experimentally found to be equal to 0.3. The function $G(F)$ is a non-linear function of $F$ and does not change very rapidly if $F$ is large. This is desired since the FD for the pure noise surface might be large but not larger than a certain value.

It has been shown [8] that a method that uses variable window width for smoothing and feature extraction, gives very good segmentation results since there is more accurate identification of textures at the inner regions, while the boundaries between textures are not blurred significantly. A similar approach is also used here for all classification methods, since we also consider that an image may contain more than one textures. The steps of the algorithm are the following:

Step 1. Smoothing of the features by averaging their values in a square window of size 33 x 33 and map the result to its central pixel.

Step 2. Apply the testing phase of the classification system for an initial estimation of classes.

Step 3. A sliding window of size 33 x 33 merges small regions to large regions. This window classifies the feature vector that is mapped to its central pixel to class j, if the number of feature vectors that are associated to class j, is larger than the number of features that are associated to any other class.

Step 4. Classification remains unchanged for the feature vectors that correspond to pixels that are further away from the boundaries for a distance D that is equal to half the width of the smoothing window. The reason is that the smoothing window will not blur the boundaries for more than D on one and D on the other side. The rest of the pixels are considered as an ambiguous class and they will be further examined.

Step 5. Apply the procedure to the ambiguous class iteratively, by reducing the size of the smoothing and merging windows, so that the new ambiguous class is estimated.

For the experiments four iterations were used. The smoothing and merging window sizes were 33 x 33, 25 x 25, 17 x 17 and 9 x 9.

## 4. Experimental Results

FAMNN-m is compared with the standard FAMNN with total number of templates equal to 357 and 746. FAMNN-m is also compared to the 1-NN algorithm. The described iterative classification is applied in all classification methods. All four methods are compared with respect to classification of a testing set and the classification of the training set when it is affected by uniform white noise. FAMNN and 1-NN are compared with respect to the segmentation of images that consist of textures selected from the training set. The set that is referred as training set is the twenty textures from which the feature vectors that were used to train the networks are created.

The testing set consists of twenty textures. Each texture of the testing set is a different realization of the corresponding texture of the training set. The percentage of correct classification (PCC) for the testing set is given in Table 1.

| | *1-NN* | *FAMNN 367 templates* | *FAMNN 746 templates* | *FAMNN-m 746 templates* |
|---|---|---|---|---|
| PCC (%) | 96.2 | 95.1 | 96.2 | 94.7 |

**Table 1. Percentage of correct classification for all methods, for noisy free textures that are a different realization of training textures**

In this case all algorithms perform well with PCC close to 95%. The difference between the best and the worst PCC is 1.5 %. This difference is not significant especially because the selection of the testing set was based on visual estimation of similarity with the training set. In Figure 3 three of the textures of the training set and the corresponding ones from the testing set are presented.

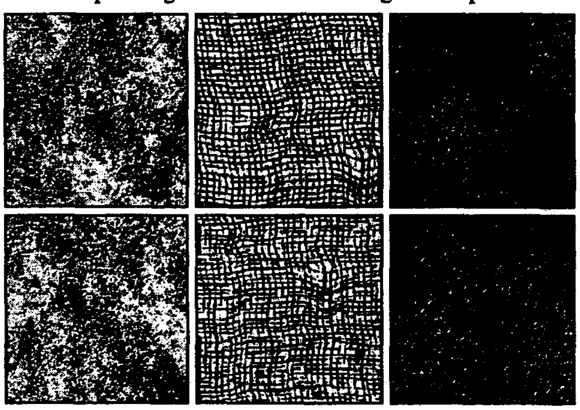


**Figure 3. Textures used for training set (above) and testing set (below)**

The four algorithms are compared with respect to the classification performance in the case where the textures are affected by additive uniform noise. The results are presented in Table 2 and in Figure 4. In the absence of noise and when the standard deviation of the noise is 7.2 the PCC is similar for all methods and it is close to 100%. The PCC is larger for FAMNN-m if the standard deviation is 13, 18.8 and 24.5. The PCC of FAMNN-m is larger than the PCC of the 1-NN and the difference is almost constant and close to 4.5% for these cases. The difference between the PCC FAMNN-m and the other two methods is even larger and it increases as the standard deviation of the noise increases.

| | *noise Free* | *st.dev 7.2* | *st.dev 13.5* | *st.dev 18.8* | *st.dev 24.5* |
|---|---|---|---|---|---|
| *1-NN* | 100.0 | 100.0 | 94.1 | 85.2 | 67.4 |
| *FAMNN 367 templates* | 99.9 | 99.4 | 91.5 | 77.9 | 52.5 |
| *FAMNN 746 templates* | 100.0 | 99.8 | 93.3 | 78.3 | 57.9 |
| *FAMNN-m 746 templates* | 99.6 | 99.7 | 98.3 | 89.1 | 72.1 |

