

Land Cover Classification from SPOT Multispectral And Panchromatic Images Using Neural Network Classification of Fuzzy Clustered Spectral and Textural Features

By George Lemeschewsky and Ann Frazier

Open-File Report 95-651

U.S. Department of the Interior
U.S. Geological Survey
National Mapping Division

Land Cover Classification from SPOT Multispectral And Panchromatic Images Using Neural Network Classification of Fuzzy Clustered Spectral and Textural Features

George Lemeshefsky and Ann Frazier
U.S. Geological Survey
521 National Center
Reston, VA 22092
Open-File Report 95-651

ABSTRACT

A technique is described for doing land cover classification using a neural network to integrate and classify SPOT multispectral and derived texture data. Orientated texture energy was derived from the higher spatial resolution SPOT panchromatic band with directional spatial filtering techniques. The multispectral and textural data were each clustered using a reported fuzzy learning vector quantization technique, which also produced fuzzy segmented images. The fuzzy membership values of data to clusters was then input to a multilayer neural network that had been trained using training samples of the land cover classes. Using an output layer that scales individual outputs to the range from zero to one and their sum to one, allows interpretation of the neural network outputs as class probabilities and the potential for post classification improvements by probabilistic relaxation techniques. Initial land cover classification results for neural network test and training data in a mixed rural-urban area are promising.

Any use of trade, product, or firm names is for descriptive purposes only and does not imply endorsement by the U.S. Government.

I. INTRODUCTION

This is an interim report on an experimental technique for doing land cover classification by application of fuzzy clustering and segmentation techniques to multisource, remotely sensed images, and the classification of the fuzzy data by a trained neural network (NN). The multisource data includes SPOT multispectral (XS) images and texture features derived from the higher resolution SPOT panchromatic (PAN) image. Directional filtering techniques were applied to the SPOT PAN image to obtain measures of orientated texture energy, as described in section II. A fuzzy learning vector quantization clustering procedure was then applied to both the derived texture and the SPOT XS principal component images. To help visualize the results of this procedure, a reported technique for mapping fuzzy membership function values to color images was applied. These procedures and their results are described in section III. The fuzzy membership values of the XS and texture feature data were then input to a three layer, feedforward NN, which was trained as a classifier. By using a NN with "softmax" output layer (Bridle, 1990), individual outputs can be interpreted as the probability of land cover class, (section IV). Section V describes the collection of test and training data used to train the NN, the classification results, and proposed postclassification processing with a reported probabilistic relaxation technique. The conclusions on this preliminary work are included in section VI.

In the following, image data are the georeferenced, cubic interpolated to 10-m sample distance SPOT PAN and XS images. The study region, 5,120 by 5,120m, is centered on Chestertown, Md., and includes mixed urban and agricultural areas.

II. TEXTURE ENERGY

Laws (1980) defined texture energy as the amount of variation within a spatial image window and described image segmentation tests based on texture energy. Convolving the image with various combinations of horizontal and vertical spatial filters produced directional filtered images. Orientated texture energy was computed as the local spatial average of the absolute value of the filtered images.

In this study, texture features were derived from a 10-m ground sample distance SPOT PAN image. A measure of spatial strength was derived from the squared output (that is orientated energy) of four fixed orientation, quadrature pair bandpass filters at $\theta = 0, 45, 90,$ and 135 degrees as described by Freeman and Adelson (1991).

For this study, and merely to reduce numerical range, orientated texture data $T(\theta)$ was calculated from orientated energy $E(\theta)$ as $T(\theta) = E(\theta)^{1/2}$. Orientated energy $E(\theta)$ is (Freeman and Adelson, 1991)

$$E(\theta) = [G(\theta)]^2 + [H(\theta)]^2 \quad (1)$$

where θ was one of the above fixed orientations and spatial filters $G(\theta)$ and $H(\theta)$ were derived from the fourth derivative of a Gaussian and its Hilbert transform using their filter coefficient tables. $G(\theta)$ and $H(\theta)$ were computed from x-y separable, 13 tap spatial filters.

