Salzburg Interest Group on Integrated Systems

siGis-004-1995 December 1, 1995

Land Cover Classification of Landsat Images Using Problem–Adapted Artificial Neural Networks

International Workshop on Soft Computing in Remote Sensing Data Analysis, December 4–5, 1995, Milan, Italy

Roland Schwaiger
rschwaig@cosy.sbg.ac.at

Reinhold Huber
reini@cosy.sbg.ac.at

Helmut A. Mayer
helmut@cosy.sbg.ac.at

Department of Computer Science
University of Salzburg

siGis
Salzburg Interest Group on Integrated Systems

Correspondence to:

Universität Salzburg
Institut für Computerwissenschaften und Systemanalyse
Jakob–Haringer–Straße 2
A–5020 Salzburg
Austria

Land Cover Classification of Landsat Images Using Problem–Adapted Artificial Neural Networks

Roland Schwaiger, Reinhold Huber and Helmut A. Mayer
(rschwaig@cosy.sbg.ac.at, rhuber@cosy.sbg.ac.at, helmut@cosy.sbg.ac.at)

University of Salzburg, Department of Computer Science
Jakob–Haringer–Straße 2, A–5020 Salzburg, Austria

ABSTRACT

Artificial Neural Networks (ANN) have gained increasing popularity as an alternative to statistical methods for classification of Remote Sensing Data. Their superiority to some of the classical statistical methods has been shown in the literature. Therefore, ANNs are commonly used for segmentation and classification purposes.

In this paper we address a land cover classification problem using multi–spectral Landsat Thematic Mapper (TM) data employing ANNs. We concentrate on the search for the problem–adapted network topology and the appropriate number of training epochs for Multi–Layer Feed–Forward ANNs. To prevent the ANN from overfitting the number of training epochs is essential.

For the automatic generation of problem–adapted topologies a method based on Genetic Algorithms (GA) is employed. With this approach populations of ANNs are generated, trained, and evaluated for the land–cover classification task. Individuals which solve the given task well receive a high fitness value and are selected to form the next generation.

A system performing this artificial evolution of ANNs - the netGEN system - has been implemented by the authors. Due to its computational complexity, netGEN has been designed as a parallel system, i.e. the learning phase of the ANNs is distributed among an arbitrary amount of workstations within a heterogenous cluster. Results for the chosen test site "Nationalpark Hohe Tauern", where Landsat TM data, a Digital Elevation Model (DEM), and ground–truth data taken from ground measurements are available, are given.

1. INTRODUCTION

ANNs have been successfully applied in the field of Remote Sensing Data Analysis over the last few years. Results on *Land Use/Land Cover* (LULC) classification from various imagery have been reported, e.g. Bischof *et al.* [1] and Civco [2] on Landsat TM images, Wu *et al.* [3] on SPOT images and Wilkinson *et al.* [4] on a combination of ERS-1 SAR and Landsat TM images. ANNs assume no distributional form for the class densities which is an advantage over statistical classifiers and has also been studied in recent years [5], [6].

Due to their reported properties, their independency from a priori statistical parameters and their inherent parallel nature, we decided to use ANNs to solve our LULC task. In particular, we use a form of *Multilayer Feed–Forward Networks* and supervised learning by means of *Error–Back–Propagation*. One problem which has not been investigated in detail in recent papers, although, it was mentioned in [1] and [2], is the problem of finding the appropriate topology (structure), e.g. the number of nodes, layers, connections, the connectivity, of the ANN. Commonly, a satisfying ANN topology is found by experiments or by applying rules–of–thumb.

We use a GA to adapt the ANN topology to our specific classification task. Each individual of a generation, a neural network in a population of ANNs, is trained and tested so as to evaluate ANN performance and to assign a fitness to the individual. After a selection process the individuals with the highest fitness are used to form the next generation. In the following we will use the term *netGEN* for our implementation of this method.

netGEN has been implemented as a parallel system where the learning of the ANNs is distributed among several workstations. The learning of the individual ANN is performed by an ANN simulator, namely, the *SNNS* (Stuttgart Neural Network Simulator) [7]. For the message passing between the individual modules we employed the *PVM* (Parallel Virtual Machine) library [8].

