COMPUTATIONAL INTELLIGENCE FOR CLASSIFICATION OF REMOTELY SENSED IMAGES

Peter Sinčák, *Howard Veregin*, *Norbert Kopčo*

Abstract: This study focuses on a comparison of three non-traditional classification methods – supervised classification based on fuzzy sets, fuzzy ARTMAP classification and modular neural network classification – using a multispectral video mosaic with a pixel size of one meter. Fuzzy ARTMAP classification and supervised classification based on fuzzy sets yield accuracy levels that are consistent with or better than traditional methods, while modular neural network classification performs quite poorly. Results show significant variations in accuracy on a per-class basis.

Key words: *Image classification, fuzzy sets, ARTMAP, modular networks*

Received: July 3, 1998
Revised and accepted: September 21, 1998

1. Introduction

1.1 Objectives

Our interest in high-resolution urban imagery stems from our ultimate goal, which is to use classification (based on Computational Intelligence) to derive inputs to urban stormwater runoff models [30]. However this type of imagery poses special problems for classification, since it contains high within-class variation as a result of the small pixel size coupled with the spatial complexity of the urban scene. Our attempts to use traditional classification methods based on statistical methods have not been highly successful due to the difficulty of discriminating among classes in feature space. We have been able to obtain a maximum classification accuracy of

*Peter Sinčák
Computational Intelligence Group, Laboratory of AI, Department of Cybernetics and Artificial Intelligence, Technical University of Košice, Košice, Slovak Republic
†Howard Veregin
Department of Geography, University of Minnesota, Minneapolis, Minnesota, USA
‡Norbert Kopčo
Computational Intelligence Group, Laboratory of AI, Department of Cybernetics and Artificial Intelligence, Technical University of Košice, Košice, Slovak Republic, (Currently at CNS, University of Boston, USA)

only 72 percent. The non-traditional classification methods we evaluate here are less dependent on statistical assumptions and are, at least in theory, better able to model highly non-linear discrimination functions separating classes in feature space. The methods we have adapted and implemented in this study are modular neural network classification, fuzzy ARTMAP classification, and supervised classification based on fuzzy sets. The modular neural network method has been shown to be capable of approximating complex discriminant functions separating classes in feature space [12]. We hypothesize that it may therefore be useful for our imagery. The fuzzy ARTMAP classifier is a supervised classification technique based on Adaptive Resonance Theory and fuzzy sets. It is a "model-free" approach and should yield good results regardless of the statistical characteristics of the input data. Our use of supervised classification based on fuzzy sets (which is distinct from the more common "fuzzy classification") is more experimental, as this approach has not, to our knowledge, been used for image classification purposes. Our goal is to assess its potential in this domain. Our implementation of this method is adapted from speech recognition applications [22].

For our imagery, the supervised fuzzy and fuzzy ARTMAP methods perform about as well as traditional methods in terms of overall accuracy levels. The modular neural network method performs quite poorly. Results also show that the three methods exhibit significant differences in accuracy on a per-class basis. Even when overall accuracy is relatively low, some classes may be classified accurately.

1.2 Data

The experiments reported in this paper were performed on high-resolution multi-spectral video imagery. Video imagery is a relatively new data source for remote sensing, and shows promise for a wide range of applications [16]. For our purposes (to derive inputs to urban stormwater runoff models), the main advantage of video imagery is that it provides sufficient resolution for a detailed analysis of urban land cover classes. No existing commercial satellite system provides multi-spectral imagery with a comparable pixel size. Indeed this limitation has severely impacted the utility of commercial imagery for urban remote sensing applications. The classes of interests are in Tab. I.

class	defined as
A'	shingle roofs
B'	shadow
C'	water
D'	deciduous trees
E'	grass
F'	asphalt
G'	concrete
H'	grass/stubble

Tab. I *Classes of interests on the image.*

Our study area is a portion of the Mud Run Creek urban watershed in southwest Akron, Ohio. Video imagery was post-processed by digitizing the red, green and blue bands from selected video frames, yielding a vector of RGB values in the range 0-255 for each pixel. Adjacent frames within subcatchment 24 were mosaicked digitally using ground control points derived from large-scale engineering maps. At this stage, the imagery was also resampled to produce exact $1-m$ resolution. The digital mosaic was then cropped to conform to the dimensions of subcatchment. The mosaic contains 420 rows and 1100 columns of pixels, giving 462,000 pixels in each band.