A fifth, nondirectional measure of texture energy was computed as the average

$$T(0^\circ-360^\circ) = (1/4) ([T(0^\circ)] + [T(45^\circ)] + [T(90^\circ)] + [T(135^\circ)]). \quad (2)$$

To reduce the dependence of local image intensity on texture energy, filtered images $T(0^\circ)$, $T(45^\circ)$, $T(90^\circ)$, and $T(135^\circ)$ were divided by $T(0^\circ-360^\circ)$. Applying a 7 by 7 moving average window to the four normalized images and the $E(0^\circ-360^\circ)$ image resulted in five orientated texture feature images.

Texture data at any x-y location of all five images was considered a five element texture vector for subsequent clustering.

III. FUZZY CLUSTERING

The derived texture and SPOT XS principal component data were clustered by means of a fuzzy learning vector quantization (FLVQ) technique reported by Bezdek (1993). There are two steps to this clustering: (1) FLVQ to get cluster prototype vectors, and (2) calculation of fuzzy membership function values between the data and the cluster prototype vector.

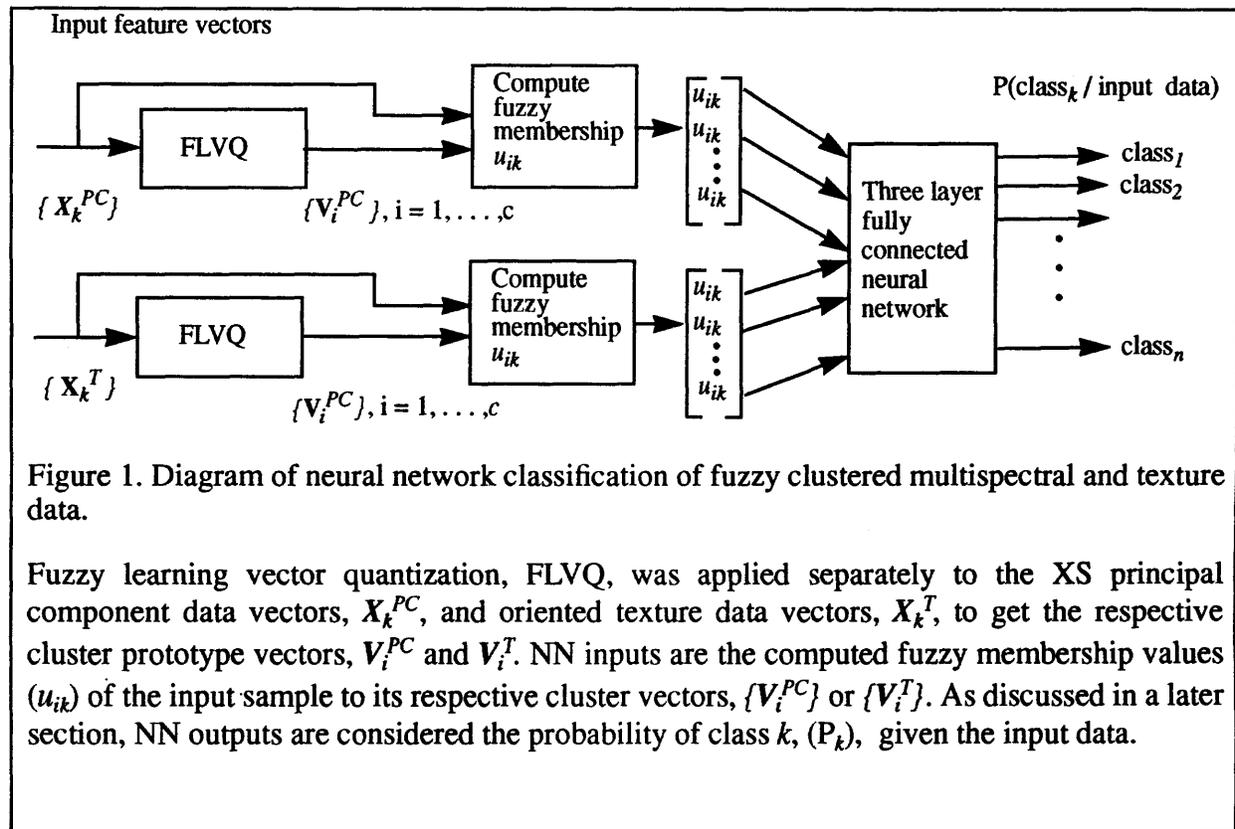


Figure 1. Diagram of neural network classification of fuzzy clustered multispectral and texture data.

