Working Paper

Spectral Pattern Recognition and Fuzzy Artmap Classification: Design Features, System Dynamics and Real World Simulations

Manfred M. Fischer
Sucharita Gopal

WP-96-063
June 1996

IIASA International Institute for Applied Systems Analysis □ A-2361 Laxenburg □ Austria
Telephone: +43 2236 807 □ Fax: +43 2236 71313 □ E-Mail: info@iiasa.ac.at

Spectral Pattern Recognition and Fuzzy Artmap Classification: Design Features, System Dynamics and Real World Simulations

Manfred M. Fischer
Sucharita Gopal

WP-96-063
June 1996

International Institute for Applied Systems Analysis □ A-2361 Laxenburg □ Austria
Telephone: +43 2236 807 □ Fax: +43 2236 71313 □ E-Mail: info@iiasa.ac.at

Spectral Pattern Recognition and Fuzzy ARTMAP Classification:
Design Features, System Dynamics and Real World Simulations

Manfred M. Fischer
Department of Economic and Social Geography
University of Economics and Business Administration
A-1090 Vienna, Augasse 2-6

fax: ++43-1-31336-703
phone: ++43-1-31336-4836
e-mail: mmf@mailbox.wu-wien.ac.at

Sucharita Gopal
Department of Geography
Boston University
675 Commonwealth Avenue
2215 Boston, Massachusetts

fax: ++1-617-353-8399
phone: ++1-617-353-5744
e-mail: suchi@bursb.bu.edu

Paper prepared for the Book on *Recent Developments in Spatial Analysis - Spatial Statistics, Behavioural Modelling and Neurocomputing*, edited by Manfred M. Fischer and Arthur Getis, Springer Verlag, Vienna, March 1996

Abstract

Classification of terrain cover from satellite radar imagery represents an area of considerable current interest and research. Most satellite sensors used for land applications are of the imaging type. They record data in a variety of spectral channels and at a variety of ground resolutions. Spectral pattern recognition refers to classification procedures utilizing pixel-by-pixel spectral information as the basis for automated land cover classification. A number of methods have been developed in the past to classify pixels [resolution cells] from multispectral imagery to a priori given land cover categories. Their ability to provide land cover information with high classification accuracies is significant for work where accurate and reliable thematic information is needed. The current trend towards the use of more spectral bands on satellite instruments, such as visible and infrared imaging spectrometers, and finer pixel and grey level resolutions will offer more precise possibilities for accurate identification. But as the complexity of the data grows, so too does the need for more powerful tools to analyse them.

It is the major objective of this study to analyse the capabilities and applicability of the neural pattern recognition system, called fuzzy ARTMAP, to generate high quality classifications of urban land cover using remotely sensed images. Fuzzy ARTMAP synthesizes fuzzy logic and Adaptive Resonance Theory (ART) by exploiting the formal similarity between the computations of fuzzy subsethood and the dynamics of category choice, search and learning. The paper describes design features, system dynamics and simulation algorithms of this learning system, which is trained and tested for classification (8 a priori given classes) of a multispectral image of a Landsat-5 Thematic Mapper scene (270 x 360 pixels) from the City of Vienna on a pixel-by-pixel basis. Fuzzy ARTMAP performance is compared with that of an error-based learning system based upon the multi-layer perceptron, and the Gaussian maximum likelihood classifier as conventional statistical benchmark on the same database. Both neural classifiers outperform the conventional classifier in terms of classification accuracy. Fuzzy ARTMAP leads to out-of-sample classification accuracies, very close to maximum performance, while the multi-layer perceptron - like the conventional classifier - shows difficulties to distinguish between some land use categories.

Keywords: Fuzzy ARTMAP, neural networks, supervised pattern classification, remote sensing

1. Introduction

Spectral pattern recognition deals with classifications that utilize pixel-by-pixel spectral information from satellite radar imagery. The literature on neural network applications in this area is relatively new, dating back only about six to seven years. The first studies established the feasibility of error-based learning systems such as backpropagation [see Key et al. 1989, McClellan et al. 1989, Benediktsson et al. 1990, Hepner et al. 1990]. Subsequent studies analysed backpropagation networks in more detail and compared them to standard statistical classifiers such as the Gaussian maximum likelihood [see Bischof et al. 1992, Kanellopoulos et al. 1993, Fischer et al. 1994].

In this paper we analyse the capability and applicability of a different class of neural networks, called fuzzy ARTMAP, to multispectral image classification. Fuzzy ARTMAP synthesizes fuzzy logic and Adaptive Resonance Theory (ART) models by describing the dynamics of ART category choice, search and learning in terms of analog fuzzy set-theoretic rather than binary set-theoretic operations. The paper describes design features, system dynamics and simulation algorithms of this learning system, which is trained and tested for classification of a multispectral image of a Landsat-5 Thematic Mapper (TM) scene (270x360 pixels) from the City of Vienna on a pixel-by-pixel basis. Fuzzy ARTMAP performance is compared with that of a backpropagation system and the Gaussian maximum likelihood classifier on the same database.

The paper is organized in seven sections. Section 2 gives a brief mathematical description of the unsupervised learning system, called ART 1, which is a prerequisite to understanding the ARTMAP system. Section 3 shows how two ART 1 modules are linked together to form the ARTMAP supervised learning system for binary pattern recognition problems. Section 4 leads to one generalization of ARTMAP, called fuzzy ARTMAP, that learns to classify continuous valued rather than binary patterns, and to a simplified version of the general fuzzy ARTMAP learning system, which will be used as general purpose remote sensing classifier in this study. Section 5 describes the remote sensing classification problem which is used to test the classifier's capabilities. The simulation results are given in section 6 and compared with those obtained by the backpropagation network and the conventional maximum likelihood classifier. The last section contains a summary discussion.

2. Adaptive Resonance Theory and ART 1

The basic principles of adaptive resonance theory (ART) were introduced by Stephen Grossberg in 1976 as a theory of human cognitive information processing [Grossberg 1976

a,b]. Since that time the cognitive theory has led to a series of ART neural network models for category learning and pattern recognition. Such models may be characterized by a system of ordinary differential equations [Carpenter and Grossberg 1985, 1987a] and have been implemented in practice using analytical solutions or approximation to these differential equations.

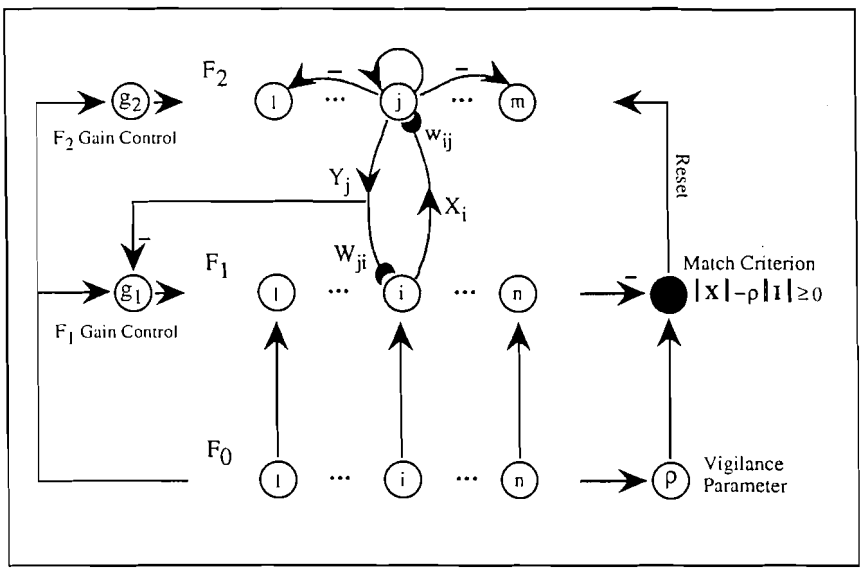

Figure 1: The ART1 Architecture

ART models come in several varieties, most of which are unsupervised, and the simplest are ART 1 designed for binary input patterns [Carpenter and Grossberg 1987a] and ART 2 for continuous valued [or binary] inputs [Carpenter and Grossberg 1987b]. This section describes the ART 1 model which is a prerequisite to understanding the learning system fuzzy ARTMAP. The main components of an ART 1 system are shown in figure 1. Ovals represent fields [layers] of nodes, semicircles adaptive filter pathways and arrows paths which are not adaptive. Circles denote nodes [processors], shadowed nodes the vigilance parameter, the match criterion and gain control nuclei that sum input signals. The F_1 nodes are indexed by i and F_2 nodes by j [categories, prototypes]. The binary vector $\mathbf{I}=(I_1,...,I_n)$ forms the bottom-up input [input layer F_0] to the field [layer] F_1 of n nodes whose activity vector is denoted by $\mathbf{X}=(X_1,...,X_n)$. Each of the n nodes in field [layer] F_2 represents a class or category of inputs around a prototype [cluster seed or recognition category] generated during self-organizing activity of ART 1. Adaptive pathways lead from each F_1 node to all F_2 nodes [bottom up adaptive filter], and from each F_2 node to all F_1 nodes [top down adaptive filter]. All paths are excitatory unless marked with a minus sign.

Carpenter and Grossberg designed the ART 1 network using previously developed building blocks based on biologically reasonable assumptions. The selection of a winner F_2 node, the

top down and bottom up weight changes, and the enable/disable [reset] mechanism can all be described by realizable circuits governed by differential equations. The description of the ART 1 simulation algorithm below is adapted from Carpenter et al. (1991a, b). We consider the case where the competitive layer F_2 makes a choice and where the ART system is operating in a fast learning mode.