1.3 Classification methods

There are numbers of traditional techniques for supervised and unsupervised classification. For supervised classification the best-known technique is a Bayesian classification based on the a priori assumption of a Gaussian distribution of training data [28]. For unsupervised classification the main methods include ISODATA and other types of clustering [19], [20]). Significant progress in classification has also been made using fuzzy logic and neural networks. An interesting comparison of these techniques can be found in [3] and [14]. Neural-based classification of remote sensing images has been reported in [13], [17], [8] and [4]. Applications of fuzzy systems in pattern recognition can be found in [31] and [22]. A large number of experiments have also been done with neuro-fuzzy approaches to classification problems. An excellent review can be found in [18]. The problem of the influence of training set characteristics on neural classification can be found in [10]. An interesting methodological paper about the future of classification approaches in remote sensing can be found in [32].

We have adapted and implemented three non-traditional supervised classification approaches in this study – classification based on fuzzy sets, fuzzy ARTMAP classification and modular neural network classification. We hypothesize that all three of these approaches will yield improved accuracy levels relative to traditional approaches, due to the less restrictive nature of the assumptions and the proven ability of these methods to model complex non-linear discriminant functions. The following discussion presents the theoretical background for each method, its implementation details, and the results obtained for our imagery.

2. Classification Based on ARTMAP Neural Networks

This section of the paper describes a classification procedure based on ARTMAP neural networks. ARTMAP is a recurrent neural network and is generally able to solve complex classification problems. This method is "model-free" and we hypothesize that it will produce good results regardless of the statistical characteristics of the input data.

2.1 Theory and background

There are many approaches to classification in the neural networks domain. ART networks belong to the class of recurrent neural network topologies with unsupervised or supervised training approaches. ART neural networks are based on

Adaptive Resonance Theory developed by [6]. The main goal of Adaptive Resonance Theory is to provide a solution to the "stability-plasticity dilemma". The network has to be plastic (ready to adapt if new, unknown inputs are presented) but also be stable in case it is presented with inputs that are already known to the network (to prevent permanent recoding). The ART unsupervised training neural network family encompasses various types of networks. ART1 represents unsupervised classification of binary inputs while ART2 is a non-binary extended version. For supervised classification the fundamental model is ARTMAP. Fuzzy ARTMAP is an extension of ARTMAP that incorporates fuzzy set operations.

Because ART1 is the fundamental structure in the Adaptive Resonance Theory, we first describe the ART1 algorithm. We then describe the ARTMAP, fuzzy ART and fuzzy ARTMAP networks.

2.1.1 ARTMAP neural networks

The ARTMAP neural network according to [6] was designed as a modification of the ART1 neural network for supervised classification. ARTMAP uses the idea of matching two unsupervised classification procedures with the aim of providing one supervised result. The two unsupervised networks are used to produce results which are then processed in a subnetwork connecting these two networks. In its basic version, an ARTMAP network consists of two ART1 subnets denoted ART_a and ART_b. The input pattern is presented to the input of the subnet ART_a (layer F_0^a) and the corresponding category is presented to the input of subnet ART_b (layer F_0^b). Both subnets then generate internal codes. These codes are compared in another subnet called the *Mapfield*. If they do not match, an *Inter* − *ART* − *Reset* is activated which forces the ART_a subnet to find a correct category.

If the category codes which define the category of the presented patterns satisfy certain conditions, the ART_b subnet can be omitted and the category code can be fed directly to the *Mapfield*. In the present research an ARTMAP network in Fig. 1.

The ARTMAP neural network training procedure can be summarized as follows:

1. Pattern $\bar{x}$ is presented to layer F_0^a.

2. The code of the category associated with $\bar{x}$ is presented to layer F_0^b.

3. The subnetwork ART_a determines a winner in layer F_2^a.

4. A match test between the winners from ART_a and layer F_0^b is done in the *Mapfield* subnetwork F^{ab}.

5. If there is a mismatch between the ART_a winner and the F_0^b category code, the "Match-tracking" procedure is initiated. This procedure is similar to the rules for activation of the *Reset* neuron. The current winning neuron in F_2^a is reset (deactivated) and a search for another winner is initiated. This implies a return to step 3.

6. If there is a match in the *Mapfield*, then adaptation of weights $F_1^a \rightarrow F_2^a$ and $F_2^a \rightarrow F^{ab}$ is done.

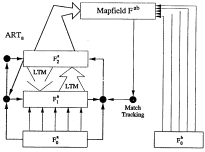

Fig. 1 *General topology of ARTMAP neural network.*

7. The algorithm returns to step 1 until stable coding of all input patterns $\bar{x}$ is achieved.

A more precise description of the *Mapfield* layer and more information about practical implementation issues can be found in [6] and [2].