Fuzzy learning vector quantization, FLVQ, was applied separately to the XS principal component data vectors, X_k^{PC} , and oriented texture data vectors, X_k^T , to get the respective cluster prototype vectors, V_i^{PC} and V_i^T . NN inputs are the computed fuzzy membership values (u_{ik}) of the input sample to its respective cluster vectors, $\{V_i^{PC}\}$ or $\{V_i^T\}$. As discussed in a later section, NN outputs are considered the probability of class k , (P_k), given the input data.

In this study each texture or XS principal components (PC) image was clustered separately. The PC transformation (Richards, 1993) was applied to the three band XS data to reduce the interband correlation. Clustering used the full dimensional XS-PC data. After clustering, NN inputs were the fuzzy membership values of texture and PC feature vectors to their respective predetermined clusters. Figure 1 is a diagram of the fuzzy clustering and NN classification technique used in this study. (Vectors are denoted in bold type).

The FLVQ technique by Bezdek (1993) is outlined below. See Bezdek's paper for a complete description and also a comparison with Kohonen learning vector quantization (LVQ).

Although there are similarities between the FLVQ and Kohonen's LVQ, one important advantage of FLVQ is that none of the heuristics associated with the LVQ training algorithms are needed (Bezdek, 1993). Another advantage of fuzzy clustering is that the results can be depicted as fuzzy segmented images (Hall and others, 1992). Fuzzy segmented XS and texture images were used as an interpretation guide to help the collection of network training data.

For FLVQ, the desired number, c , of cluster prototype vectors, V_i , where $i = 1, \dots, c$ and their initial values are first defined and then each V_i is repeatedly updated to minimize an error criterion. The vector update is calculated from the fuzzy membership, u_{ik} , between data vectors, X_k , where $k=1, \dots, n$, and cluster prototype vectors, V_i .

The update for V_i at iteration t is

$$V_{i,t} = V_{i,t-1} + \sum_{k=1}^n \alpha_{ik,t} (X_k - V_{i,t-1}) / \sum_{s=1}^n \alpha_{is,t}. \quad (3)$$

Learning rate, $\alpha_{ik,t}$, is computed from fuzzy membership, $u_{ik,t}$, for parameter m_t and D_{ik} is the Euclidean distance between V_i and X_k .

$$\alpha_{ik,t} = (u_{ik,t})^{m_t} = \left(\sum_{j=1}^c D_{ik,t} / D_{jk,t} \right)^{-2m_t / (m_t - 1)} \quad (4)$$

Parameter m_t is

$$m_t = m_0 + t[(m_f - m_0) / T] \quad (5)$$

where m_0 and m_f are predefined initial and final values. The number of updates, T , of cluster prototype vectors V_i is

$$t = 1, 2, \dots, T. \quad (6)$$

In this preliminary study V_i and NN training and test data were from the same 512 x 512 pixel image area. To reduce computations, FLVQ was applied to the principal component, X^{PC} , and texture energy images, X^T , after subsampling by a factor of 2 to get 256 x 256 pixel samples.

For the clustering process of figure 1, there were 10 PC cluster vectors, V_i^{PC} , and 5 texture cluster vectors, V_i^T . Equation 3 was repeated for $T=20$ and for parameter m_t of equation 5 having initial, m_0 , and final, m_f , values of 3.0 and 1.2, respectively. After 20 updates, convergence error E_t , which is defined as

$$\sum_{i=1}^c \|V_{i,t} - V_{i,t-1}\| \quad (7)$$

was 1.32 and 9.14, respectively, for PC and texture data values ranging from 0 to 255. The resulting V_i are the cluster prototype vectors.

Next, NN test and training data were computed from the fuzzy membership, u_{ik} of X_k^{PC} (or X_k^T)

to their respective cluster vectors, where

$$u_{ik} = \left(\sum_{s=1}^c D_{ik} / D_{sk} \right)^{-2/(m-1)} \quad (8)$$

and D_{ik} was the Euclidean distance between X_k and V_i . Parameter m , which was 2, determines the weighting of distances between data and cluster vectors. For $m = 2$, the distances are weighted inversely.