F_1-Activation

Each F_1 node can receive input from three sources: the $F_0 \rightarrow F_1$ bottom-up input; non-specific gain control signals; and top-down signals from the m nodes [winner-take-all units] of F_2, via an $F_2 \rightarrow F_1$ adaptive filter. A node is said to be *active* if it generates an output signal equal to 1. Output from inactive nodes equals 0. In ART 1 a F_1 node is active if at least 2 of the 3 input signals are large. This **rule for F_1 activation** is called the **2/3 Rule** and realized in its simplest form as follows: The ith F_1 node is active if its net input exceeds a fixed threshold:

$$X_i = \begin{cases} 1 & \text{if } I_i + g_1 + \Sigma_{j=1}^{n} Y_j W_{ji} > 1 + \overline{W} \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where I_i is the binary $F_0 \rightarrow F_1$ input, g_1 the binary non-specific F_1 gain control signal, and term $\Sigma_{j=1}^{n} Y_j W_{ji}$ the sum of $F_2 \rightarrow F_1$ signals Y_j via pathways with adaptive weights W_{ji}, and $\overline{W}$ $(0 < \overline{W} < 1)$ is a constant. Hereby the F_1 *gain control* g_1 is defined as

$$g_1 = \begin{cases} 1 & \text{if } F_0 \text{ is active and } F_2 \text{ is inactive} \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

It is important to note that F_2 activity inhibits g_1, as shown in figure 1. These laws for F_1 activation imply that, if F_2 is inactive, then

$$X_i = \begin{cases} 1 & \text{if } I_i = 1 \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

If exactly one F_2 node J is active, the sum $\Sigma_{i=1}^{n} X_i W_{ji}$ in equation (1) reduces to the single term W_{Ji}, so that

$$X_i = \begin{cases} 1 & \text{if } I_i = 1 \text{ and } W_{Ji} > \overline{W} \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Rules for Category Choice [F_2 choice]

F_2 nodes interact with each other by lateral inhibition. The result is a competitive winner-take all response. The set of committed F_2 nodes [prototypes] is defined as follows. Let T_j denote the total input from F_1 to the jth F_2 processor, given by

$$T_j = \Sigma_{i=1}^{n} \; X_i \; w_{ij} \; , \tag{5}$$

where w_{ij} represent the $F_1 \rightarrow F_2$ [i.e. bottom-up or forward] adaptive weights. If some $T_j > 0$, define the F_2 choice index J by

$$T_J = \max_{j=1,\ldots,m} \{T_j\} . \tag{6}$$

Characteristically, J is uniquely defined. Then the components of the F_2 output vector $\mathbf{Y} = (Y_1,\ldots, Y_m)$ are

$$Y_i = \begin{cases} 1 & \text{if } j = J \\ 0 & \text{if } j \neq J. \end{cases} \tag{7}$$

If two or more indices j share maximal input, then one of these is chosen at random.

Learning Laws: Top Down and Bottom Up Learning

The *learning laws* as well as the rules for choice and search, may be described, using the following notation. Let $\mathbf{A} = (A_1,\ldots, A_m)$ be a binary m-dimensional vector, then the norm of **A** is defined by

$$\| \mathbf{A} \| = \Sigma_{i=1}^{m} \; | A_i | . \tag{8}$$

Let **A** and **B** be binary m-dimensional vectors, then a third binary m-dimensional vector $\mathbf{A} \cap B$ may be defined by

$$(\mathbf{A} \cap \mathbf{B}) \; = 1 \quad \text{if and only if} \quad A_i = 1 \text{ and } B_i = 1 \tag{9}$$

All *ART 1 learning* is gated by F_2 activity. That is, the bottom up (forward) and the top down (backward or feedback) adaptive weights w_{ij} and W_{ji} can change only when the Jth F_2 node is active. Both types of weights are functions of the F_1 vector $\mathbf{X}$.

Stated as a differential equation, the *top-down* or *feedback learning rule* is

$$\frac{d}{dt} W_{ji} = Y_j (X_i - W_{ji}) \tag{10}$$

where learning by W_{ji} is gated by Y_j. When the Y_j gate opens [i.e., when $Y_j > 0$], then learning begins and W_{ji} is attracted to X_i:

$$W_{ji} \rightarrow X_i \tag{11}$$

In vector terms: if $Y_j > 0$, then $\mathbf{W}_j = (W_{j1},...,W_{jn})$ approaches $\mathbf{X} = (X_1,...,X_n)$. Such a learning rule is termed *outstar learning rule* [Grossberg 1969].

Initially, all W_{ji} are maximal, i.e.

$$W_{ji}(0) = 1 \tag{12}$$

Thus [with fast learning where the adaptive weights fully converge to equilibrium values in response to each input pattern] the top-down [feedback] weight vector W_j is a binary vector at the start and end of each input presentation. By (3), (4), (9), (11) and (12), the *binary* F_1 *activity* [output] vector is given by

$$\mathbf{X} = \begin{cases} \mathbf{I} & \text{if } F_2 \text{ is inactive} \\ \mathbf{I} \cap \mathbf{W}_j & \text{if the Jth } F_2 \text{ node is active} \end{cases} \tag{13}$$

When F_2 node J is active, by (4) and (10) learning causes

$$\mathbf{W}_J \rightarrow \mathbf{I} \cap \mathbf{W}_{J\,(old)} \tag{14}$$

In this *learning update rule* $\mathbf{W}_{j(old)}$ denotes $\mathbf{W}_j$ at the start of the current input presentation. By (11) and (13), $\mathbf{X}$ remains constant during learning, even though $|\mathbf{W}_J|$ may decrease.

The first time an F_2 node J becomes active it is said to be *uncommitted* . Then, by (12) - (14)

$$\mathbf{W}_J \rightarrow \mathbf{I}. \tag{15}$$

The bottom up or forward weights have a slightly more complicated learning rule which leads to a similar, but normalized result. The combination with F_2 nodes which undergo cooperative and competitive interactions is called *competitive learning*. Initially all F_2 nodes are uncommitted. Forward weights w_{ij} in $F_1 \rightarrow F_2$ paths initially satisfy

$$w_{ij}(0) = \alpha_j. \tag{16}$$

where the parameters α_j are ordered according to $\alpha_1 > \alpha_2 > \ldots > \alpha_n$ for any admissible $F_0 \rightarrow F_1$ input I.

Like the top-down weight vector W_J, the bottom-up $F_1 \rightarrow F_2$ weight vector $\mathbf{w}_J = (w_{1J}, \ldots, w_{iJ}, \ldots, w_{nJ})$ also becomes proportional to the F_1 output vector $\mathbf{X}$ when the F_2 node J is active. But in addition the forward weights are scaled inversely to $\| \mathbf{X} \|$, so that

$$W_{iJ} \longrightarrow \frac{X_i}{\beta + \|\mathbf{X}\|} \tag{17}$$

with $\beta > 0$ [the small number β is included to break ties]. This $F_1 \rightarrow F_2$ learning law [called the Weber Law Rule, Carpenter and Grossberg 1987a] realizes a type of competition among the weights $\mathbf{W}_J$ adjacent to a given F_2 node J.

By (13), (14) and (17), during learning

$$\mathbf{w}_J \longrightarrow \frac{\mathbf{I} \cap \mathbf{W}_{J\,(\text{old})}}{\beta + \| \mathbf{I} \cap \mathbf{W}_{J\,(\text{old})} \|}. \tag{18}$$

(18) establishes the update rule for forward weights. The w_{ij} initial values are required to be sufficiently small so that an input $\mathbf{I}$ which perfectly matches a previously learned vector $\mathbf{w}_J$ will select the F_2 node J rather than an uncommitted node. This is accomplished by assuming that

$$0 < \alpha_j = w_{ij}(0) < \frac{1}{\beta + \|\mathbf{I}\|} \tag{19}$$

for all $F_1 \rightarrow F_2$ inputs $\mathbf{I}$. When $\mathbf{I}$ is first presented, $\mathbf{X} = \mathbf{I}$, so by (5), (14), (16), and (18), the $F_1 \rightarrow F_2$ input vector $T = (T_1, \ldots, T_m)$ obeys

$$T_j(\mathbf{I}) = \Sigma_{i=1}^{n} \; I_i \; w_{ij} = \begin{cases} \| \mathbf{I} \| \alpha_j & \text{if j is an uncommitted node} \\ \dfrac{\| \mathbf{I} \cap \mathbf{W}_j \|}{\beta + \| \mathbf{W}_j \|} & \text{if j is a committed node.} \end{cases} \qquad (20)$$

(20) is termed the choice function in ART 1, where ß is the choice parameter and ß ≅ 0. The limit ß→0 is called *conservative limit*, because small ß-values tend to minimize recoding during learning. If ß is taken so small then - among committed F_2 nodes - T_j is determined by the size $\| \mathbf{I} \cap \mathbf{W}_j \|$ relative to $\| \mathbf{W}_j \|$. Additionally, α_j values are taken to be so small that an uncommitted F_2 node will generate the maximum T_j value in (20) only if $\| \mathbf{I} \cap \mathbf{W}_j \| = 0$ for all committed nodes. Larger values of α_j and ß bias the system toward earlier selection of uncommitted nodes when only poor matches are to be found among the committed nodes [for a more detailed discussion see Carpenter and Grossberg 1987a].