2.1.2 Fuzzy ARTMAP neural networks

A fuzzy version of ARTMAP network was developed with the aim of improving the performance of the ARTMAP algorithm. To achieve this goal, standard set operators defined by certain equations are replaced by fuzzy set operations. For example, the "fuzzy AND" operation for two vectors $\bar{x}$, $\bar{y}$ is $(\bar{x} \wedge \bar{y}) = \min(x_i, y_i)$ The following parts of the ART1 algorithm are then modified.

1. In ART1 the winner is determined using a usual equation [6], which can be rewritten using the vector norm operator. The fuzzy version of this equation is

$$T_j = \frac{|\bar{I} \wedge \bar{w}_j|}{\alpha + |\bar{w}_j|}. \tag{1}$$

2. A similar modification is done for the rule governing the *Reset* neuron activation , i.e.,

$$\frac{|\bar{I} \wedge \bar{w}_j|}{\alpha + |\bar{I}\,|} \geq \rho. \tag{2}$$

3. Changes are also made in weights adaptation. In the fuzzy ARTMAP algorithm the adaptation of weights is done as follows.

$$\bar{w}_j^{new} = \beta(\bar{I} \wedge \bar{w}_j^{old}) + (1-\beta)\bar{w}_j^{old} \tag{3}$$

where β is a "learning rate" parameter. If $\beta = 1$, this adaptation is referred to as "fast learning".

The fuzzy ARTMAP neural network is then created from the ARTMAP network by replacing the ART1 subnets by fuzzy ART subnets.

2.2 Implementation

The fuzzy ARTMAP neural network was implemented by the authors into the Stuttgart Neural Networks Simulator (Zell et al., 1994) and all the experiments were carried out using this implementation of the algorithm. The fuzzy ARTMAP algorithm requires that the pattern data satisfy the following conditions.

1. The values of the inputs must be in the range $[0, 1]$.
2. No pattern may be a subset of any other pattern in the pattern set.

To satisfy these two conditions the input pattern set was reformatted so that every input was described by a vector $[r, g, b, 1-r, 1-g, 1-b]$ where r, g, and b are pixel RGB values scaled to a range of $[0, 1]$. This normalizes the patterns such that the length of each pattern (as defined by the sum of its components) is always 3. This type of normalization is called "complement coding."

Categories were encoded using the vector $[c_1, c_2, \ldots, c_8]$ where

$$c_i = \begin{cases} 1 & \text{if } i \text{ is the category to which given pattern belongs,} \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Multiple experiments were carried out for different random orderings of the patterns. A voting strategy for improving classification was also tested. This strategy is generally used to minimize the dependency of classification on the ordering of the presented patterns (this dependency is a known deficiency of the fuzzy ARTMAP algorithm). The improvement achieved by this method was only marginal, the reason being that dependency on ordering is minimized if the patterns are presented in random order.

2.3 Results

Results for the training data give an overall accuracy (PCC) of 99.5 percent and a kappa statistic of 0.993. In terms of per-class accuracy statistics, all classes are at or nearly at 100 percent. It has been shown that for a pattern set without contradictory patterns, the fuzzy ARTMAP algorithm achieves perfect classification accuracy [6]. The small errors observed in the training set results suggest that the training set may contain a small number of contradictory patterns, i.e., pairs of patterns with the same RGB values assigned to different classes. Results for the test data are more imperfect; see Tab. II.

	Classified Image A–H Test Sites A'–H'								Σ
	A'	**B'**	**C'**	**D'**	**E'**	**F'**	**G'**	**H'**	
A	155	3	0	6	2	0	27	0	193
B	2	83	0	16	0	0	0	0	101
C	1	0	1227	0	0	0	0	0	1228
D	7	8	0	385	0	0	0	0	400
E	13	3	0	4	777	9	4	29	837
F	1	0	0	0	7	634	3	0	645
G	24	0	0	0	0	5	529	0	558
H	0	0	0	0	38	0	1	774	813
Σ	201	97	1227	411	824	648	564	803	4775
PCC=0.95581 Kappa=0.94699									

Tab. II *Contingency table for fuzzy ARTMAP classification.*

Overall accuracy levels and per-class accuracy statistics for the fuzzy ARTMAP approach are better than those attained using traditional classifiers. This is particularly true for shingle roofs and deciduous trees, although concrete, asphalt and shadow show similar tendencies. This classified image is in Fig. 4.

3. Supervised Classification Based on Fuzzy Sets

Classification can be described as a mapping from an input feature space to an output class space. Examples of classification procedures with crisp inputs and fuzzy outputs include the fuzzy BP classifier [18] and the fuzzy ISODATA clustering procedure. Both approaches yield a vector of membership function values for each input relative to each cluster in feature space. The basic principles of fuzzy classification are presented in [21]. In this section we describe a somewhat different approach, called supervised classification based on fuzzy logic, in which both inputs and outputs are crisp.