Two fuzzy segmented images were produced. One for the 256- x 256-sample PC feature image and one for the 256- x 256-sample texture feature image. Segmented color images were made by assigning a color to each of the k image feature samples calculated as the product of the fuzzy membership value of the feature vector of the nearest cluster and the defined color of that cluster (Hall and others, 1992). That is, the red, green, and blue color values (R, G, B) of image feature sample k are

$$R_k = u'_{ik} R_i \quad (9)$$

$$G_k = u'_{ik} G_i \quad (10)$$

$$B_k = u'_{ik} B_i \quad (11)$$

where u'_{ik} is the maximum value of u_{ik} (equation 8) for $i=1, \dots, c$. Equal, or tie values of u_{ik} , were resolved randomly.

An alternative color assignment method wherein u_{ik} modulates only the color's intensity may be to transform (R_i, G_i, B_i) to intensity (I) hue (H) and saturation (S) and then transform $(u'_{ik} I, H, S)$ to (R_k, G_k, B_k) . This is subject to further study.

Figures 2 and 3 show the results of fuzzy segmentation applied to the texture and PC feature images, after they were subsampled from 512 x 512 to 256 x 256 samples. The sample distance is 20 m. Figure 2 shows the result of fuzzy clustering and segmentation using 10 cluster prototype vectors, $(V_i, i=1, \dots, 10)$ applied to the SPOT XS PC image. The color of each pixel sample was computed from the color of the nearest cluster prototype times the fuzzy membership to that cluster per equations (9, 10, 11).

Figure 3 is the result of fuzzy clustering (five clusters) and segmentation applied to the five-band texture feature image. Texture features (that is, orientated energy) were derived from the 10-m SPOT PAN image as per section II. As a spatial reference, a gray-level slice from the PAN image is superimposed on the segmented image.

These images were helpful in interpreting land cover classes and should prove valuable in manual compilation of land cover and land use regions.

IV. NEURAL NETWORK

A NN trained as a classifier can be considered a process that maps input data to desired output class. As shown in figure 1, classification of the fuzzy clustered data is by a multilayer NN. There are numerous reported applications of, and empirical comparisons between, multilayer NN classifiers and statistical classification techniques. Recently Ruck and others (1990) proved that every fully connected feedforward multilayer NN, when backpropagation

trained as a classifier, approximates the Bayes optimal discriminant function. They also proved that individual NN outputs, when trained to represent class membership, approximate the *a posteriori* probability function for the class being trained.

The advantage of this type of NN classifier is that it is distribution-free and thus requires no knowledge of the statistical distribution of the data (Benediktsson, and others, 1990). They have a further advantage over statistical classification methods when there is no knowledge of the distribution functions or when the data are non-Gaussian.

Another advantage is that probabilistic relaxation methods can be applied to the NN output data because they approximate the *a posteriori* probabilities. These post classification methods improve classification accuracy by incorporating spatial information that describes the influence of neighboring pixels on the probable classification of a central pixel (Richards, 1993; Gong and Howarth, 1989).

This study used a single hidden layer, fully connected, feedforward NN to classify preclustered data in the form of fuzzy membership values. For the output layer, the output of each individual processing node, or neuron, represents one of the N land cover classes; N was equal to 8. Because the NN input data values vary inversely with distance to the cluster vectors, this approach has some similarity to radial basis function networks (Rogers and Kabrisky, 1991).

Because network output layer neurons had normalized exponential output nonlinearities, described by Bridle (1990) as “softmax”, the neuron output, O_i , for class i , where $i = 1, \dots, N$ are likened to probabilities. That is

$$O_i > 1 \quad \forall i \quad \text{and} \quad \sum_{i=1}^N O_i = 1. \quad (12a,b)$$

It is unknown at this time whether these output values approximate class conditional probabilities because the NN inputs are fuzzy memberships and also because of the “softmax” output layer. The application of probabilistic relaxation methods to these output data is the subject of further study.

V. NETWORK TRAINING AND TEST SAMPLE COLLECTION

To train the NN to classify the data to a given set of land cover classes, the classes had to be defined, and then areas representative of each class, known as training samples, had to be collected. Data sources for determining what land cover classes were present in the study area included the U.S. Geological Survey (USGS) 1:24,000-scale topographic map, National Wetlands Inventory map, National Aerial Photography Program color-infrared photos from April 1989, SPOT (© 1988, SPOT Image Corporation, Reston, Virginia) XS and PAN images from July 7, 1988, and the fuzzy segmented texture and PC images. Also an unsupervised classification of the XS image was useful in choosing areas representative of each class.