**Table 2. Percentage of correct classification of the 20 textures that were used for training when they are affected by white noise**

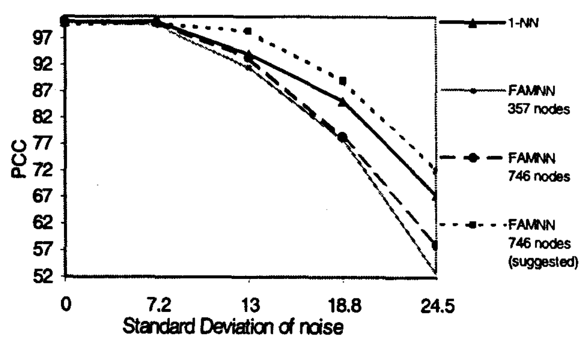
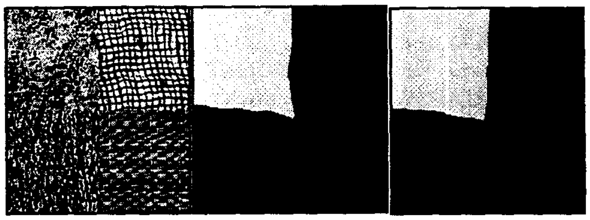



**Figure 4. Percentage of correct classification of the 20 textures that were used for training when they are affected by white noise**

The segmentation performance was also tested for the 1-NN and for the proposed method. One result is presented in Figure 5. The purpose of this experiment is to show that the smoothing window and the window that is used for feature extraction are sufficiently large, so that more than one texture in the same image can be identified. It has been shown experimentally that iterative algorithms that use different sizes of smoothing window can help to avoid blurring at the regions close to the boundaries between textures. Here it is shown that there is about 2-3% misclassification at the boundaries because of the existence of more than one different texture in the same image.



(a) (b) (c)

**Figure 5. Segmentation of an image consisting of 4 textures from the training Set. (a) Original image, (b) Segmented image using 1 - NN, (c) Segmented image using the FAMNN - m**

Only the effect of uniform white noise was tested since FD values are almost identical when the images are affected by uniform or Gaussian white noise as it is shown in Figure 3. This result is due to the fact since FD is related to frequency as it is shown in Figure 2. White noise, independently of its exact form, has a flat power spectrum. Even though FD does not depend only on frequency it is very much related to it, especially when we are dealing with completely random textures.

## 5. Conclusions

In this paper a FAMNN variation is proposed. A modification of the input that defines the cluster, in which an input feature vector belongs, improves the PCC in the case where the textures are affected by noise. This modification is feature dependent since it takes advantage of the characteristic that the FD feature values increase in the presence of noise. The PCC of FAMNN-m is larger than the original FAMNN and the 1-NN in a noisy environment. In the absence of noise the performance of FAMNN-m is similar to the performance of the standard FAMNN. If the variance of the noise that contaminates the textures could be estimated, then the FAMNN could adapt so that the classification results are further improved.

Also, the segmentation results show that the different parameters such as the size of the smoothing window and the size of the window where the feature vectors are computed are sufficient with the help of the iterative algorithm, so that different textures can be identified in the same image.

## 6. References


[1] Anil K. Jain, and Farshid Farrokhinia, "Unsupervised texture segmentation using Gabor filters", *Pattern Recognition*, Vol. 24, No 12, pp. 1167-1185, 1991.

[2] Dennis Dunn, and William E. Higgins, "Optimal Gabor filters for texture segmentation", *IEEE Transactions on image processing,* Vol. 4, No. 7, pp. 947-964, July 1995.

[3] Andreas Teuner, Olaf Pichler and Bedrich J. Hosticka, "Unsupervised texture segmentation of images using tuned matched Gabor filters", *IEEE Transactions on image processing,* Vol. 4, No. 6, pp. 863-870, June 1995.

[4]B.B Chaundhuri, and Nirupam Sarkar, "Texture segmentation using fractal dimension", *IEEE Transactions on pattern analysis and machine intelligence,* Vol. 17, No. 1, pp. 72-77, January 1995.

[5] B. Dubuc, C. Roques-Carmes, C. Tricot, and S. W. Zucker, "The variation method: a techique to estimate the fractal dimension of surfaces", *SPIE, Vol. 845, Visual Communications and image processing II*, pp. 241-248, 1987.

[6] T. Kasparis, N.S. Tzannes, M. Bassiouni and Q. Chen, "Texture Description based on Fractal and Energy Features", *Computers and Electrical Engineering, Vol. 21, No. 1, pp 21-32, 1995.*

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

[8] Charalampidis, D. , Kasparis, T., and Rolland, J., "Segmentation of textured images based on multiple fractal feature combinations", *Proceedings of SPIE, Visual Information Processing VII*, pp. 25-35, 1998.