The application area is the Austrian National Park "Hohe Tauern", an alpine region with elevations ranging from about 1000 meters up to 2500 meters. The data set consists of the 6 non-thermal Landsat TM bands and a DEM which has also been used for prior geo–coding. A ground–truth map containing 35 land cover classes as the result of a ground data collection campaign is available.

2. EVOLUTION OF ARTIFICIAL NEURAL NETWORKS

Various methods have been suggested for the automatic construction of ANN topologies. Among these are *Network Growing* (OBS [9], OBD [10], etc.), *Network Pruning* (Cascade Correlation [11], etc.), and *Evolutionary*

Design of networks. In the netGEN system we chose the latter approach by employing Genetic Algorithms. We restricted the genetic search to Feed–Forward Networks which are trained by standard Error–Back–Propagation with a fixed *learn rate* and no *momentum term*.

Out of numerous existing genotype representations of an ANN [12], we adopted a *direct encoding* method introduced by *Miller & Todd* [13]. Their original encoding - in the following called *Miller–Matrix* (MM) - uses an adjacency matrix to represent the topology of an ANN. Each entry $w_{ij} \in \{0(\text{no connection}), L(\text{learnable connection})\}$ in the matrix at position (i, j) indicates the absence or presence of a directed connection between unit j and unit i. Further, they added a column for the biases of the units. Hence, for a network with n units the MM has dimension $n \times (n + 1)$.

In our *Modified–Miller–Matrix* MMM encoding scheme [14] we dropped the bias column, since all the units in our ANNs are assumed to be learnable. As the number of units varies only slightly with the MM encoding scheme (due to isolated units), we added *Neuron Markers* in the main diagonal (i, i) of the MMM, indicating the absence or presence of a particular neuron i. As a consequence, the only topology parameters which have to be set are the number of input and output neurons (depending on the problem) and the maximum number of neurons.

In order to combat overfitting [15], we also encoded the number of training epochs with the additional benefit of another ANN parameter being set automatically.

In addition to our modifications of the MM encoding scheme, the GA does not operate on the matrix representation, but on the concatenation of the rows of the non–zero part of the MMM's lower triangle matrix representing the possible connections. All MMM entries clamped to zero need not be processed by the GA, hence, they are not encoded in the genotype.

The general form of the genotype looks like:

$$e_1 e_2 \ldots e_{n_1} m_1 m_2 \ldots m_{n_2} x_1 x_2 \ldots x_{n_3},$$

where

1. $e_i \in \{0, 1\}, i = 1, \ldots, n_1$ is the binary representation of the number of epochs,
2. $m_i \in \{0, 1\}, i = 1, \ldots, n_2$ are the neuron markers, where $m_i = 0$ indicates the absence and $m_i = 1$ indicates the presence of a neuron and
3. $x_i \in \{0, 1\}, i = 1, \ldots, n_3$ indicate the absence/presence of a connection. The x_i represent possible connections (non–zero entries in the MMM). Reflexive connections, connections within input and output layer, and connections from unit i to unit j with $i < j$, i. e. constant zero entries in the MMM, are not encoded.

This enables the use of the standard genetic operator *2–Point–Crossover* [16] which has a more global effect on the bit string representation (and therefore, the ANN topology) than the MM crossover operator exchanging rows or columns of two individuals. The mutation operator used in netGEN is the standard bit–flip with mutation probability p_m.

For the GA selection phase we chose *Binary Tournament Selection*, for it showed mostly better results than *Proportionate Selection Methods*. First analytical steps, possibly confirming this observation, have been reported recently [17].