Rules for Search

It is important to note that ART 1 overcomes the stability - plasticity dilemma by accepting and adapting the prototype of a category [class] stored in F_2 only when the input pattern is *"sufficiently similar"* to it. In this case, the input pattern and the stored prototype are said to *resonate* [hence the term *resonance* theory]. When an input pattern fails to match any existing prototype [node] in F_2, a new category is formed [as in Hartigan's (1975) leader algorithm], with the input pattern as the prototype, using a previously uncommitted F_2 unit. If there are no such uncommitted nodes left, then a novel input pattern gives no response [see Hertz et al. 1991].

A dimensionless parameter ρ with $0 < \rho \leq 1$ which is termed *vigilance parameter* establishes a matching [similarity] criterion for deciding whether the similarity is good enough for the input pattern to be accepted as an example of the chosen prototype. The degree of match [similarity] between bottom-up input **I** and top-down expectation $\mathbf{W}_j$ is evaluated at the orienting subsystem of ART 1 [see figure 1] which measures whether prototype J adequately represents input pattern **I**. A *reset* occurs when the match fails to meet the criterion established by the parameter ρ.

In fast-learning ART 1 with choice at F_2, the *search process* may be characterized by the following steps:

Step 1: Select one F_2 node J that maximizes T_j in (20), and read-out its top-down [feedback] weight vector $\mathbf{W}_J$.

Step 2: With J active, compare the F_1 output vector $\mathbf{X} = \mathbf{I} \cap \mathbf{W}_j$ with the $F_0 \rightarrow F_1$ input vector $\mathbf{I}$ at the orienting subsystem [see figure 1].

Step 3A: Suppose that $\mathbf{I} \cap \mathbf{W}_J$ fails to match $\mathbf{I}$ at the level required by the ρ-criterion, i.e. that

$$\| \mathbf{X} \| = \| \mathbf{I} \cap \mathbf{W}_j \| < \rho \| \mathbf{I} \| \tag{21}$$

This mismatch causes the system to reset and inhibits the winning node J for the duration of the input interval during which $\mathbf{I}$ remains on. The index of the chosen prototype [F_2 node] is reset to the value corresponding to the next highest $F_1 \rightarrow F_2$ input $\mathbf{T}_j$. With the new node active, steps 2 and 3A are repeated until the chosen prototype satisfies the similarity [resonance] criterion (21).

Step 3B: Suppose that $\mathbf{I} \cap \mathbf{W}_J$ meets the similarity [match function] criterion, i.e.

$$\| \mathbf{X} \| = \| \mathbf{I} \cap \mathbf{W}_j \| \geq \rho \| \mathbf{I} \| \tag{22}$$

then ART 1 search ends and the last chosen F_2 node J remains active until input $\mathbf{I}$ shuts off [or until ρ increases].

In this state, called *resonance*, both the feedforward ($F_1 \rightarrow F_2$) and the feedback ($F_2 \rightarrow F_1$) adaptive weights are updated if $\mathbf{I} \cap \mathbf{W}_{j\,(old)} \neq \mathbf{W}_{j\,(old)}$. If ρ is chosen to be large [i.e. close to 1], the similarity condition becomes very stringent so that many finely divided categories [classes] are formed. A ρ–value close to zero gives a coarse categorization. The vigilance level can be changed during learning.

Finally, it is worth noting that ART 1 is exposed to discrete presentation intervals during which an input is constant and after which F_1 and F_2 activities are set to zero. Discrete presentation intervals are implemented by means of the F_1 and F_2 gain control signals [g_1, g_2]. Gain signal g_2 is assumed [like g_1 in (2)] to be 0 if F_0 is inactive. When F_0 becomes active, g_2 and F_2 signal thresholds are assumed to lie in a range where the F_2 node which receives the largest input signal can become active.

3. The ARTMAP Neural Network Architecture

ARTMAP is a neural network architecture designed to solve supervised pattern recognition problems. The architecture is called ARTMAP because it maps input vectors in $\Re^n$ [such as feature vectors denoting spectral values of a pixel] to output vectors in $\Re^m$ [with m<n], representing predictions such as land use categories, where mapping is learned by example from pairs $\{\mathbf{A}^{(p)}, \mathbf{B}^{(p)}\}$ of sequentially presented input and output vectors p=1,2,3,... and $\mathbf{B}^{(p)}$ is the correct prediction given $\mathbf{A}^{(p)}$. Figure 2 illustrates the main components of a binary ARTMAP system. The system incorporates two ART 1 modules, ART_a and ART_b. Indices a and b identify terms in the ART_a and ART_b modules, respectively. Thus, for example ρ_a and ρ_b denote the ART_a and ART_b vigilance (similarity) parameters, respectively.

During supervised learning ART_a and ART_b read vector inputs **A** and **B**. The ART_a complementing coding preprocessor transforms the vector $\mathbf{A}=(A_1,...,A_{na})$ into the vector $I^a = (\mathbf{A}, \mathbf{A}^C)$ at the ART_a field F_0^a, where $\mathbf{A}^C$ denotes the complement of **A**. The complement coded input I^a to the recognition system is the 2na-dimensionable vector

$$\mathbf{I}^a = (\mathbf{A}, \mathbf{A}^C) = (A_1,...,A_{na}; A_1^C,..., A_{na}^C), \tag{23}$$

where

$$A_i^C := 1 - A_i . \tag{24}$$

Complement coding achieves normalization while preserving amplitude information [see Carpenter et al. 1991a]. I^a is the input to the ART_a field F_1^a. Similarly, the input to the ART_b field F_1^b is the vector $I^b = (\mathbf{B}, \mathbf{B}^C)$.

If ART_a and ART_b were disconnected, each module would self-organize category groupings for the separate input sets $\{\mathbf{A}^{(p)}\}$ and $\{\mathbf{B}^{(p)}\}$, respectively, as described in section 2. In an ARTMAP architecture design, however, ART_a and ART_b are connected by an inter-ART module, including a map field that controls the learning of an associative map from ART_a recognition categories (i.e. compressed representations of classes of examplars $\mathbf{A}^{(p)}$) to ART_b recognition categories (i.e. compressed representations of classes of examplars $\mathbf{B}^{(p)}$). Because the map field is the interface, where signals from F_2^a and F_2^b interact, it is denoted by F^{ab}. The nodes of F^{ab} have the same index j [j=1,...,m_b] as the nodes of F_2^b because there is a one-to-one correspondence between these sets of nodes.

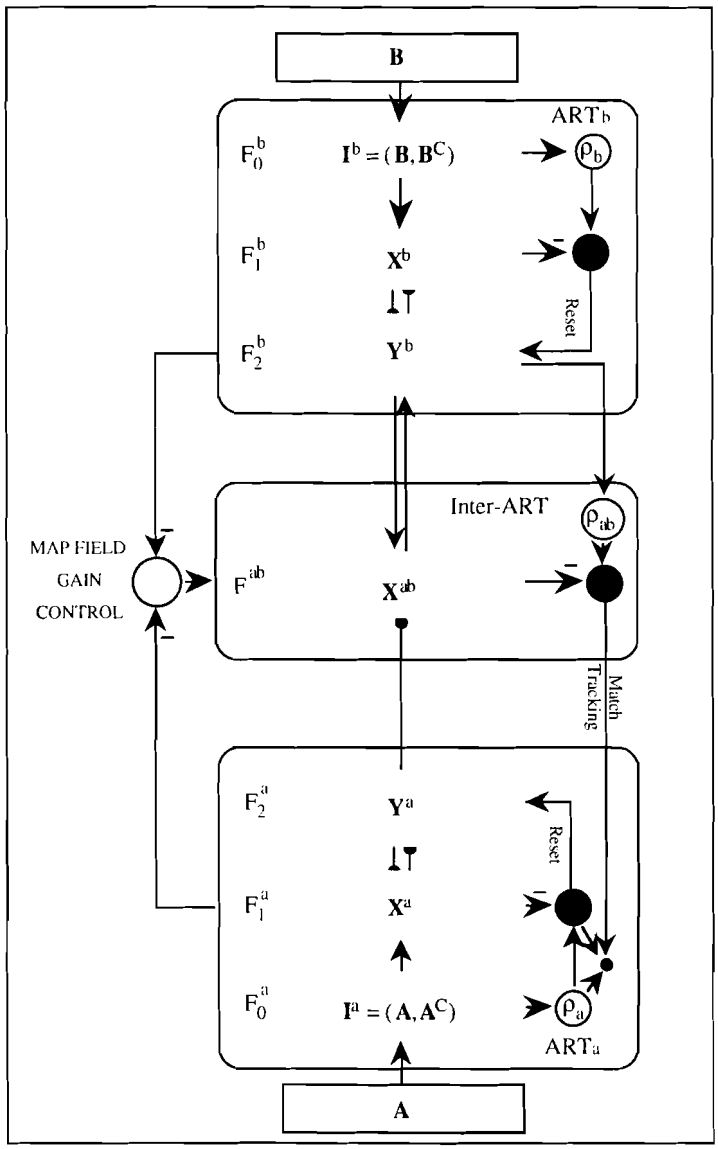

Figure 2: Block Diagram of an ARTMAP System

ART_a and ART_b operate as outlined in section 2 with the following additions. First, the ART_a vigilance [similarity] parameter ρ_a can increase during inter-ART reset according to the match tracking rule. Second, the map field F^{ab} can prime ART_b. This means, if F^{ab} sends nonuniform input to F_2^b in the absence of an $F_0^b \rightarrow F_1^b$ input **B**, then F_2^b remains inactive. But as soon as an input arrives, F_2^b selects the node **J** receiving the largest $F^{ab} \rightarrow F_2^b$ input. Node **J**, in turn, sends to F_1^b the top-down input weight vector $\mathbf{W}_J^b$. Rules for the control strategy, called match tracking, are specified in the sequel [Carpenter et al. 1991a].