3.1 Theory and background

Supervised classification based on fuzzy sets has been used for speech recognition [22] and we have adapted it here for image processing purposes. Our adaptations focus on the similarity vector calculation and the use of a hashing function which is not found in the speech recognition application. The utility of the approach for image processing is not known, since it has to our knowledge not been applied in this domain before. The approach differs from other methods in the classification step, where an unknown input pattern is compared against each member of the training set using a membership function and similarity vector calculation. We suspect that this comparison-based approach may prove valuable when dealing with non-compact clusters in feature space. The classification task can be broken down into a learning step and a classification step. The learning step is as follows. First, assume that we have observations in n-dimensional feature space that are to be discriminated into m classes,

$$(c_1, c_2, \dots, c_j, \dots, c_m)$$

Initially a pattern can be described as

$$\mathbf{x} = \{x_1, x_2, \dots, x_i, \dots, x_n\} \tag{5}$$

where x_i is a feature from the n-dimensional feature space. Generally we can consider the following training set structure

$$\underbrace{\mathbf{x}^1, \dots, \mathbf{x}^{k_1}}_{c_1}, \underbrace{\mathbf{x}^1, \dots, \mathbf{x}^{k_2}}_{c_2}, \underbrace{\mathbf{x}^1, \dots, \mathbf{x}^{k_3}}_{c_3}, \dots \underbrace{\mathbf{x}^1, \dots, \mathbf{x}^{k_m}}_{c_m} \tag{6}$$

This means that we assume that the training set consists of m classes of interest, with each class j having k_j patterns, and each pattern having the structure shown in equation (5). For each pattern in the training set we can calculate the value $\mu(l, j, i)$, which represents a membership function of the l^{th} pattern for class j for feature i. Thus the definition intervals are $l \in \langle 1, k_j \rangle$, $j \in \langle 1, m \rangle$ and $i \in \langle 1, n \rangle$. The computation is

$$\mu(l, i, j) = \left\{ 1 + \left| \frac{\hat{x}_i(j) - x_i^l(j)}{E} \right|^F \right\}^{\alpha} \tag{7}$$

where $\hat{x}_i(j)$ is the mean value of the i^{th} feature of the overall training vectors in class j, and $x_i^l(j)$ is a particular value of feature i for the l^{th} pattern, which belongs to class j. This equation represents a typical membership function. Coefficients E, F and α are important for membership function determination. This means that we are constructing a new feature space; pattern $\mathbf{x}$ from class j in equation (5) is now transformed into the new feature space as follows,

$$\mathbf{x} = \{\mu(l, 1, j), \mu(l, 2, j) \dots, \mu(l, n, j)\} \tag{8}$$

where l is pattern $\mathbf{x}^l$ in class j. Each class j of m now has its own k_j patterns $\mathbf{x}$ described by its membership function vector according to equation (8). The classification step is based on a discrimination rule procedure, which calculates the so-called "similarity vector" between an unknown vector $\mathbf{y}$ and class j in n-dimensional feature space,

$$\mathbf{s}(y, l_j) = \{s(y, l_j, 1), s(y, l_j, 2), \dots, s(y, l_j, n)\} \tag{9}$$

In this equation, $s(y, l_j, i)$ is a value which represents the similarity between an unknown vector $\mathbf{y}$ and the l^{th} pattern in the j^{th} class, for the i^{th} feature. The calculation of such an element of the similarity vector can be done as follows,

$$s(y, l_j, i) = \left\{ 1 + Q_i \left| 1 - \frac{\mu(y, j, i)}{\mu(l, j, i)} \right| \right\}^{-2z} \tag{10}$$

where $\mu(l, j, i)$ is from (7), Q_i is a parameter and $\mu(y, j, i)$ is a particular calculation for an unknown vector $\mathbf{y}$ in the form

$$\mu(y, j, i) = \left\{ 1 + \left| \frac{\hat{x}_i(j) - y_i}{E} \right|^F \right\}^{\alpha} \tag{11}$$

This represents the membership function of $\mathbf{y}$ for class j considering the mean values of feature i for class j. For equation (9) the module of this vector is calculated in the usual way, i.e.,

$$|\mathbf{s}(y, l_j)| = \sqrt{\sum_{i=1}^{n} (s(y, l_j, i))^2} \tag{12}$$

According to the above considerations we obtain

$$\begin{array}{ll} k_1 \text{ values of } |\mathbf{s}(y, l_1)| & l_1 \in \langle 1, k_1 \rangle \\ k_2 \text{ values of } |\mathbf{s}(y, l_2)| & l_2 \in \langle 1, k_2 \rangle \\ \dots & \\ \dots & \\ k_m \text{ values of } |\mathbf{s}(y, l_m)| & l_3 \in \langle 1, k_m \rangle \end{array}$$

Thus if we have

$$\mathbf{K} = \sum_{j=1}^{m} k_j$$

then we have $\mathbf{K}$ different modules of vectors for a single unknown vector $\mathbf{y}$.