Also related to this study, several merged or sharpened 10-m sample distance SPOT images were produced and their utility for interpreting land cover types evaluated. Basically following a technique described by Toet (1992) for multiresolution image fusion, coregistered PAN and XS data were merged. In this image, urban and residential areas were easily distinguished. The PAN-XS image was then merged with a USGS side-looking airborne radar image (SLAR). Compared to the PAN-XS image, forested and nonforested areas were easily

distinguished in this PAN-XS-SLAR image merge.

ERDAS (ERDAS, 1990) image classification software was used to delineate training samples. The XS image was displayed and training samples were defined by drawing polygons around areas representative of each class. Some iteration was required to develop classes that were fairly distinct within the test image. To test this, a spectral signature was developed for each class and then a maximum likelihood classifier was used on the samples. Some classes that could not be well distinguished were combined. The final eight land use and land cover classes chosen were river, pond, wetlands, forest, urban/residential, and three different crop types.

The final training sample polygon file based on these eight classes was then converted into a raster image so that pixel values within a class polygon were equal to the class value, and zero elsewhere. A number of software programs were developed to convert XS^{PC} and texture images and the training sample file into an ASCII format usable by the NN classifier. With the XS^{PC} , texture, and the training sample images as inputs, these programs create an ASCII file that has one line for each NN training (or test) sample. Each line contains the pixel/data value from each of the input files (three bands of SPOT XS^{PC} and five texture bands in this case), and the designated class value.

As shown in figure 1, the image data values were then replaced by their computed fuzzy membership values (equation 8). The ASCII file was then arbitrarily split to form two distinct NN data-class sample files; one to be used for training the NN, and the other for testing the resulting network. Rather than keeping half the data for training and half for testing, and to increase the number of training samples per class, 80 percent of the data-class samples were used for training. The number of training samples per class ranged from 600 to 9,000 and depended on the size of the land cover area. Although desirable for certain classes, it was not possible to collect more training samples per class.

The fully connected, feedforward, single hidden layer NN for these tests had 15 inputs. Ten inputs were the fuzzy membership values of XS^{PC} data to the 10 prototype cluster vectors previously established by FLVQ; 5 inputs were the fuzzy membership values of texture data to the 5 previously established texture prototype cluster vectors. The NN integrates this multisource data into the classification process.

In the output layer, a value of 1.0 for an individual neuron output, O_i , (equation 12a) represents the desired class, i . In training the network as a classifier, the interconnection weights are adjusted such that the desired class (that is, neuron output O_i) approximates the true class of the training data. The NN output layer used the “softmax” activation function of Bridle (1990) as described in NeuralWare (1993). The other layers had $TanH$ transfer functions. There were 15 hidden layer nodes and 8 output layer nodes. Each output is roughly the probability of land cover class i where $i = 1, \dots, N$. That is, for eight classes there were eight outputs; or 1 of N code where $N = 8$.

Backpropagation training (NeuralWare, 1993) was used with weight updates after each sample presentation. The learning rates for the hidden and output layers were 0.15 and 0.075; corresponding momentum values were 0.2, 0.2. After training for 600,000 sample presentations, the classification accuracy for both training and test files was measured. At this time, no classification sensitivity tests versus number of hidden nodes were made.

Tables 1 and 2 give classification results for the training and test files where class assignment per

sample presentation was from the NN output with maximum value. Table 3 gives class number versus land cover type.

Table 1: Classification rate, training data; 24,640 samples
Desired class

Actual class	Desired class							
	1	2	3	4	5	6	7	8
1	0.9790	0.0000	0.0000	0.0122	0.0222	0.0000	0.0103	0.0900
2	0.0000	0.8913	0.0145	0.0007	0.0000	0.0000	0.0000	0.0401
3	0.0000	0.0898	0.9837	0.0001	0.0000	0.0000	0.0000	0.0409
4	0.0149	0.0038	0.0000	0.9832	0.0032	0.0322	0.0009	0.0312
5	0.0002	0.0000	0.0000	0.0006	0.9731	0.0000	0.0000	0.0305
6	0.0000	0.0085	0.0000	0.0008	0.0000	0.9678	0.0000	0.0000
7	0.0012	0.0000	0.0000	0.0012	0.0016	0.0000	0.9599	0.0483
8	0.0046	0.0066	0.0017	0.0013	0.0000	0.0000	0.0289	0.7190