Let $\mathbf{X}^a = (X_1^a, \ldots, X_{na}^a)$ denote the F_1^a output vector and $\mathbf{Y}^a = (Y_1^a, \ldots, Y_{ma}^a)$ the F_2^a output vector. Similarly, let denote $\mathbf{X}^b = (X_1^b, \ldots, X_{nb}^b)$ the F_1^b output vector and $\mathbf{Y}^b = (Y_1^b, \ldots, Y_{mb}^b)$ the F_2^b output vector. The map field F^{ab} has m_b nodes and binary output vector $\mathbf{X}^{ab}$. Vectors $\mathbf{X}^a$, $\mathbf{Y}^a$, $\mathbf{X}^b$, $\mathbf{Y}^b$ and $\mathbf{X}^{ab}$ are set to the zero vector, $\mathbf{0}$, between input presentations.

The $F_2^a \rightarrow F^{ab}$ adaptive weights z_{kj} with k=1,...,m_a and j=1,...,m_b obey an outstar learning law similar to that governing the $F_2^b \rightarrow F_1^b$ weights, namely

$$\frac{d}{dt} z_{kj} = \mathbf{Y}_k^a (\mathbf{X}_k^{ab} - z_{kj}) \tag{25}$$

Each vector $(z_{k1},...,z_{kmb})$ is denoted by $\mathbf{z}_k$. According to the learning rule established by (25), the $F_2^a \rightarrow F^{ab}$ weight vector $\mathbf{z}_k$ approaches the map field F^{ab} activity vector $\mathbf{X}^{ab}$ if the K-th F_2^a node is active. Otherwise $\mathbf{z}_k$ remains constant. If node k has not yet learned to make a prediction, all weights z_{kj} are set equal to 1, using an assumption, analogous to equation (12), i.e. $z_{kj}(0)=1$ for k=1,...,m_a, and j=1,...,m_b.

During resonance with ART_a category K active, $\mathbf{z}_K \rightarrow \mathbf{X}^{ab}$. In fast learning, once K learns to predict the ART_b category J, that association is permanent [i.e. $z_{KJ} = 1$ and $z_{Kj} = 0$ with $j \neq J$ for all time].

The F^{ab} output vector $\mathbf{X}^{ab}$ obeys

$$\mathbf{X}^{ab} = \begin{cases} \mathbf{Y}^b \cap \mathbf{z}_K & \text{if the K-th } F_2^a \text{ node is active and } F_2^b \text{ is active} \\ \mathbf{z}_K & \text{if the K-th } F_2^a \text{ node is active and } F_2^b \text{ is inactive} \\ \mathbf{Y}^b & \text{if } F_2^a \text{ is inactive and } F_2^b \text{ is active} \\ \mathbf{0} & \text{if } F_2^a \text{ is inactive and } F_2^b \text{ is inactive.} \end{cases} \tag{26}$$

When ART_a makes a prediction that is incompatible with the actual ART_b input [i.e. $\mathbf{z}_k$ is disconfirmed by $\mathbf{Y}^b$], then this mismatch triggers on ART_a search for a new category as follows. At the start of each input presentation the ART_a vigilance [similarity] parameter ρ_a equals a baseline vigilance $\overline{\rho}_a$. The map field vigilance parameter is ρ_{ab}. If a mismatch at F^{ab} occurs, i.e. if

$$\| \mathbf{X}^{ab} \| < \rho_{ab} \| \mathbf{I}^b \|, \tag{27}$$

then match tracking is triggered to search a new F_2^a node. Match tracking starts a cycle of ρ_a adjustment and increases ρ_a until it is slightly higher than the F_1^a match value $\|A \cap \mathbf{W}_k^a\| \|\mathbf{I}^a\|^{-1}$, where W_k^a denotes the top-down $F_2^a \rightarrow F_1^a$ ART_a weight vector $(W_1^a,\ldots,W_{na}^a)$. Then

$$\|\mathbf{X}^a\| = \| \mathbf{I}^a \cap \mathbf{W}_k^a \| < \rho_a \|\mathbf{I}^a\| \tag{28}$$

where I^a is the current ART_a input vector and K is the index of the active F_2^a node. When this occurs, ART_a search leads either to ARTMAP resonance, where a newly chosen F_2^a node K satisfies both the ART_a matching criterion [see also equation (21)]:

$$\| \mathbf{X}^a \| = \| I^a \cap \mathbf{W}_K^a \| \geq \rho_a \| I^a \| \tag{29}$$

and the map field matching criterion:

$$\| \mathbf{X}^{ab} \| = \| \mathbf{Y}^b \cap \mathbf{z}_K \| \geq \rho_{ab} \| \mathbf{Y}^b \| \tag{30}$$

or, if no such node K exists, to the shut-down of F_2^a for the remainder of the input presentation [Carpenter et al. 1993].

4. Generalization to Fuzzy ARTMAP

Fuzzy ARTMAP has been proposed by Carpenter et al. (1991b) as a direct generalization of ARTMAP for supervised learning of recognition categories and multidimensional maps in response to arbitrary sequences of continuous-valued [and binary] patterns not necessarily interpreted as fuzzy set of features. The generalization to learning continuous and binary input patterns is achieved by using fuzzy set operations rather than standard binary set theory operations [see Zadeh 1965]. Figure 3 summarizes how the crisp logical ARTMAP operations of category choice, matching and learning translate into fuzzy ART operations when the crisp (non-fuzzy or hard) intersection operator ($\cap$) of ARTMAP is replaced by the fuzzy intersection or [component-wise] minimum operator ($\wedge$). Due to the close formal homology between ARTMAP and fuzzy ARTMAP operations [as illustrated in figure 3], there is no need to describe fuzzy ARTMAP in detail here, but for a better understanding it is important to stress differences to the ARTMAP approach.

Fuzzy ARTMAP in its most general form inherits the architecture as outlined in figure 2 and employs two fuzzy ART modules as substitutes for the ART 1 subsystems. It is noteworthy that fuzzy ART reduces to ART 1 in response to binary input vectors [Carpenter at al. 1993]. Associated with each F_2^a [F_2^b] node $k=1,...,m_a$ [$j=1,...,m_b$] is a vector $\mathbf{W}_k^a$ [$\mathbf{W}_j^b$] of adaptive weights which subsumes both the bottom-up and top-down weight vectors of ART 1.

Fuzzy ARTMAP dynamics are determined by a choice parameter $\beta > 0$, a learning parameter $\gamma \in [0,1]$; and three vigilance [similarity] parameters: The ART_a vigilance parameter ρ_a, the

Figure 3: Comparison between Binary and Fuzzy ARTMAP

	Binary ARTMAP	Fuzzy ARTMAP
ART_a Category Choice [β choice parameter]	$T_k(\mathbf{I}^a) = \frac{\|\mathbf{I}^a \cap \mathbf{W}_k^a\|}{\beta + \|\mathbf{W}_k^a\|}$	$T_k(\mathbf{I}^a) = \frac{\|\mathbf{I}^a \wedge \mathbf{W}_k^a\|}{\beta + \|\mathbf{W}_k^a\|}$
ART_b Category Choice [β choice parameter]	$T_j(\mathbf{I}^b) = \frac{\|\mathbf{I}^b \cap \mathbf{W}_j^b\|}{\beta + \|\mathbf{W}_j^b\|}$	$T_j(\mathbf{I}^b) = \frac{\|\mathbf{I}^b \wedge \mathbf{W}_j^b\|}{\beta + \|\mathbf{W}_j^b\|}$
ART_a Matching Criterion [ρ_a ART_a vigilance parameter]	$\|\mathbf{I}^a \cap \mathbf{W}_k^a\| \geq \rho_a \|\mathbf{A}\|$	$\|\mathbf{I}^a \wedge \mathbf{W}_k^a\| \geq \rho_a \|\mathbf{A}\|$
ART_b Matching Criterion [ρ_b ART_b vigilance parameter]	$\|\mathbf{I}^b \cap \mathbf{W}_j^b\| \geq \rho_b \|\mathbf{B}\|$	$\|\mathbf{I}^b \wedge \mathbf{W}_j^b\| \geq \rho_b \|\mathbf{B}\|$
Map Field F^{ab} Matching Criterion [ρ_{ab} Map Field vigilance parameter]	$\|\mathbf{X}^{ab}\| = \|\mathbf{Y}^b \cap z_k\| \geq \rho_{ab} \|\mathbf{Y}^b\|$	$\|\mathbf{X}^{ab}\| = \|\mathbf{Y}^b \wedge z_k\| \geq \rho_{ab} \|\mathbf{Y}^b\|$
ART_a $F_2^a \rightarrow F_1^a$ Learning Weight Updates [γ learning parameter]	$\mathbf{W}_K^a(\text{new}) = \gamma (\mathbf{A} \cap \mathbf{W}_K^a(\text{old})) + (1-\gamma) \mathbf{W}_K^a(\text{old})$	$\mathbf{W}_K^a(\text{new}) = \gamma (\mathbf{A} \wedge \mathbf{W}_K^a(\text{old})) + (1-\gamma) \mathbf{W}_K^a(\text{old})$
ART_b $F_2^b \rightarrow F_1^b$ Learning Weight Updates [γ learning parameter]	$\mathbf{W}_J^b(\text{new}) = \gamma (\mathbf{B} \cap \mathbf{W}_J^b(\text{old})) + (1-\gamma) \mathbf{W}_J^b(\text{old})$	$\mathbf{W}_J^b(\text{new}) = \gamma (\mathbf{B} \wedge \mathbf{W}_J^b(\text{old})) + (1-\gamma) \mathbf{W}_J^b(\text{old})$