Besides a "good" encoding of the problem, the choice of an appropriate fitness function is essential to the GA and its capability of finding (sub)optimal solutions. As for the LULC problem, we experimented with various fitness functions using penalties so as to assess the distance of a misclassification to a correct classification. An ANN whose observed winner neuron in the output layer is close to the wanted winner with one particular test pattern receives a small penalty which increases with distance [18]. However, at the time we arrived at the best solutions with a simple fitness function (the classification accuracy of an individual ANN on the test data set) without penalties.

The following GA and ANN parameters have been used with all the experiments in this paper:
Population Size = 50, Generations = 30, Crossover Probability $p_c = 0.6$, Mutation Probability $p_m = 0.005$, Crossover = 2–Point, Selection Method = Binary Tournament.
Network Topology = Feed–Forward, Transfer Function (all neurons) = Sigmoid, Training = Back–Propagation, Learning Parameter $\eta = 0.2$, No Momentum Term, Number of Epochs = Evolutionary, Training Set Size = 197, Test Set Size = 1000.

3. SYSTEM LAYOUT AND IMPLEMENTATION

The netGEN system [14] has been implemented on a network cluster of SPARC and NeXT workstations in order to parallelize the learning phase of ANNs.

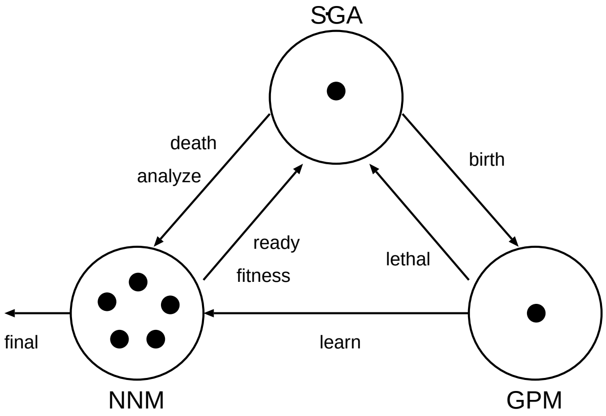

Figure 1: The netGEN system.

In figure 1 the main components, the Simple Genetic Algorithm (SGA) [19], the Genotype/Phenotype Mapping (GPM), and the Neural Network Manager (NNM), of the netGEN system are shown.

SGA generates a blueprint for each ANN, a genotype in GA terminology, which is processed by the GPM module mapping the ANN genotype to a phenotype adapted to the Neural Network Simulator. The mapping module's task is to prepare valid configuration files and to dismiss ANN topologies which cannot be processed by netGEN. If the topology is valid, a command, e.g. *learn*, is passed to the NNM module. The NNM interfaces to the ANN simulator, evaluates the fitness of each ANN and passes it back to the SGA. With the parallel implementation a set of NNM processes run on different machines and/or different processors.

netGEN uses the *SPMD* (Single Program Multiple Data) paradigm which is an extension to the model of computation introduced by *Flynn* [20] where *SISD, MISD, SIMD* and *MIMD* machines are distinguished. In SPMD the same program is run on the processors of a MIMD machine, in our case copies of NNM programs run on different machines which make up the Parallel Virtual Machine (PVM). A multiple data stream is observed, because the ANNs are trained with different network configurations.

Furthermore, we like to mention that we achieve *implicit load balancing* which means that more ANNs are passed to the faster machines within the PVM. A simple time–out mechanism prevents the system from deadlock and performance reducing situations when waiting for results from crashed or very slow machines.

4. DATA PREPROCESSING

The National Park "Hohe Tauern" which serves as the classification test area is located in the South of the Austrian Federal State Salzburg. The Landsat TM data have been obtained from path 192 for August 7th 1991 at 9:30 am. Landsat TM channel 6, the thermal infrared channel, is not used in the classification data set. The remaining 6 channels have been geo–coded to the grid resolution of 25 meter pixels. The DEM which is available at a resolution of 50 meters has also been oversampled to 25 meter pixel resolution.