Table 2: Classification rate, test data; 6164 samples.
Desired class

Actual class	Desired class							
	1	2	3	4	5	6	7	8
1	0.9922	0.0000	0.0000	0.0005	0.0125	0.0000	0.0718	0.0861
2	0.0000	0.8447	0.0097	0.0065	0.0000	0.0000	0.0037	0.0000
3	0.0000	0.0000	0.9903	0.0037	0.0000	0.0000	0.0000	0.0000
4	0.0059	0.0795	0.0000	0.9714	0.0000	0.0054	0.0483	0.2196
5	0.0000	0.0000	0.0000	0.0060	0.9812	0.0000	0.1301	0.0000
6	0.0000	0.0114	0.0000	0.0120	0.0000	0.9946	0.0000	0.0059
7	0.0001	0.0000	0.0000	0.0000	0.0000	0.0000	0.4610	0.0178
8	0.0001	0.0644	0.0000	0.0000	0.0063	0.0000	0.2788	0.6706

Table 3: Class number and land cover type.

Class number	1	2	3	4	5	6	7	8
Land cover type	forest	ponds	river	urban	crop 1	crop 2	crop 3	wetland

The classification accuracy of the test file should be less than that of the training file since the former consists of samples not seen by the NN. This was not true for classes 1, 3, 5, or 6 and may be due to the differences between the test and training data. Class test samples were merely from the last 20 percent of the respective data-class files and thus correspond to the latter part of class polygons. An alternative approach to give more uniform sampling would be to split every five consecutive data-class samples into four training and one testing data samples. This is subject to further study.

The test file performance for samples not in the training set gives a measure of network generalization. Generalization, one of the most important properties of NN's, is the ability to induce a relationship that interpolates and extrapolates from the training examples in a sensible way (Hertz and others, 1991). Because too many weights in the NN can lead to poor generalization, additional classification tests with less hidden layer nodes (and less weights) are needed to determine if the relatively low test classification accuracy for classes 7 and 8 is due to poor network generalization or nonseparable classes.

A very rough comparison between maximum likelihood (ML) classification (of only XS^{PC} data) and NN classification (of table 2) is given in table 4. Although this data was similar to but not identical to the NN XS^{PC} class-data file, it allows some comparison of results.

Table 4: ML classification contingency table for XS^{PC} data; 31,369 samples

Class	1	2	3	4	5	6	7	8
1	0.947	0.000	0.000	0.001	0.000	0.000	0.000	0.003
2	0.000	0.690	0.000	0.000	0.000	0.000	0.000	0.000
3	0.000	0.016	0.989	0.001	0.000	0.000	0.000	0.017
4	0.009	0.234	0.000	0.877	0.000	0.091	0.020	0.013
5	0.000	0.000	0.000	0.000	0.960	0.000	0.000	0.004
6	0.000	0.000	0.000	0.032	0.000	0.909	0.000	0.000
7	0.002	0.000	0.000	0.012	0.010	0.000	0.933	0.093
8	0.042	0.060	0.011	0.076	0.030	0.000	0.047	0.871

Comparing urban class 4 of table 2 with table 4, NN classification accuracy for the fuzzy clustered XS^{PC} and texture data was 0.9714, whereas ML classification of XS^{PC} data was 0.877. In the fuzzy segmented texture image, texture appears an important feature in distinguishing urban and nonurban areas. Also, the NN classifier integrated texture and XS data into the classification process. This high classification accuracy may be due to the texture data, however, comparative tests without texture data are necessary to confirm this. It is noted that urban class training and test data were collected merely by drawing one or more polygons in urban image areas such as the city and subdivisions.

The benefit of classifying XS^{PC} instead of XS data is the subject of further study. Typically NN training data should be uncorrelated (Wasserman, 1993) therefore principle component data were used. For comparison, the differences in class accuracy results for maximum likelihood ML classification of the XS data-class file and XS^{PC} were between 0 and 2 percent.

Although these preliminary results are encouraging, further tests are planned including classifying the entire image using the trained network to spatially visualize classification results and also postclassification processing with probabilistic relaxation techniques (Gong and Howarth, 1989).