Note: $\cap$ denotes the crisp AND [intersection] operator and $\wedge$ its fuzzy counterpart

ART_b vigilance parameter ρ_b and the map field vigilance parameter ρ_{ab} with ρ_a, ρ_b, $\rho_{ab} \in [0,1]$. The choice functions $T_k(\mathbf{A})$ and $T_j(\mathbf{B})$ are defined as in figure 3, where the fuzzy intersection ($\wedge$) for any n-dimensional vectors $\mathbf{S}=(S_1,...,S_n)$ and $\mathbf{T}=(T_1,...,T_n)$ is defined by

$$(\mathbf{S} \wedge \mathbf{T})_i = \min_i (S_i, T_i). \tag{31}$$

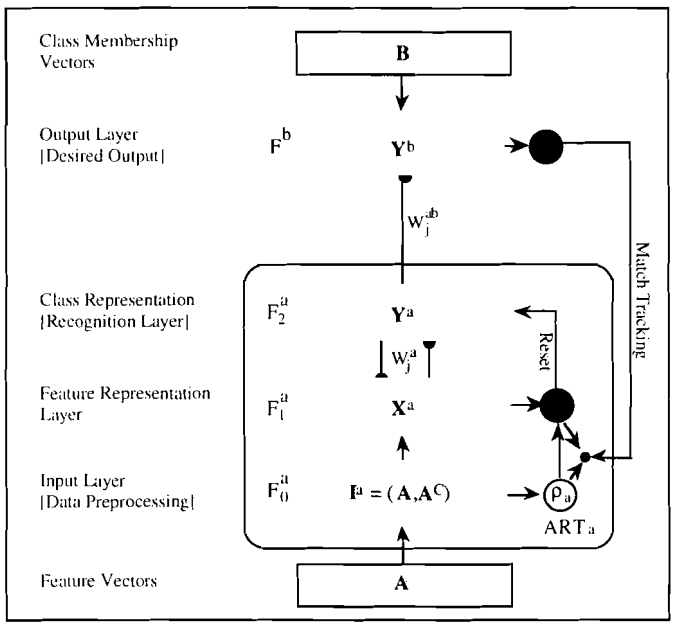

Figure 4: The Fuzzy ARTMAP Classifier: A Simplified ARTMAP Architecture

The fuzzy choice functions $T_k(\mathbf{A})$ and $T_j(\mathbf{B})$ (see figure 3) can be interpreted as a fuzzy membership of the input **A** in the k-th category and the input **B** in the j-th category, respectively. In the conservative limit (i.e. $\beta \to 0$) the choice function $T_k(\mathbf{A})$ primarily reflects the degree to which the weight vector $\mathbf{W}_k^a$ is a fuzzy subset of the input vector **A**. If

$$\frac{\| \mathbf{I}_1^a \wedge \mathbf{W}_k^a \|}{\| \mathbf{W}_k^a \|} = 1, \tag{32}$$

then $\mathbf{W}_k^a$ is a fuzzy subset of I^a and category k is said to be a *fuzzy subset* choice for input I^a. When a fuzzy subset exists, it is always selected over other choices. The same holds true for $T_j(I^b)$. [Carpenter et al. 1992]. Resonance depends on the degree to which I^a [I^b] is a fuzzy set of $\mathbf{W}_k^a$[$\mathbf{W}_k^b$], by the matching criteria [or functions] outlined in figure 3. The close linkage between fuzzy subsethood and ART choice, matching and learning forms the foundations of the computational features of fuzzy ARTMAP [Carpenter et al. 1992]. Especially if category K is a fuzzy subset ART_a choice, then the ART_a match function value ρ_a is given by

$$\rho_a = \frac{\| \mathbf{I}^a \wedge \mathbf{W}_K^a \|}{\| \mathbf{I}^a \|} = \frac{\| \mathbf{W}_K^a \|}{\| \mathbf{I}^a \|}. \tag{33}$$

Once search ends, the ART_a weight vector $\mathbf{W}_K^a$ is updated according to the equation

$$\mathbf{W}_K^a(\text{new}) = \gamma (\mathbf{A} \wedge \mathbf{W}_K^a(\text{old})) + (1 - \gamma) \mathbf{W}_K^a(\text{old}) \tag{34}$$

and similarly the ART_b weight vector $\mathbf{W}_J^b$:

$$\mathbf{W}_J^b(\text{new}) = \gamma (\mathbf{B} \wedge \mathbf{W}_J^b(\text{old})) + (1 - \gamma) \mathbf{W}_J^b(\text{old}) \tag{35}$$

where $\gamma = 1$ corresponds to fast learning as described in figure 3.

The aim of fuzzy ARTMAP is to correctly associate continuous valued ART_a inputs with continuous valued ART_b inputs. This is accomplished indirectly by associating categories formed in ART_a with categories formed in ART_b. For a pattern classification problem at hand, the desired association is between a continuous valued input vector and some categorical code which takes on a discrete set of values representing the a priori given classes. In this situation the ART_b network is not needed because the internal categorical representation which ART_b would learn already exists explicitly. Thus, the ART_b and the map field F^{ab} can be replaced by a single F^b as shown in figure 4.

5. The Spectral Pattern Recognition Problem

The spectral pattern recognition problem considered here is the supervised pixel-by-pixel classification problem in which the classifier is trained with examples of the classes [categories] to be recognized in the data set. This is achieved by using limited ground survey information which specifies where examples of specific categories are to be found in the imagery. Such ground truth information has been gathered on sites which are well represented of the much larger area analysed from space. The image data set consists of 2,460 pixels [resolution cells] selected from a Landsat-5 Thematic Mapper (TM) scene [270 x 360 pixels] from the city of Vienna and its northern surroundings (observation date: June 5, 1985; location of the center: 16°23' E, 48°14' N; TM Quarter Scene 190-026/4). The six Landsat TM spectral bands used are blue (SB1), green (SB2), red (SB3), near IR (SB4), mid IR (SB5) and mid IR (SB7), excluding the thermal band with only a 120m ground resolution. Thus, each TM pixel represents a ground area of 30 x 30 m^2 and has six spectral band values ranging over 256 digital numbers (8 bits).

The purpose of the multispectral classification task at hand is to distinguish between the eight categories of urban land use listed in Table 1. The categories chosen are meaningful to photo-interpreters and land use managers, but are not necessarily spectrally homogeneous. This prediction problem, used to evaluate the performance of fuzzy ARTMAP in a real world context, is challenging. The pixel-based remotely sensed spectral band values are noisy and sometimes unreliable. The number of training sites is small relative to the number of land use categories [one-site training case]. Some of the urban land use classes are sparsely distributed in the image. Conventional statistical classifiers such as the Gaussian maximum likelihood classifier have been reported to fail to discriminate spectrally inhomogeneous classes such as C6 [see, e.g., Hepner et al. 1990]. Thus, there is evidently a need for new more powerful tools [Barnsley 1993].

Table 1: Categories Used for Classification and Number of Training/Testing Pixels

Category Number	Description of the Category	Pixels	
		Training	Testing
C1	Mixed grass and arable farmland	167	83
C2	Vineyards and areas with low vegetation cover	285	142
C3	Asphalt and concrete surfaces	128	64
C4	Woodland and public gardens with trees	402	200
C5	Low density residential and industrial areas (suburban)	102	52
C6	Densely built up residential areas (urban)	296	148
C7	Water courses	153	77
C8	Stagnant water bodies	107	54
Total Number of Pixels for Training and Testing		1,640	820

Ideally, the ground truth at every pixel of the scene should be known. Since this is impractical, one training site was chosen for each of the eight above mentioned land use categories. The training sites vary between 154 pixels [category: suburban] and 602 pixels [category: woodland and public gardens with trees]. The above mentioned six TM bands provide the data set input for each pixel, with values scaled to the interval [0,1]. This approach resulted in a data base consisting of 2,460 pixels [about 2.5 percent of all the pixels in the scene] that are described by six-dimensional feature vectors, each tagged with its correct category membership. The set was divided into a training set [two thirds of the training site pixels] and a testing set by stratified random sampling, stratified in terms of the eight categories. Pixels from the testing set are not used during network training [parameter estimation] and serve only to evaluate out-of-sample test [prediction, generalization] performance accuracy when the trained classifier is presented with novel data. The goal is to predict the correct land use category for the test sample of pixels.

Ideally, a good classifier is one which after training with the training set of pixels is able to predict pixel assignments over much wider areas of territory from the remotely sensed data without the need for further ground survey [see Wilkinson et al. 1995]. The performance of any classifier, thus, depends upon three factors: the adequacy of the training set of pixels and, therefore, the choice of the training sites; the in-sample performance of the classifier; and the out-of-sample or generalization performance of the trained classifier. Of these three factors, the first is often outside the control of the data analyst, and thus outside of the scope of this paper.

6. Fuzzy ARTMAP Simulations and Classification Results

In this real world setting, fuzzy ARTMAP performance is examined and compared with that of multi-layer perceptron and that of the conventional maximum likelihood classifier. In-sample and out-of sample performance is measured in terms of the fraction of the total number of correctly classified pixels [i. e. the sum of the elements along the main diagonal of the classification error matrix].