Classification is now fairly straightforward, such that we look for $\max\{|\mathbf{s}(y, l)|\}$. If we find a maximum we can decide that if

$$\max\{|\mathbf{s}(y, l)|\} \geq T \tag{13}$$

where T is a selected threshold, then the unknown vector $\mathbf{y}$ belongs to the class to which vector l belongs. If equation (13) is not true then the unknown vector does not belong to any of the classes of interest.

3.2 Implementation

The classification method described above is computationally intensive. Therefore a hashing function was used to reduce the number of computations. The training set was augmented with the sum of the pixel reflectance values, $S = R + G + B$. Next, in equation (11), the vector similarities $\mathbf{s}(y, l_j)$ were performed only for those cases in which $\mathbf{y}_S^l$ was in the range $(\mathbf{x}_S^l - TR, \mathbf{x}_S^l + TR)$ where TR is a selected threshold. We computed

$$\mathbf{x}_S^l = \mathbf{x}_R^l + \mathbf{x}_G^l + \mathbf{x}_B^l \tag{14}$$

and

$$\mathbf{y}_S^l = \mathbf{y}_R^l + \mathbf{y}_G^l + \mathbf{y}_B^l \tag{15}$$

The hashing approach increased the speed of processing by a factor of 10.

The complete algorithm was implemented as an independent C program in a Linux environment. Experiments showed that the best parameter values were as follows: $E = 5$, $F = 3$ and W_i, $z = 1$. Parameter T was set to 1.3 from the definition range $\langle 0, \sqrt{3} \rangle$ and TR was set to 10. The values of these parameters are data-dependent and it is difficult to set universal values for general use. The final processing time was about 5 minutes on a Pentium system running Linux.

3.3 Results

Results for the 4880 pixels in the training set give an overall classification accuracy (PCC) of 98.5 percent and a kappa statistic of 0.979. Results for the test data are more imperfect; see Tab. III. An nalysis of the contingency table reveals frequent confusion among concrete, asphalt and shingle roofs. This is perhaps not surprising, since shingles often have an asphalt base and since concrete and asphalt have a similar appearance, especially after extensive road wear has taken place. There is also evidence of confusion between water, deciduous trees and shadow. Again, this result is not surprising since water and shadow both have low reflectance values, and since areas of deciduous tree cover contain a mixture of tree crowns interspersed with shadows. The classes with the highest accuracies are water, grass and grass-stubble.

Overall accuracy levels are marginally inferior to our best results using traditional classifiers. However, comparison of per-class accuracy statistics shows that the fuzzy classifier produces superior results for several classes, especially shingle roofs and grass. Other classes, such as water (which is spectrally quite uniform), does not show an improvement in accuracy for the fuzzy method. Results are somewhat mixed, but the fuzzy method would appear to have some utility in image classification. The classified image is shown in Fig. 4.

	Classified Image A–H Test sites A'–H'								$\sum$
	A'	B'	C'	D'	E'	F'	G'	H'	
A	146	7	0	6	6	5	24	0	194
B	3	77	0	18	0	0	0	0	98
C	0	0	1227	1	0	0	0	0	1228
D	7	10	0	376	1	0	0	0	394
E	5	3	0	9	749	3	2	36	807
F	2	0	0	0	16	634	8	1	661
G	36	0	0	1	1	3	524	1	566
H	2	0	0	0	51	3	6	765	827
$\sum$	201	97	1227	411	824	648	564	803	4773
PCC=0.94199 Kappa=0.93041									

Tab. III *Contingency table for supervised classification using fuzzy sets.*

4. Modular Neural Network Classification

This section of the paper describes the adaptation and implementation of a modular neural network approach for image classification. A good review of modular neural networks can be found in [15]. Research suggests that such an approach can be effective when there are convergence problems during the training procedure [12]. The modular neural network approach has been shown to be capable of approximating complex discriminant functions, as might be observed in feature space for high-resolution urban imagery. The basic topology of modular neural networks is shown in Fig. 2.