VI. CONCLUSION

A technique was described for doing land cover classification from multisource data using a NN trained as a classifier. Network inputs were the fuzzy membership values of derived orientated texture data and multispectral data. This technique allows for the supervised classification of fuzzy clustered data. It also gives continuous value, class probability-like outputs and the potential for postclassification processing with probabilistic relaxation techniques to improve classification accuracy.

Although this is an interim report on preliminary test results, the incorporation of texture with multispectral data appears to improve the classification of urban land cover; additional tests are required to confirm this.

Used for clustering, the FLVQ process requires few parameter adjustments. One benefit of fuzzy clustering is that it leads to fuzzy segmented images. Fuzzy segmented principal components images were useful in interpreting land cover categories from 10- and 20-m SPOT images. Their utility as auxiliary data for the interpretation and manual delineation of land cover regions will be evaluated. Texture based fuzzy segmentation and automated classification of urban areas from the SPOT PAN image or radar data, or both, appears promising and are the subjects of further study.

REFERENCES

- Benediktsson, J.A., Swain, P.H., and Ersoy, O.K., 1990, Neural network approaches versus statistical methods in classification of multisource remote sensing data: *IEEE Trans. Geoscience and Remote Sensing*, v. 28, no. 4, p. 540-552.
- Bezdek, J.C., 1993, A review of probabilistic, fuzzy and neural models for pattern recognition: *Journal of Intelligent and Fuzzy Systems*, v. 1, no. 1, p. 1-25.
- Bridle, J.S., 1990, Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition *in* Fogelman, F., and Herault, J., eds., *NATO ASI Series*, v. f68, Neurocomputing: Berlin, Springer-Verlag, p. 227-236.
- ERDAS, Inc., 1990, Image Processing Module, version 7.4, Atlanta, GA ERDAS, Inc., p. 1-218.
- Freeman, W.T., and Adelson, E.H., 1991, The design and use of steerable filters: *IEEE Trans. Pattern Intelligence and Machine Intelligence*, v. 13, no. 9, p. 891-906.
- Gong, P., and Howarth, P.J., 1989, Performance analyses of probabilistic relaxation methods for land-cover classification: *Remote Sens. Environ.*, v. 30, p. 33-42.
- Hall, L.O., Bensaid, A.M., Clark, L.P., Velthuizen, R.P., Silbiger, M.L., and Bezdek, J.C., 1992, A comparison of neural networks and fuzzy clustering techniques in segmenting magnetic resonance images of the brain: *IEEE Trans. Neural Networks*, v. 3, no. 5, p. 672-683.
- Hertz, J., Anders, K., Palmer, R.G., 1991, Introduction, chapt. 1, p. 1-10, and Performance of Multi-layer Feed-forward Networks, chapt. 6.4, p. 141-147 of *Introduction to the Theory of Neural Computation*, Hertz, J., Anders, K., Palmer, R.G.: Redwood City, CA, Addison-Wesley.
- Laws, K.L., 1980, Rapid texture identification *in* *Image Processing for Missile Guidance: Proc. SPIE*, 238, p. 376-380.
- NeuralWare, 1993, Neural computing, A Technology Handbook for Professional II/Plus and NeuralWorks Explorer. Pittsburgh, PA: NeuralWare, Inc.
- Richards, J.A., 1993, Context Classification, chapt. 8.8, p. 195-201 of *Remote Sensing Digital Image Analysis*, 2nd ed., J.A. Richards: Berlin, Springer-Verlag.
- Rogers, S.K., and Kabrisky, M., 1991, Hybrid Networks, chapt. 6.6 of *Biological and Artificial Neural Networks for Pattern Recognition*, Rogers, S.K., and Kabrisky, M.: Bellingham, WA, SPIE, The International Society for Optical Engineering, p. 72-74.
- Ruck, D.W., Rogers, S.K., Kabrisky, M., Oxley, M.E., and Suter, B.W., 1990, The multilayer perceptron as an approximation to a Bayes optimal discriminant function: *IEEE Trans. Neural Networks*, v. 1, no. 4, p. 296-298.
- SPOT Image Corporation, 1988: Reston, VA: SPOT Image Corp.
- Toet, A., 1992, Multiscale contrast enhancement with applications to image fusion: *Optical Engineering*, v. 31, no. 5, p. 1026-1031.

Wasserman, P.D., 1993, Neural Engineering, chapt. 11 *of* Advanced methods in neural computing: New York, VanNostrand-Reinhold, p. 214-244.