During training and testing, a given pixel provides an ART_a input $\mathbf{A}=(A_1, A_2, A_3, A_4, A_5, A_6)$ where A_1 is the blue, A_2 the green, A_3 is the red, A_4 the near infrared, A_5 and A_6 the mid infrared [1.55-1.75 μm and 2.08-2.35 μm, respectively] spectral band values measured at each pixel. The corresponding ART_b input vector **B** represents the correct land use category of the pixel's site:

$$\mathbf{B} = \begin{cases} (1,0,0,0,0,0,0,0) & \text{for mixed grass and arable farmland; category 1} \\ (0,1,0,0,0,0,0,0) & \text{for vineyards and areas with low vegetation cover; category 2} \\ (0,0,1,0,0,0,0,0) & \text{for asphalt and concrete surfaces; category 3} \\ (0,0,0,1,0,0,0,0) & \text{for woodland and public gardens with trees; category 4} \\ (0,0,0,0,1,0,0,0) & \text{for low density residential and industrial areas; category 5} \\ (0,0,0,0,0,1,0,0) & \text{for densely built-up residential areas; category 6} \\ (0,0,0,0,0,0,1,0) & \text{for water courses; category 7} \\ (0,0,0,0,0,0,0,1) & \text{for stagnant water bodies; category 8} \end{cases}$$

During training vector **B** informs the fuzzy ARTMAP classifier of the land use category to which the pixel belongs. This supervised learning process allows adaptive weights to encode the correct associations between **A** and **B**. The remote sensing problem described in section 5 requires a trained fuzzy ARTMAP network to predict the land use category of the test set pixels, given six spectral band values measured at each pixel.

Fuzzy ARTMAP is trained incrementally, with each spectral band vector **A** presented just once. Following a search, if necessary, the classifier selects an ART_a category by activating an F_2^a

node K for the chosen pixel, and learns to associate category K with the ART_b land use category of the pixel. With fast learning (γ=1), the class prediction of each ART_a category K is permanent. If some input **A** with a different class prediction later chooses this category, match tracking will raise vigilance ρ_a just enough to trigger a search for a different ART_a category. If the finite input set is presented repeatedly, then all training set inputs learn to predict with 100% classification accuracy, but start to fit noise present in the remotely sensed spectral band values.

All the simulations described below use the simplified fuzzy ARTMAP architecture outlined in figure 3, with three parameters only: a choice parameter $\beta>0$, the learning parameter γ=1 [fast learning], and an ART_a vigilance parameter $\rho_a \in]0,1]$. In each simulation, the training data set represents 1,640 pixels and the testing data set 820 pixels. Fuzzy ARTMAP was run with five different random orderings of the training and test sets, since input order may affect in-sample and out-of-sample performance. All simulations were carried out at the Department of Economic Geography (WU-Wien) on a SunSPARCserver 10-GS with 128 MB RAM.

Training Time in Epochs

Figure 5: In-Sample and Out-of-Sample Classification Error During Training
(β=0.001, γ=1.0, ρ_a=0.001)

Training Time in Epochs

Figure 6: Effect of Choice Parameter β on the Number (m_a) of ARTa Categories
(γ=1.0, ρ_a=0.001)

Table 2 summarizes out-of-sample performance [measured in terms of classification accuracy] on 15 simulations, along with the number of ART_a categories generated and the number of

epochs needed to reach any asymptotic training set performance [i. e. about 100% in-sample classification accuracy; figure 5 shows how in-sample and out-of-sample performance changes depending on the number of training epochs of fuzzy ARTMAP]. Each run had a different, randomly chosen presentation order for the 1,640 training and the 820 testing vectors. The choice parameter β was set, first, near the conservative limit at value β=0.001, and then at the higher values of β=0.1 and β=1.0. These β-value inputs were repeatedly presented in a given random order until 100% training classification accuracy was reached. This required six to eight epochs in the cases of β=0.001 and β=0.1, while for β=1.0 eight to ten epochs were necessary. There seems to be a tendency that the number of epochs needed for 100% training set performance is increasing with higher β-values. All simulations used fast learning [γ=1.0], which generates a distinct ART_a category structure for each input ordering. The

Table 2: Fuzzy ARTMAP Simulations of the Remote Sensing Classification Problem: The Effect of Variations in Choice Parameter β [ρ_a=0.0]

Choice Parameter β	Out-of-Sample Performance	Number of F_a^2 Nodes	Number of Epochs
β = 0.001			
Run 1	98.54	125	6
Run 2	99.26	116	8
Run 3	99.51	121	6
Run 4	99.39	126	6
Run 5	99.75	148	7
Average	99.29	127	6.5
β = 0.1			
Run 1	99.26	126	7
Run 2	99.90	115	6
Run 3	99.36	115	7
Run 4	99.51	124	7
Run 5	99.26	127	7
Average	99.26	121	7
β = 1.0			
Run 1	99.98	218	10
Run 2	98.17	202	8
Run 3	98.90	212	8
Run 4	98.50	236	10
Run 5	98.40	232	10
Average	98.75	220	9

Table 3: Fuzzy ARTMAP Simulations of the Remote Sensing Classification Problem: The Effect of Variations in Vigilance ρ_a [$\beta = 0.001$]

Vigilance (Similarity) Parameter ρ_a	In-Sample Performance	Out-of-Sample Performance	Number of F_a^2 Nodes
$\rho_a = 0.95$			
Run 1	97.01	96.20	285
Run 2	97.00	96.20	298
Run 3	96.15	95.60	276
Run 4	96.21	95.36	276
Run 5	95.06	94.39	286
Average	96.36	95.82	284
$\rho_a = 0.75$			
Run 1	93.00	92.00	52
Run 2	92.26	93.29	47
Run 3	91.82	90.00	42
Run 4	93.00	93.04	53
Run 5	90.31	91.83	53
Average	92.08	92.03	50
$\rho_a = 0.50$			
Run 1	92.20	91.40	43
Run 2	90.20	89.51	43
Run 3	94.45	94.76	44
Run 4	93.35	93.42	43
Run 5	92.98	93.90	45
Average	92.62	92.59	44
$\rho_a = 0.0$			
Run 1	90.70	90.60	35
Run 2	92.26	91.22	44
Run 3	90.97	90.30	34
Run 4	91.95	90.73	40
Run 5	92.56	92.44	32
Average	91.69	91.06	37

number of F_2^a nodes ranged from 116 to148 in the case of β=0.001, 115 to 127 in the case of β=0.1, and 202 to 236 in the case of β=1.0. This tendency of increasing number of ART_a categories with increasing β-values and increasing training time is illustrated in figure 6. All simulations used ρ_a= 0.0 which tends to minimize the number of F_2^a nodes compared with higher ρ_a-values not shown in Table 2. The best average result [averaged over five independent simulation runs] was obtained with β=0.01 and 6.5 epoch training [99.29% classification accuracy]. All the 15 individual simulation runs reached an out-of-sample performance close to 100% [range: 98.40 to 99.90%].

Table 3 shows how in-sample and out-of-sample performance changes depending on the number of F_2^a nodes with ρ_a=0.95, 0.75, 0.50 and 0.0. In these simulations, learning is incremental, with each input presented only once [in ART terminology: one epoch training]. The choice parameter is set to β=0.001. The best overall results, in terms of average in-sample and out-of-sample performance were obtained with an ART_a vigilance close to one [96.36% and 95.82%, respectively]. For ρ_a=0.0 the in-sample and out-of-sample performances decline to 91.69% and 91.06%, respectively. But the runs with ρ_a= 0.0 use much fewer ART_a categories [32 to 44] compared to ρ_a= 0.95 [276 to 298 ART_a categories], and generate stable performance over the five runs. Increasing vigilance creates more ART_a categories. One final note to be made here is that most fuzzy ARTMAP learning occurs on the first epoch, with the test set performance on systems trained for one epoch typically over 92% that of systems exposed to inputs for six to eight epochs (compare Table 3 with Table 2).

Table 4: Fuzzy ARTMAP Simulations of the Remote Sensing Classification Problem: The Effect of Variations in Training Size [ρ_a= 0.0, β = 0.001]

Number of Training Pixels	In-Sample Performance	Out-of-Sample Performance	Number of F_a^2 Nodes
164	83.2	80.1	19
1,640	93.0	92.0	33
16,400	99.3	99.2	135
164,000	99.3	99.2	225

Table 4 summarizes the results of the third set of fuzzy ARTMAP simulations carried out, in terms of both in-sample and out-of-sample performance along with the number of F_2^a nodes. The choice parameter β was set near the conservative limit at value β=0.001 and ART_a vigilance at ρ_a= 0.0. Training lasted for one epoch only. As training size increases from 164 to 164,000 pixel vectors both in-sample and out-of-sample performances increase, but so does the number of ART_a category nodes. In-sample classification accuracy increases from 83,2% to 99,3%, and out-of-sample classification accuracy from 80.1% to 99.2%, while the number of ART_a category nodes increases from 19 to 225. Each category node k requires six learned weights

W_k^a in Art_a. One epoch training on 164 training pixels creates 19 ART_a categories and so uses 72 ART_a adaptive weights to achieve 80.1% out-of-sample classification accuracy [820 test pixels], while one epoch training on 164,000 pixels requires 225 ART_a categories and, thus, 1,350 ART_a adaptive weights to arrive at an out-of-sample performance of 99,2%. Evidently, the fuzzy ARTMAP classifier becomes arbitrarily accurate provided the number of F_2^a nodes increases as needed.