The ground–truth is the digital version of the vegetation map for the "Hohe Tauern" area. The vegetation map was published in 1985 and contains 35 different vegetation classes for the area in question [21].

The area in question is a 646 by 661 pixels scene. Different pixels from this area have been chosen for training, verification and, finally, classification. The 35 vegetation classes have been reduced to 6. This was necessary, because of the similarity of many classes, e.g. 14 types of grassland are distinguished in the map.

The large time lag between the completion of the vegetation map and the Landsat data take results in a degradation of the ground–truth accuracy. To reduce the impact of possible ground–truth inaccuracies, two filters have been applied in preprocessing, a *k–Nearest Neighbor* (k–NN)–filter in the feature space and another one in the image space.

The use of the k–NN–filter to prevent overfitting has been discussed in [15]. Only training samples with similar spectral signature in terms of the Euklidian distance are retained for a ground–truth class. A training sample is deleted from the pattern set, if the majority of the k neighbors belongs to a different ground–truth class. Thus, spectral coherence is increased for patterns of the same ground–truth class.

The second filter ensures class homogeneity within a $m \times m$ window in the spatial domain, where $k = (m \times m) - 1$, around the pixel in question. A training sample is deleted from the pattern set, if not all pixels in the spatial neighborhood belong to the same ground–truth class. Pixels representing mixed class membership which are found next to class boundaries, eventually, caused by inaccurate ground–truth, are removed by the spatial filter.

5. A LANDSAT TM LAND COVER CLASSIFICATION EXAMPLE

For the given region three classification methods have been applied, a Maximum–Likelihood classification, using the GRASS [22] `i.maxlik` function, an ANN classification using a fully connected ANN with one hidden layer, and an ANN classification using a topology generated by netGEN (22 Units, 136 Connections, 1430 Epochs). Results are given in table 1 using a confusion table, where the columns are the reference categories, i.e. the ground–truth categories, and the rows are the classification categories. Each entry is given in percent of the number of samples in the considered reference class. The bottom line gives the *producer's accuracy* [23], i.e. the number of correctly classified samples from a category to the total number of reference samples of that category. Both ANNs were trained using the same training set, the training set for the Maximum–Likelihood classification was obtained using the GRASS function `i.gensig`.

	Maximum–Likelihood						7–7–6 ANN						netGEN ANN					
	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
1	83.9	43.5	20.1	0.9	0.0	5.7	69.9	22.9	7.9	0.3	0.0	12.3	73.2	26.9	11.0	0.3	0.0	13.6
2	12.3	45.6	19.8	1.5	0.0	0.7	16.4	44.3	13.6	0.6	0.0	1.3	3.0	65.1	2.0	0.1	0.0	2.2
3	2.8	8.5	53.5	38.1	0.8	22.2	13.4	30.6	62.1	14.3	0.0	20.4	21.8	4.4	68.9	14.0	0.0	19.7
4	0.1	0.3	4.4	46.7	19.9	6.1	0.0	1.0	15.0	78.6	28.7	14.2	0.1	0.5	16.1	80.4	34.0	13.1
5	0.0	0.0	0.0	7.4	73.1	4.2	0.0	0.0	0.0	4.8	71.0	3.1	0.0	0.1	0.0	2.8	65.9	2.8
6	0.9	2.1	2.1	5.3	6.1	55.0	0.2	1.2	1.3	1.4	0.2	48.7	1.3	3.0	2.0	2.4	0.1	48.6
	53.6						68.2						73.2					

1...Forest, 2...Shrubs, 3...Grassland, 4...No Vegetation, 5...Glacier, 6...Water

Table 1: Classification accuracy of different classification methods.

Figure 2 compares the different classification maps with the ground–truth map.