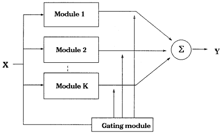

Fig. 2 *Modular neural network.*

A modular neural network consists of "expert neural networks" and "gating neural networks" whose role is to manage the expert networks. There are a number of training procedures used on this topology; we tested a training approach described in [12] and developed for classification tasks.

4.1 Theory and background

Given an input pattern $\mathbf{x}$, we define $\mathbf{K}$ expert neural modules such that the vector output from the i^{th} module is

$$\mathbf{y_i} = f(\mathbf{x}^\mathbf{T}, \mathbf{w_i}) \tag{16}$$

where $\mathbf{w_i}$ is the vector of weights in module i. For a gating neural network the activation of layers is given by

$$\mathbf{a} = \mathbf{x}^\mathbf{T}\mathbf{v} \tag{17}$$

where **a** is a vector of activation values and **v** is a weight matrix. We define the so-called "softmax function" as

$$g = \frac{\exp(a)}{\sum_{=} \exp(a)} \tag{18}$$

The overall output from the modular neural network is then computed as

$$\mathbf{y} = \sum_{i=1}^{K} g_i \mathbf{y_i} \tag{19}$$

From a statistical point of view we can state that in classification tasks

- inputs to the neural network have in general a Gaussian distribution,
- outputs of the neural network have a Bernoulli distribution as follows,

$$p(\mathbf{d}|\mathbf{x}) = \prod_{i=1}^{q} y_i^{d_i}(1-y_i)^{1-d_i} \tag{20}$$

where

d_i is the desired value of the i^{th} output neuron of the expert module,
y_i is the calculated value of the i^{th} output neuron of the expert module,
q is the number of neurons in the expert module.

Thus the general output from a modular neural network is

$$l_K(\theta) = \ln \sum_{i=1}^{K} g_i \prod_{j=1}^{q} y_{ij}^{d_{ij}}(1-y_{ij})^{1-d_{ij}} \tag{21}$$

where K is the number of expert modules and θ is a set of parameters that influence the modular neural network procedure. Neurons in both the expert and gating networks have a sigmoidal activation function,

$$y_{ij} = \frac{1}{1+\exp(-\mathbf{x}^T\mathbf{w}_{ij})} \tag{22}$$

where y_{ij} is the output of the j^{th} neuron of the i^{th} module. Generally, a modular neural network consists of the following steps.

1. Initialization of weights within the range $\langle -1, 1 \rangle$.
2. Adaptation of weights.
   - The outputs from the gating networks are

$$a_i(t) = \frac{1}{1+\exp(-\mathbf{x}^T\mathbf{v}_i(t))} \tag{23}$$

$$g_i(t) = \frac{a_i(t)}{\sum_{j=1}^{K} a_j(t)} \tag{24}$$

- The output of each expert network is

$$y_{ij}(t) = \frac{1}{1+\exp(-\mathbf{x}^T\mathbf{w}_{ij}(t))} \tag{25}$$

- We calculate the following parameter as

$$h_i(t) = \frac{g_i(t) \prod_{k=1}^{q} y_{ik}^{d_{ik}}(1-y_{ik})^{1-d_{ik}}(t)}{\sum_{j=1}^{K} g_j(t) \prod_{k=1}^{q} y_{jk}^{d_{jk}}(1-y_{jk})^{1-d_{jk}}(t)} \tag{26}$$

- Adaptation of weights in the expert network is

$$\mathbf{w}_{ij}(t+1) = \mathbf{w}_{ij}(t) + \gamma h_i(t)(d_j(t) - y_{ij}(n)) \frac{\exp(-\mathbf{x}^T\mathbf{w}_{ij})}{(1+\exp(-\mathbf{x}^T\mathbf{w}_{ij}))^2}\mathbf{x} \tag{27}$$

- Adaptation of weights in the gating network is

$$\mathbf{v}_i(t+1) = \mathbf{v}_i(t) + \gamma(\mathbf{h}_i(n - g_i(t)) \frac{\exp(-\mathbf{x}^T\mathbf{v}_i)}{(1+\exp(-\mathbf{x}^T\mathbf{v}_i))^2}\mathbf{x} \tag{28}$$

where γ is the learning rate.

The last two equations represent the training strategy of modular neural networks for weights adaptation.

4.2 Implementation

The modular neural network approach described in the previous section was implemented in C in a Linux environment. A number of different topologies were tested. The final topology has 12 expert networks and one gating network. Learning parameters ranged between $\langle 0.1, 0.8 \rangle$ and training was terminated after 8000 cycles.