Table 5: Performance of Fuzzy ARTMAP Simulations of the Remote Sensing Classification Problem: Comparison with the Multi-Layer Perceptron and the Gaussian Maximum Likelihood Classifier

Classifier	Epochs*	Hidden Units/ ART_a Categories	Adaptive Weight Parameters	In-Sample Classification Accuracy	Out-of-Sample Classification Accuracy	Computation Costs (in terms of CPU time)
Fuzzy ARTMAP	8	116	812	100.00	99.26	2.1
Multi-Layer Perceptron	92	14	196	92.13	89.76	15.1
Gaussian Maximum Likelihood	-	-	-	90.85	85.24	1.4

*one pass through the training data set

Fuzzy ARTMAP: $\beta = 00.1$, $\gamma = 1.0$, $\rho_a = 0.0$, asymptotic training

Multi-Layer-Perceptron: logistic hidden unit activation, softmax output unit activation, network pruning, epoch based stochastic version of backpropagation with epoch size of three, learning rate $\gamma = 0.8$

Finally, fuzzy ARTMAP performance is compared with that of a multi-layer perceptron classifier as developed and implemented in Fischer et al. [1994], using the same training and testing set data. Table 5 summarizes the results of the comparison of the two neural classifers in terms of the in-sample and out-of-sample classification accuracies and the CPU-time along with the number of epochs [i. e. one pass through the training data set] and the number of hidden units/ART_a category nodes [a hidden unit is somewhat analoguos to an ART_a category for purposes of comparison] to reach asymptotic convergence. The fuzzy ARTMAP classifier has been designed with the following specifications: choice parameter near the conservative limit at value β=0.001, learning parameter γ=1.0, constant ART_a vigilance ρ_a=0.0, repeatedly presentation of inputs in a given order until 100% training set performance was reached. Stability and match tracking allow fuzzy ARTMAP to construct automatically as many ART_a categories as are needed to learn any consistent training set to 100% classifications accuracy. The multi-layer perceptron classifier is a pruned feedforward network with 14 logistic hidden units and eight softmax output units, using an epoch-based stochastic version of the backpropagation algorithm (epoch size: 3 training vectors, no momentum update, learning parameter γ=0.8). The Gaussian maximum likelihood classifier based on parametric density estimation by maximum likelihood was chosen because it represents a widely used standard for comparison that yields minimum total classification error for Gaussian class distributions.

Figure 7: The Fuzzy ARTMAP Classified Image

The fuzzy ARTMAP classifier has an outstanding out-of-sample classification accuracy of 99.26% on the 820 pixels testing data set. Thus the error rate (0.74%) is less than 1/15 that of the multi-layer perceptron and 1/20 that of the Gaussian maximum likelihood classifier. A more careful inspection of the classification error [confusion] matrices [see appendix] shows that there is a significant confusion between the urban [densely built-up residential areas] and water courses land use categories in the case of both the multi-layer perceptron and the Gaussian maximum likelihood classifiers, though the multi-layer perceptron outperforms the Gaussian maximum likelihood algorithm by 5 per cent points. The fuzzy ARTMAP neural network approach evidently accommodates more easily a heterogeneous class label such as *densely built-up residential areas* to produce a visually and numerically correct map, even with smaller numbers of training pixels [see figure 7].

The primary computational difference between the fuzzy ARTMAP and the multi-layer perceptron algorithms is speed. The backpropagation approach to neural network training is extremely computation-intensive, taking about one order of magnitude more time than the time

for fuzzy ARTMAP, when implemented on a serial workstation. Although this situation may be alleviated with other, more efficient training algorithms and parallel implementation, it remains one important drawback to the routine use of multi-layer perceptron classifiers. Finally, it should be mentioned that in terms of total number of pathways [i.e. the number of weight parameters] needed for the best performance, the multilayer perceptron classifier is superior to fuzzy ARTMAP, but at the above mentioned higher computation costs and the lower classification accuracies.

Finally, it should be mentioned that in terms of total number of pathways [i.e. the number of weight parameters] needed for the best performance, the multilayer perceptron classifier is superior to fuzzy ARTMAP, but at the above mentioned higher computation costs and the lower classification accuracies.

7. Summary and Conclusions

Classification of terrain cover from satellite radar imagery represents an area of considerable current interest and research. Satellite sensors record data in a variety of spectral channels and at a variety of ground resolutions. The analysis of remotely sensed data is usually achieved by machine-oriented pattern recognition techniques, of which classification based on maximum likelihood, assuming Gaussian distribution of the data, is the most widely used one. We compared fuzzy ARTMAP performance with that of an error-based learning system based i. e. the multi-layer perceptron and the Gaussian maximum-likelihood classifier as conventional statistical benchmark on the same database. Both neural network classifiers outperform the conventional classifier in terms of map user's, map producer's and total classification accuracies. The fuzzy ARTMAP simulations did lead by far to the best out-of-sample classification accuracies, very close to maximum performance.

Evidently, the fuzzy ARTMAP classifier accommodates more easily a heterogenenous class label such as "densely built-up residential areas" to produce a visually and numerically correct urban land use map, even with smaller numbers of training pixels. In particular, the Gaussian maximum likelihood classifier tends to be sensitive to the purity of land use category signatures and performs poorly if they are not pure.

The study shows that the fuzzy ARTMAP classifier is a powerful tool for remotely sensed image classification. Even one epoch of fuzzy ARTMAP training yields close to maximum performance. The unique ART features such as speed and incremental learning may give the fuzzy ARTMAP multispectral image classifier the potential to become a standard tool in remote sensing especially when it comes to use data from future multichannel satellites such as the 224

channel Airborne Visible and Infrared Imaging Spectrometer [AVIRIS], and to classifiying multi-data and multi-temporal imagery or when extending the same classification to different images. In conclusion, we would like to mention that the classfier leads to crisp rather than fuzzy classifications, and, thus, looses some attractivity of *fuzzy* pattern recognition systems. This is certainly, one direction for further improving the classifier.

Acknowledgements: The authors thank Professor Karl Kraus (Department of Photogrammetric Engineering and Remote Sensing, Vienna Technical University) for his assistance in supplying the remote sensing data used in this study. This work has been funded by the Austrian Ministry for Science, Research and Art [funding contract no. EZ 308.937/2-W/3/95]. The authors also like to thank Petra Staufer (Wirtschaftsuniversität Wien) for her help.

References

Barnsley, M. (1993): Monitoring urban areas in the EC using satellite remote sensing, *GIS Europe* 2(8), pp. 42-44.

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 Transactions on Geoscience and Remote Sensing* 28, pp. 540-552.

Bezdak, J. C. and Pal, S. K. (eds.) (1992): *Fuzzy Models for Pattern Recognition*. New York. IEEE.

Bischof, H., Schneider, W. and Pinz, A. J. (1992): Multispectral classification of Landsat-images using neural networks, *IEEE Transactions on Geoscience and Remote Sensing* 30 (3), pp. 482-490.

Carpenter, G. A. (1989): Neural network models for pattern recognition and associative memory, *Neural Networks* 2, pp. 243-257.

Carpenter, G. A. and Grossberg, S. (1985): Category learning and adaptive pattern recognition, a neural network model, Proceedings of the Third Army Conference on Applied Mathematics and Computing, ARO-Report 86-1, pp. 37-56.

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

Carpenter, G. A. and Grossberg, S. (1987b): ART 2: Stable self-organizing of pattern recognition codes for analog input patterns, *Applied Optics* 26, pp. 4919-4930.

Carpenter, G. A. and Grossberg, S. (eds.) (1991). *Pattern Recognition by Self-organizing Neural Networks*. Cambridge, MA: MIT Press.

Carpenter, G. A. and Grossberg, S. (1995). Fuzzy ART. In B. Kosko (ed.) *Fuzzy Engineering*, Carmel: Prentice Hall.

Carpenter, G. A., Grossberg, S. and Reynolds, J. H. (1991a). ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network, *Neural Networks* 4, pp. 565-588.

Carpenter, G. A., Grossberg, S. and and Rosen, D. B. (1991b). Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system, *Neural Networks* 4, pp. 759-771.

Carpenter, G. A., Grossberg, S. and Ross, W. D. (1993). ART-EMAP: A neural network architecture for object recognition by evidence accumulation, *Proceedings* of the World Congress on Neural Networks (WCNN-93), Hillsdak, N.J.: Lawrence Ertbaum Associates, III, pp. 643-656.

Carpenter, G. A., Grossberg, S., Markuzon, N., Reynolds, J.H., and Rosen, D. B. (1992). Fuzzy ART: A neural network architecture for incremental supervised learning of analog multidimensional maps, *IEEE Transactions on Neural Networks* 3, pp. 698-713.

Dawson, M. S., Fung, A. K. and Manry, M. T. (1993): Surface parameter retrieval using fast learning neural networks, *Remote Sensing Reviews* 7, pp. 1-18.

Fischer, M. M., Gopal, S., Staufer, P. and Steinnocher, K. (1994). Evaluation of neural pattern classifiers for a remote sensing application, Paper presented at the 34th European Congress of the Regional Science Association, Groningen, August 1994.

Grossberg, S. (1969): Some networks that can learn, remember, and reproduce any number of complicated space-time patterns I, *Journal of Mathematics and Mechanics* 19, pp. 53-91.

Grossberg, S. (1976a): Adaptive pattern classification and universal recoding, I: Parallel development and coding of neural feature detectors, *Biological Cybernetics* 23, pp. 121-134.

Grossberg, S. (1976b): Adaptive pattern classification and universal recoding, II: Feedback, expectation, olfaction and illusion, *Biological Cybernetics* 23, pp. 121-134.