6. DISCUSSION

As can be seen from table 1, the overall accuracy for the ANN generated by netGEN is 73 percent. Considering the time span of 6 years between vegetation map publication and satellite data take, it is obvious that the 1985 ground–truth is just an approximation to the 1991 land cover. Due to this discrepancy, the relative low accuracy compared to similar work can be explained.

The netGEN ANNs require a surprisingly small training data set (197 samples), although, it has to be mentioned that the performance of a standard topology like the 7–7–6 ANN comes very close to netGEN ANNs by increasing the number of samples in the training data set. We believe that this effect depends mainly on the rather simple fitness function we are currently using, i. e. the netGEN topology could still be improved.

Furthermore, by encoding the number of training epochs in the ANN genotype representation overfitting of netGEN ANNs could not be observed, but this could also be credited to the k–NN filtering or to the specific topology of netGEN ANNs, which will be investigated closely in our future research.

7. OUTLOOK

We have presented the netGEN system which has been applied to a land cover classification problem. Further research interests and implementation issues include the employment of a variety of genetic operators, an evolutionary approach to the generation of training data sets, evaluation of more complex fitness functions, and experiments with different input representations, e. g. texture, slope.

8. ACKNOWLEDGEMENTS

We wish to thank the *ZGIT*, the Center for Geographic Information Processing, at the Department of Geography, University of Salzburg, for providing TM and ground–truth data and Thomas Fischer of the Austrian Forest–Technical Service in Torrent and Avalanche Control, District Salzburg, for helpful comments on the vegetation map.