4.3 Results

Results for the modular neural network classifier for the training data are quite poor. The PCC is 55.14 percent and the kappa statistic is 0.47. With the exception of water, which is the most spectrally uniform class, all classes have low per-class accuracy statistics. It is believed that the main reason that the modular neural network performs so poorly is due to the gating network. It seems that the adaptation approach of the gating network works well for simple functions, but fails for very complex classification problems. Test data show essentially the same patterns. The contingency Tab. IV shows confusion between almost every class, and the classified image (see Fig. 4) exhibits high-frequency variation not present in the other classified images.

Thus a pixel selected at random from the water class on the modular neural network image has a higher probability of actually being water than a pixel selected from the water class on any other image. Again, these results indicate that overall accuracy statistics can mask important variations in per-class accuracy statistics, and that even an image with a low overall accuracy level can have high accuracy for some classes. A graph presenting per-class accuracies for classifiers is shown in Fig. 3.

	Classified Image A–H Test Sites A'–H'								Σ
	A'	**B'**	**C'**	**D'**	**E'**	**F'**	**G'**	**H'**	
A	44	9	1	9	72	64	193	23	415
B	70	28	4	23	100	145	114	70	554
C	0	2	1206	0	0	0	0	2	1210
D	14	26	2	295	33	12	34	1	417
E	28	13	2	41	403	245	58	145	935
F	4	3	0	4	95	93	11	30	240
G	12	16	9	30	5	24	37	5	138
H	29	0	3	9	116	65	117	527	866
Σ	201	97	1227	411	824	648	564	803	4775
PCC=0.55141 Kappa=0.47027									

Tab. IV *Contingency table for modular neural network classification.*

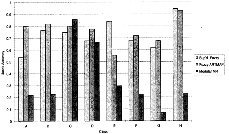

Fig. 3 *Classification results for per-class accuracy.*

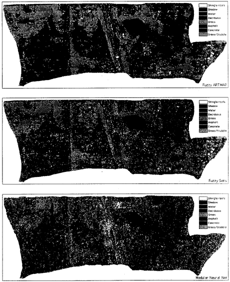

Fig. 4 *General topology of ARTMAP neural network.*

5. Conclusion

The results of this study show that both supervised classification based on fuzzy sets and fuzzy ARTMAP classification can be used effectively for classification problems in remote sensing. In contrast, poor results were obtained using modular neural networks. We suspect that the latter approach is not suited to remotely sensed data where classes are often not clearly defined in feature space. However, the degree to which these results are generalizable is not known, since our experiments are based on only one set of data. An additional unresolved problem is the effect of training data characteristics, and in particular the unequal number of patterns (pixels) for different classes. In general we observed better results for classes with more patterns, but this might be due to the fact that the larger classes tended to be those with the lowest spectral variability (e.g., water). While consistency in class size is desirable, a uniform number of patterns for each class does not necessarily guarantee that all classes will be classified with equal accuracy, since different classes exhibit different levels of internal homogeneity. Adapted supervised classification based on fuzzy sets works well and some further improvement could be done considering more complex membership functions.

Acknowledgments

This publication is based on work sponsored by the U.S.-Slovak Science and Technology Joint Fund in Cooperation with the Ministry of Ecology of the Slovak Republic and the US Environmental Protection Agency under Project Number 94077.

References

[1] Baldwin J.F, Martin T.P., Pilsworth B.W.: Fril – fuzzy and evidential reasoning in artificial intelligence. John Wiley and Sons, New York NY, 1995.

[2] Bartafai G., Tauritz D.: ART FAQ, 1996, Internet URL: http://www.wi.leidenuniv.nl/art

[3] Benediktsson J., Swain P., Ersoy O.: Neural network approaches versus statistical methods in classification of multisource remote sensing data. IEEE Transactions on Geoscience and Remote Sensing, **28**, 4, 1990, 540-552.

[4] Carpenter G.A., Gjaja M., Gopal S., Woodcock C.: ART neural networks for remote sensing: vegetation classification from Landsat TM and Terrain Data. IEEE Transactions on Geoscience and Remote Sensing, **35**, 2, 1997, 308-325.

[5] Carpenter G.A., Grossberg S.: A massively parallel architecture for a self-organizing neural pattern recognition machine. Computer Vision, Graphics, and Image Processing, **37**, 1987, 54-115.

[6] Carpenter G.A., Grossberg S., Markuzon N., Reynolds J., Rosen D.B.: Fuzzy ARTMAP: a neural network architecture for incremental supervised learning of analog multidimensional maps. IEEE Transactions on Neural Networks, **3**, 1992, 698-713.