Grossberg, S. (1988): Nonlinear neural networks: Principles, mechanisms, and architectures, *Neural Networks* 1, pp. 17-61.

Hara, Y., Atkins, R. G., Yueh, S. H., Shin, R. T. and Kong, J. A. (1994). Application of neural networks to radar image classification, *IEEE Transactions on Geoscience and Remote Sensing* 32, pp. 100-109.

Hartigan, J. (1975): *Clustering Algorithms*, New York et al.: Wiley.

Hepner, G. F., Logan, T., Ritter, N. and Bryant, N. (1990). Artificial neural network classification using a minimal training set comparison of conventional supervised classification, *Photogrammetric Engineering and Remote Sensing* 56, pp. 469-473.

Hertz, J., Krogh, A. and Palmer, R. G. (1991): *Introduction to the Theory of Neural Computation*. Redwood City (Cal.) et al.: Addison-Wesley.

Kanellopoulos, I., Wilkinson, G. G. and Mégier, J. (1993): Integration of neural network and statistical image classification for land cover mapping, *Proceedings of IGARSS*, pp. 511-513. Tokyo.

Key, J., Maslanik, J. A. and Schweiger, A. J. (1989): Classification of merged AVHRR and SMMR arctic data with neural networks, *Photogrammetric Engineering and Remote Sensing* 55 (9), pp. 1331-1338.

McClellan, G. E., DeWitt, R. N., Hemmer, T. H., Matheson, L. N. and Moe, G. O. (1989): Multispectral image-processing with a three-layer backpropagation network, *Proceedings of 1989 International Joint Conference on Neural Networks*, pp. 151-153. Washington, D.C.

Wilkinson, G. G., Fierens, F. and Kanellopoulos, F. (1995): Integration of neural and statistical approaches in spatial data classification, *Geographical Systems* 2, pp. 1-20.

Zadeh, L. (1965). Fuzzy sets, *Information and Control* 8, pp. 338-353.

Appendix A: In-Sample and Out-of-Sample Classification Error Matrices of the Classifiers

An error matrix is a square array of numbers set out in rows and columns which expresses the number of pixels assigned to a particualr category relative to the actual category as verified by some reference (ground truth) data. The columns represent the reference data, the rows indicate the categorization generated. It is important to note that differences between the map classification and reference data might be not only due to classification errors. Other possible sources of errors include errors in interpretation and delineation of the reference data, changes in land use between the data of the remotely sensed data and the data of the reference data (temporal error), variation in classification of the reference data due to inconsistencies in human interpretation etc.

Table A1: In-Sample Performance: Error Classification Matrices

(a) Fuzzy ARTMAP

Ground Truth Categories	Classifier's Categories								
	C1	C2	C3	C4	C5	C6	C7	C8	Total
C1	167	0	0	0	0	0	0	0	167
C2	0	285	0	0	0	0	0	0	285
C3	0	0	128	0	0	0	0	0	128
C4	0	0	0	402	0	0	0	0	402
C5	0	0	0	0	102	0	0	0	102
C6	0	0	0	0	0	293	3	0	296
C7	0	0	0	0	0	5	148	0	153
C8	0	0	0	0	0	0	0	107	107
Total	167	285	128	402	102	298	151	107	1.640

(b) Multi-Layer Perceptron

Ground Truth Categories	Classifier's Categories								
	C1	C2	C3	C4	C5	C6	C7	C8	Total
C1	157	10	0	0	0	0	0	0	167
C2	1	282	0	0	2	0	0	0	285
C3	0	0	128	0	0	0	0	0	128
C4	4	0	0	389	9	0	0	0	402
C5	0	0	2	2	98	0	0	0	102
C6	0	0	1	0	0	260	25	10	296
C7	0	0	0	0	0	60	93	0	153
C8	0	0	0	0	0	3	0	104	107
Total	162	292	131	391	109	323	118	114	1.640

(c) Gaussian Maximum Likelihood

Ground Truth Categories	Classifier's Categories								
	C1	C2	C3	C4	C5	C6	C7	C8	Total
C1	161	5	0	1	0	0	0	0	167
C2	0	284	0	0	1	0	0	0	285
C3	0	0	124	0	4	0	0	0	128
C4	0	4	0	385	13	0	0	0	402
C5	0	0	0	0	102	0	0	0	102
C6	0	0	3	0	0	214	62	17	296
C7	0	0	0	0	0	37	116	0	153
C8	0	0	0	0	0	3	0	104	107
Total	161	293	127	386	120	254	178	121	1.640

Table A2: Out-of-Sample Performance Error Classification Matrices

(a) Fuzzy ARTMAP

Ground Truth Categories	Classifier's Categories								
	C1	C2	C3	C4	C5	C6	C7	C8	Total
C1	83	0	0	0	0	0	0	0	83
C2	0	142	0	0	0	0	0	0	142
C3	0	0	64	0	0	0	0	0	64
C4	0	0	0	200	0	0	0	0	200
C5	0	0	0	0	52	0	0	0	52
C6	0	0	0	0	0	146	2	0	146
C7	0	0	0	0	0	2	75	0	77
C8	0	0	0	0	0	0	0	54	54
Total	83	142	64	200	52	148	77	54	820

(b) Multi-Layer Perceptron

Ground Truth Categories	Classifier's Categories								
	C1	C2	C3	C4	C5	C6	C7	C8	Total
C1	79	4	0	0	0	0	0	0	83
C2	1	134	6	0	1	0	0	0	142
C3	0	0	64	0	0	0	0	0	64
C4	3	2	0	194	1	0	0	0	200
C5	0	3	0	0	49	0	0	0	52
C6	0	0	0	0	0	115	30	3	148
C7	0	0	0	0	0	29	48	0	77
C8	0	0	0	0	0	1	0	53	54
Total	83	143	70	194	51	145	78	56	820

(c) Gaussian Maximum Likelihood

Ground Truth Categories	Classifier's Categories								
	C1	C2	C3	C4	C5	C6	C7	C8	Total
C1	80	3	0	0	0	0	0	0	83
C2	0	141	0	0	1	0	0	0	142
C3	0	0	62	0	1	1	0	0	64
C4	1	3	0	191	5	0	0	0	200
C5	0	5	0	0	47	0	0	0	52
C6	0	0	1	0	2	73	64	8	148
C7	0	0	0	0	0	24	53	0	77
C8	0	0	0	0	0	2	0	52	54
Total	81	152	63	191	56	100	117	60	820

Appendix B: In-Sample and Out-of-Sample Map User's and Map Producer's Accuracies of the Classifiers

Table B1: In-Sample Map User's and Map Producer's Accuracies

Category	Name	Map User's Accuracy			Map Producer's Accuracy		
		Fuzzy ARTMAP	Multi-Layer Perceptron	Gaussian Maximum Likelihood	Fuzzy ARTMAP	Multi-Layer Perceptron	Gaussian Maximum Likelihood
C1	Mixed grass & arable farmland	100.0	94.0	96.4	100.0	96.9	95.1
C2	Vineyards & areas with low vegetation cover	100.0	98.9	99.6	100.0	96.9	96.9
C3	Asphalt & concrete surfaces	100.0	100.0	96.9	100.0	97.7	97.7
C4	Woodlands & public gardens with trees	100.0	96.8	95.8	100.0	99.5	97.7
C5	Low density residential & industrial areas (suburban)	100.0	96.1	100.0	100.0	89.9	87.3
C6	Densely built up residential & industrial areas (urban)	99.0	87.8	72.3	98.3	80.5	79.8
C7	Water courses	96.7	60.8	75.8	98.0	78.8	78.8
C8	Stagnant water bodies	100.0	97.2	97.2	100.0	91.2	85.8

Note: Map user's accuracies for land use categories are calculated by dividing the number of correctly classified pixels in each category [i.e. the main diagonal elements of the classification error matrix] by the row totals.

Map producer's accuracies for land use categories are calculated by dividing the numbers of correctly classified pixels in each category [i.e. the main diagonal elements of the classification error matrix] by the columns totals.

Table B2: Out-of-Sample Map User's and Map Producer's Accuracies

Category	Name	Map User's Accuracy			Map Producer's Accuracy		
		Fuzzy ARTMAP	Multi-Layer Perceptron	Gaussian Maximum Likelihood	Fuzzy ARTMAP	Multi-Layer Perceptron	Gaussian Maximum Likelihood
C1	Mixed grass & arable farmland	100.0	95.2	96.4	100.0	95.2	98.8
C2	Vineyards & areas with low vegetation cover	100.0	94.4	99.3	100.0	93.7	92.8
C3	Asphalt & concrete surfaces	100.0	100.0	96.9	100.0	91.4	98.4
C4	Woodlands & public gardens with trees	100.0	97.0	95.5	100.0	100.0	100.0
C5	Low density residential & industrial areas (suburban)	100.0	94.2	90.4	100.0	96.1	83.9
C6	Densely built up residential & industrial areas (urban)	98.6	77.7	49.3	98.6	79.3	73.0
C7	Water courses	97.4	62.3	68.8	97.4	61.5	45.3
C8	Stagnant water bodies	100.0	98.1	96.3	100.0	86.9	86.7

Note: Map user's accuracies for land use categories are calculated by dividing the number of correctly classified pixels in each category [i.e. the main diagonal elements of the classification error matrix] by the row totals.

Map producer's accuracies for land use categories are calculated by dividing the numbers of correctly classified pixels in each category [i.e. the main diagonal elements of the classification error matrix] by the columns totals.