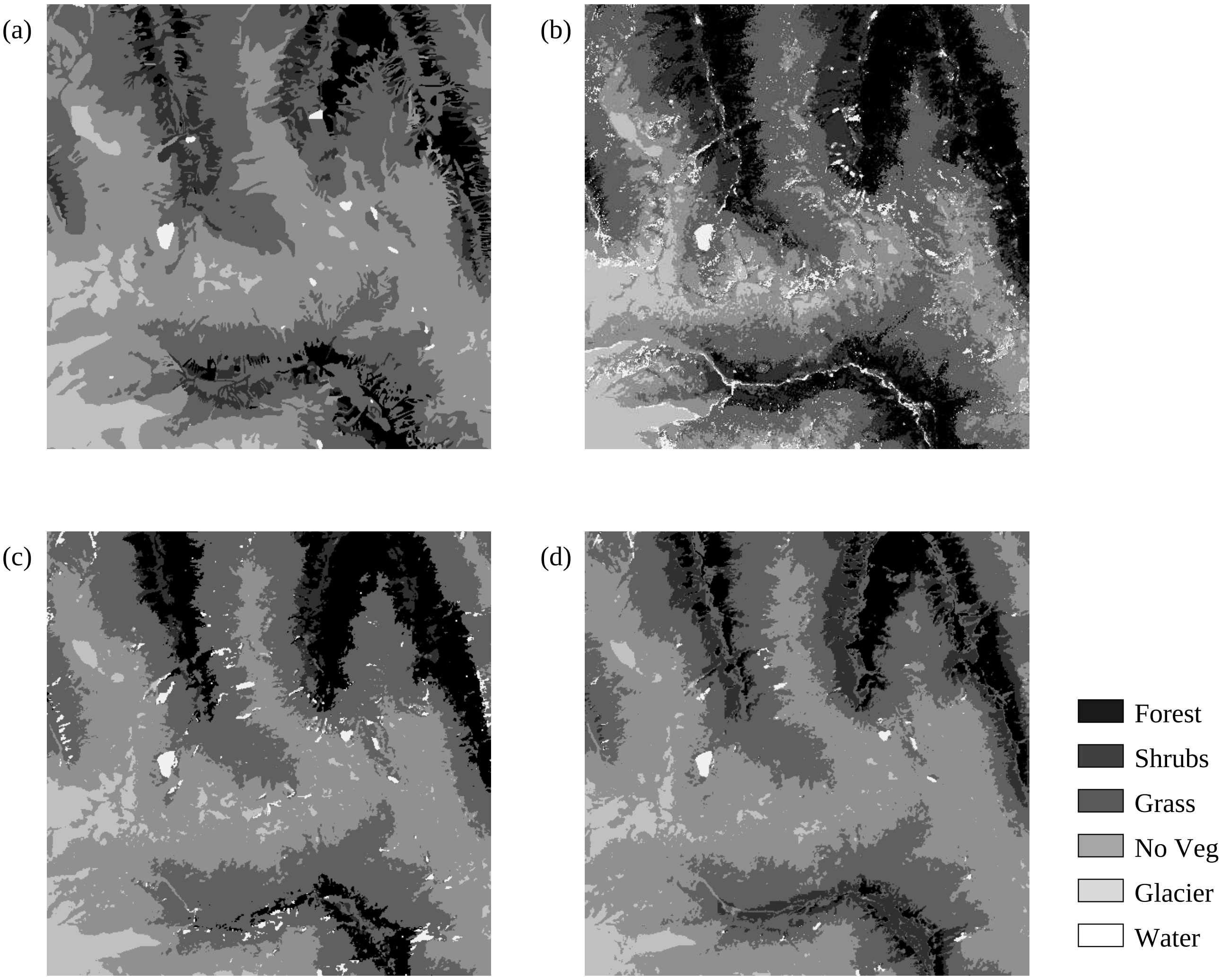

Figure 2: Comparison of different classification methods: (a) Ground–Truth, (b) Maximum–Likelihood (c) 7–7–6 ANN (d) netGEN ANN

9. REFERENCES

[1] H. Bischof, W. Schneider, and A.J. Pinz. Multispectral Classification of Landsat-images using Neural Networks. *IEEE Transactions on Geoscience and Remote Sensing*, 30(3):482–490, May 1992.

[2] Daniel L. Civco. Artificial Neural Networks for Land-Cover Classification and Mapping. *International Journal of Geographical Information Systems*, 7(2):173–186, 1993.

[3] Xiping Wu and James D. Westervelt. Using Neural Networks to Correlate Satellite Imagery and Ground-Truth Data. Technical Report EC-94/28, USACERL Special Report (SR), August 1994.

[4] G.G. Wilkinson, S. Folving, I. Kanellopoulos, N. McCormick, K. Fullerton, and J. Megier. Forest Mapping from Multi-Source Satellite Data using Neural Network Classifiers - an Experiment in Portugal. *Remote Sensing Reviews*, 12, 1994.

[5] Ian D. Downey, Clare H. Power, Iannis Kanellopoulos, and Graeme Wilkinson. A Performance Comparison of Landsat TM Land Cover Classification Based on Neural Network Techniques and Traditional Maximum Likelihood and Minimum Distance Algorithms. In *Proceedings of the 1992 Annual Conference of the UK Remote Sensing Society, Dundee*. The Remote Sensing Society, Nottingham, September 1992.

[6] Richard P. Lippman. Neural Networks, Bayesian a Posteriori Propabilities, and Pattern Classification. In *Proceedings of the NATO Advanced Study Institute From Statistics to Neural Networks, Theory and Pattern Recognition Applications, Les Arcs, France*, June 1993.

[7] Andreas Zell, Guenter Mamier, Michael Vogt, Niels Mach, Ralf Huebner, Kai-Uwe Herrmann, Tobias Soyez, Michael Schmalzl, Tilman Sommer, Artemis Hatzigeogiou, Sven Doering, and Dietmar Posselt. *SNNS Stuttgart Neural Network Simulator, User Manual.* University of Stuttgart, 1994.

[8] Al Geist, Adam Beguelin, Jack Dongarra, Weicheng Jiang, Robert Manchek, and Vaidy Sunderam. *PVM 3 User's Guide and Reference Manual.* Oak Ridge National Laboratory, 1994.