[7] Carpenter G.A., Grossberg S., Reynolds J.H.: ARTMAP: supervised real-time learning and classification of nonstationary data by a self-organizing neural network. Technical Report CAS/CNS-TR-91-001, Boston, MA: Boston University, 1991.

[8] Civco D.L., Wang Y.: Classification of multispectral, multitemporal, multisource spatial data using artificial neural networks. ACSM/ASPRS Annual Convention and Exposition, Baltimore, 1994, 1, 123-133.

[9] Congalton R.G.: A review of assessing the accuracy of classifications of remotely sensed data. Remote Sensing of Environment, 1991, **37**, 35-46.
Fitzpatrick-Lins K.: Comparison of sampling procedures and data analysis for a land-use and land-cover map. Photogrammetric Engineering and Remote Sensing, 1981, **47**, 343-351.

[10] Foody G.M., McCulloch M.B., Yates W.B.: Classification of remotely sensed data by an artificial neural network: issues related to training data characteristics. Photogrammetric Engineering and Remote Sensing, 1995, **61**, 391-401.

[11] Goodchild M.F.: Conflation: combining GIS sources. 1996, Internet URL: http://www.ncgia.ucsb.edu/research/ucgis/proposals/conflation.html

[12] Haykin S.: Neural networks: a comprehensive foundation. Macmillan, New York NY, 1994.

[13] Heerman P.D., Khazenie N.: Classification of multispectral remote sensing data using a BP neural network. IEEE Transactions on Geoscience and Remote Sensing, 1992, **30**, 1, 81-88.

[14] Hepner G., Logan T., Rittne N., Bryant N.: Artificial neural network classification using a minimum training set: comparison to conventional supervised classification. Photogrammetric Engineering and Remote Sensing, 1990, **56**, 469-473.

[15] Hrycej T.: Modular learning in neural networks. John Wiley and Sons, New York NY, 1992.

[16] King D.J.: Airborne multispectral digital camera and video sensors: a critical review of system designs and applications. Canadian Journal of Remote Sensing, 1995, **21**, 3, 245-273.

[17] Kulkarni A.: Artificial neural networks for image understanding. Van Nostrand Reinhold, New York NY, 1994.

[18] Lin C., Lee C.: Neural fuzzy systems. Prentice Hall, Englewood Cliffs NJ, 1996.

[19] Ocelikova E.: Clustering analysis as a classification tool. International Scientific Conference on Micro-Cad Systems 93, Košice, Slovak Republic, 1993.

[20] Ocelikova E.: Significance of cluster analysis. Proceedings of 1st IFAC Workshop on New Trends in Design of Control Systems, 1994, 420-423.

[21] Pal S.K., Majumder D.: Fuzzy mathematical approach to pattern recognition. John Wiley and Sons, New York, NY, 1986.

[22] Pao T.: Adaptive pattern recognition and neural networks. Addison-Wesley, Reading MA, 1989.

[23] Stehman S.: Estimating the kappa coefficient and its variance under stratified random sampling. Photogrammetric Engineering and Remote Sensing, 1996, **62**, 401-408.

[24] Stoms D.: Reasoning with uncertainty in intelligent geographic information systems. GIS'87, 1987, 693-700.

[25] Story M., Congalton R.: Accuracy assessment: a user's perspective. Photogrammetric Engineering and Remote Sensing, 1986, **52**, 397-399.

[26] Swain P.H., Davis S.M.: Remote sensing: the quantitative approach. McGraw-Hill, New York, NY, 1978.

[27] Tikunov V.S.: Some issues in the modeling approach to cartography. Unpublished manuscript, Department of Geography, University of California, Santa Barbara, 1986.

[28] Tou J., Gonzales R.: Pattern recognition principles. Addison-Wesley, Reading MA, 1974.

[29] Veregin H.: A taxonomy of error on spatial databases. Technical paper 89-12, National Center for Geographic Information and Analysis, University of California, Santa Barbara, 1989.

[30] Veregin H., Sinčák P., Gregory K., Davis L.: Integration of high resolution video imagery and Urban stormwater runoff modeling. 15th Biennial Workshop on Videography and Color Photography in Resource Assessment, 1995, 182-191.

[31] Wang F.: Fuzzy supervised classification of remote sensing images. IEEE Transactions on Geoscience and Remote Sensing 1990, **28**, 194-201.

[32] Wilkinson G.G.: Classification algorithm - where next? Proceedings of the International Workshop on SoftComputing in Remote Sensing Data Analysis, Milan, Italy, 1995, 93-99.

[33] Zell A. et al.: Stuttgart neural network simulator. Programming and User Manual, Stuttgart, Germany, 1994.