[9] B. Hassibi, D.G. Stork, and G.J. Wolff. Optimal Brain Surgeon and General Network Pruning. In *Proceeding IEEE International Conference on Neural Networks, San Francisco*, volume 1, pages 293–299, 1993.

[10] Y.L. Le Cun, J.S. Denker, and S.A. Solla. Optimal Brain Damage. In D. Touretzky, editor, *Advances in Neural Information Processing Systems, Vol. 2*, volume 2, pages 599–605. Morgan Kaufmann, 1990.

[11] S.E. Fahlman and C. Lebiere. The cascade-correlation learning architecture. In D. Touretzky, editor, *Advances in Neural Information Processing Systems, Vol. 2*, volume 2, pages 524–532. Morgan Kaufmann, 1990.

[12] Karthik Balakrishnan and Vasant Honavar. Evolutionary Design of Neural Architectures – A Preliminary Taxonomy and Guide to Literature. Technical Report CS TR #95-01, Iowa State University, Department of Computer Science, Ames, Iowa 50011–1040, U.S.A., January 1995.

[13] Geoffrey F. Miller, Peter M. Todd, and Shailesh U. Hegde. Designing Neural Networks using Genetic Algorithms. In J. David Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*, pages 379–384, San Mateo, California, 1989. Philips Laboratories, Morgan Kaufman Publishers, Inc.

[14] Reinhold Huber, Helmut A. Mayer, and Roland Schwaiger. netGEN - A Parallel System Generating Problem-Adapted Topologies of Artificial Neural Networks by means of Genetic Algorithms. In *Beiträge zum 7. Fachgruppentreffen Maschinelles Lernen der GI-Fachgruppe 1.1.3, Forschungsbericht Nr. 580, Dortmund*, August 1995.

[15] Paul L. Rosin and Freddy Fierens. Improving Neural Network Generalisation. In *Proceedings of IGARSS'95, Firenze, Italy*, July 1995.

[16] David E. Goldberg. *Genetic Algorithms in Search, Optimization & Machine Learning.* Addison-Wesley, 1989. (ISBN 0-201-15767-5).

[17] Tobias Blickle and Lothar Thiele. A Mathematical Analysis of Tournament Selection. In L. Eshelman, editor, *Genetic Algorithms: Proceedings of the 6th International Conference (ICGA95)*, San Mateo, California, 1995. Morgan Kaufman Publishers, Inc.

[18] Jon T. Richardson, Mark R. Palmer, Gunar Liepins, and Mike Hilliard. Some Guidelines for Genetic Algorithms with Penalty Functions. In J. David Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*, pages 191–197, San Mateo, California, 1989. Philips Laboratories, Morgan Kaufman Publishers, Inc.

[19] Robert E. Smith, David E. Goldberg, and Jeff A. Earickson. SGA-C: A C–Language Implementation of a Simple Genetic Algorithm. TCGA Report 91002, The Clearinghouse for Genetic Algorithms, The University of Alabama, Department of Engineering Mechanics, Tuscaloosa, AL 35487, May 1991.

[20] George S. Almasi and Allan Gottlieb. *Highly Parallel Computing.* Benjamin/Cummings, 1989. ISBN 0-8053-0177-1.

[21] Hugo M. Schiechtl and Roland Stern. *Die aktuelle Vegetation der Hohen Tauern (Nationalpark Hohe Tauern, wissenschaftliche Schriften). Matrei in Osttirol und Großglockner: Vegetationskarte 1:25000.* Universitätsverlag Wagner, Innsbruck, 1985.

[22] James D. Westervelt, Michael Shapiro, and William D. Goran. GRASS User's Reference Manual. Technical Report N-87/22, USACERL ADP Report, September 1988.

[23] Xin Zhuang, Bernard A. Engel, Xiaoping Xiong, and Chris J. Johannsen. Analysis of Classification Results of Remotely Sensed Data and Evaluation of Classification Algorithms. *Photogrammetric Engineering and Remote Sensing*, 61(4):427–433, April 1995.